International Series in Operations Research & Management Science

Volume 213

For further volumes:
http://www.springer.com/series/6161

Joe Zhu

Quantitative Models for Performance Evaluation and Benchmarking

Data Envelopment Analysis with Spreadsheets

Third Edition

Springer

Joe Zhu
School of Business
Worcester Polytechnic Institute
Worcester
Massachusetts
USA

ISBN 978-3-319-37438-3 ISBN 978-3-319-06647-9 (eBook)
DOI 10.1007/978-3-319-06647-9
Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To
Andie, Alec, and Yao

Preface

To improve performance, organizations need to constantly evaluate operations or processes related to products, services, marketing, and others. Performance evaluation and benchmarking are a widely used method to identify and adopt best practices as a means to improve performance and increase productivity, and are particularly valuable when no objective or engineered standard is available to define efficient and effective performance. For this reason, benchmarking is often used in managing service operations, because service standards (benchmarks) are more difficult to define than manufacturing standards.

Benchmarks can be established but are somewhat limited as they work with single measurements one at a time. It is difficult to evaluate an organization's performance when there are multiple performance metrics related to a system or operation. The difficulties are further enhanced when the relationships among the performance metrics are complex and involve unknown tradeoffs. It is critical to show benchmarks where multiple performance metrics exist. The current book introduces the methodology of data envelopment analysis (DEA) and its uses in performance evaluation and benchmarking under the context of multiple performance measures.

DEA uses mathematical programming techniques and models to evaluate the performance of peer units (e.g., bank branches, hospitals, and schools) in terms of multiple performance metrics/measures/features. These peer units are called Decision Making Units (DMUs). The performance of DMUs is measured based upon a set of selected performance measures/metrics. These performance metrics are classified as "inputs" and "outputs" in DEA. However, "inputs" and "outputs" in DEA do not necessarily represent inputs and outputs of production processes. For example, if one benchmarks the performance of computers, it is natural to consider different features (screen size and resolution, memory size, process speed, hard disk size, and others). One would then have to classify these features into "inputs" and "outputs" in order to apply a proper DEA analysis. However, these features may not actually represent inputs and outputs at all, in the standard notion of production. Therefore, the notion of DEA "inputs" and "outputs" is generic. DEA "inputs" and "outputs" can be inputs and outputs of production processes, but can also be general performance measures. In the former case, DEA yields an efficiency score, and the latter case a composite measure.

Because of the flexibility of DEA, researchers in a number of fields have quickly recognized that DEA is an excellent methodology for modeling operational processes. DEA's empirical orientation and absence of *a priori* assumptions have resulted in its use in a number of studies involving best-practice identification in the nonprofit sector, in the regulated sector, and in the private sector. DEA applications involve a wide range of contexts, such as education, health care, banking, armed forces, auditing, market research, retail outlets, organization effectiveness, transportation, public housing, and manufacturing.

The motivation for this book is three-fold. First, as DEA is being applied to a variety of efficiency evaluation problems, managers may want to conduct performance evaluation and analyze decision alternatives without the help of sophisticated modeling programs. For this purpose, spreadsheet modeling is a suitable vehicle. In fact, spreadsheet modeling has been recognized as one of the most effective ways to evaluate decision alternatives. It is easy for users to apply various DEA models in spreadsheets. The book introduces spreadsheet modeling into DEA, and shows how various conventional and new DEA approaches can be implemented using Microsoft® Excel and Solver. With the assistant of the developed DEA spreadsheets, the user can easily develop new DEA models to deal with specific evaluation scenarios.

Second, new models for performance evaluation and benchmarking are needed to evaluate business operations and processes under a variety of contexts. After briefly presenting the basic DEA techniques, the current book introduces new DEA models and approaches. For example, a context-dependent DEA measures the relative attractiveness of competitive alternatives. Sensitivity analysis techniques can be easily applied, and used to identify critical performance measures. Two-stage DEA models deal with multi-stage efficiency evaluation problems. DEA benchmarking models incorporate benchmarks and standards into DEA evaluation. Cross efficiency provides peer evaluation scores.

All these new models can be useful in benchmarking and analyzing complex operational efficiency in manufacturing organizations as well as evaluating processes in banking, retail, franchising, health care, e-business, public services, and many other industries.

Third, although the spreadsheet modeling approach is an excellent way to build new DEA models, an integrated easy-to-use DEA software can be helpful to managers, researchers, and practitioners. Therefore the current version includes a *DEA-Frontiers* software which is a DEA Add-In for Microsoft Excel and offers the user the ability to perform a variety of DEA models and approaches—it provides a custom Excel menu which calculates various different DEA models and can solve up to 50 DMUs, subject to the capacity of Excel Solver.

This third edition improves a number of DEA spreadsheet models. Several new DEA models and approaches are added. For example, cross efficiency approaches, and interval data treatment are new addition to the book. Bootstrapping in DEA is added into the *DEAFrontier* software. The third edition is reorganized to better present the traditional and new DEA models and approaches.

I would like to offer my sincere thank to my mentor, friend and collaborator, Dr. Lawrence M. Seiford who helped and enabled me to contribute to dual areas of

DEA methodology and applications, and I would like to acknowledge my research collaborators, in particular, Professor Wade Cook, whose efforts made this edition possible. I would like to dedicate this edition to the memory of Late Dr. William W. Cooper who constantly supported my DEA research.

I would also like to thank the following individuals for pointing errors in the manuscript: Ya Chen, Huaqing Wu, Xiaoning Xu, Baocheng Zhang, Zhixaing Zhou, and Weiwei Zhu. However, any errors in this edition are entirely my responsibility, and I would be grateful if anyone would bring any such errors to my attention.

2014 Massachusetts Joe Zhu

Contents

About the Author

Prof. Joe Zhu is one of the prominent researchers in the field of Data Envelopment Analysis (DEA). His research interests are in the areas of operations and business analytics, productivity modeling, and performance evaluation and benchmarking. He has published over 100 articles in peer-reviewed journals including *Operations Research, Sloan Management Review, European Journal of Operational Research, Journal of the Operational Research Society, Naval Research Logistics, IIE Transactions, Journal of Banking and Finance, OMEGA*, and others. He is an Area Editor of *OMEGA*, an Associate Editor of *INFOR*, and the Associate Series Editor of Springer's *International Series in Operations Research and Management Science*.

He is a Japan Society for Promotion of Science (JSPS) fellow and a William Evans Visiting Fellow of University of Otago, New Zealand. His research has been supported by KPMG Foundation, National Institute of Health, and Department of Veterans Affairs.

Chapter 1
Data Envelopment Analysis

Employees who seem to work the least can often be the most productive

1.1 Performance Evaluation and Tradeoffs

All business operations/processes involve transformation—adding values and changes to materials and turning them into goods and services that customers want. Managers are often interested in evaluating how efficiently various processes operate with respect to multiple performance measures (or metrics). Organizations are interested in knowing their performance with respect to the use of resources such labor, materials, energy, machines, and other, and the outcomes such as the quality of finished products, services, customer satisfaction. Consider hospital operations, for example. The performance measures or metrics include doctors, nurses, medical supplies, equipment, laboratories, beds, number of patients treated, number of interns and residents trained, and others. In a buyer-seller supply chain, the buyer may be interested in comparing the performance of several sellers with respect to response time, costs, flexibility, customer service, quality, and customization. Eliminating or improving inefficient operations decreases the cost and increases productivity. Performance evaluation and benchmarking help business operations/processes to become more productive and efficient.

Performance evaluation is an important continuous improvement tool for business to stay competitive and plays an important role in the global market where competition is intense and grows more so each day. Performance evaluation and benchmarking positively force any business unit to constantly evolve and improve in order to survive and prosper in a business environment facing global competition. Through performance evaluation, one can (i) reveal the strengths and weaknesses of business operations, activities, and processes, (ii) better prepare the business to meet its customers' needs and requirements, and (iii) identify opportunities to improve current operations and processes, and create new products, services and processes.

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*,
International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_1,

Single-measure based gap analysis is often used as a fundamental method in performance evaluation and benchmarking. However, as pointed out by Camp (1995), one of the dilemmas that we face is how to show benchmarks where multiple measurements exist. It is rare that one single measure can suffice for the purpose of performance evaluation. The single output to input financial ratios, such as, return on investment (ROI) and return on sales (ROS), may be used as indices to characterize the financial performance. However, they are unsatisfactory discriminants of "best-practice", and are not sufficient to evaluate operating efficiency. Since a business unit's performance is a complex phenomenon requiring more than a single criterion to characterize it. For example, as pointed out by Sherman (1984), a bank branch may be profitable when profit reflects the interest and the revenues earned on funds generated by the branch less the cost of these funds and less the costs of operating the branch. However, this profit measure does not indicate whether the resources used to provide customer services are being managed efficiently.

Further, the use of single measures ignores any interactions, substitutions or tradeoffs among various performance measures. Each business operation has specific performance measures or metrics with tradeoffs. For example, consider the tradeoff between total supply chain cost and supply chain response time, measured by the amount of time between an order and its corresponding delivery. Figure 1.1 illustrates alternate supply chain operations S1, S2, S3, and S, and the best-practice (efficient) frontier or tradeoff curve determined by them. A supply chain whose performance (or strategy) is on the efficient frontier is non-dominated (efficient) in the sense that no alternate supply chain's performance is strictly better in both cost and response time. Through performance evaluation, the efficient frontier that represents the best practice is identified, and an inefficient strategy (e.g., point S) can be improved (moved to the efficient frontier) with suggested directions for improvement (to S1, S2, S3 or other points along the frontier).

Optimization techniques can be used to estimate the efficient frontier if we know the functional forms for the relationships among various performance measures. For example, stockout levels and inventory turns are two mutually dependent variables with performance tradeoffs. Technological and process innovations can shift the cost tradeoff curves by reducing the cost of achieving lower inventories at a particular stockout level or the cost of achieving lower stockouts at a particular inventory level. Unfortunately, such information is usually not completely available.

Without *a priori* information on the tradeoffs, the functional forms cannot be specified. Consequently, we cannot fully characterize the business operations and processes. Note that the objective of performance evaluation is to evaluate the current business operation internally and to benchmark against similar business operations externally to identify the best practice. Thus, such best-practices can be empirically identified. We can empirically identify or estimate the best-practice or efficient frontier based upon observations on one business operation/process over time or similar business operations at a specific time period.

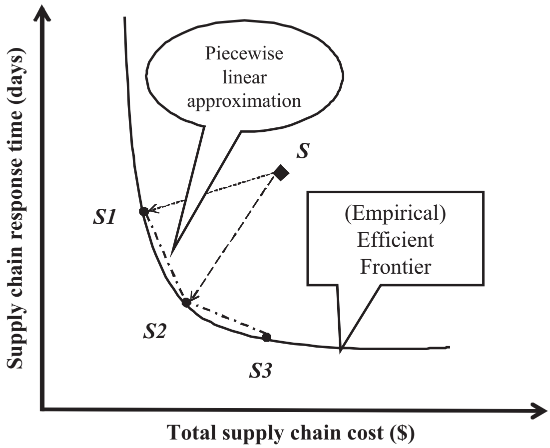


Fig. 1.1 Best practice (efficient) frontier of supply chain operations

1.2 Data Envelopment Analysis

This book is about data envelopment analysis (DEA) and its models using spreadsheet modeling. What is DEA? DEA is a data analysis tool for identifying best-practices as shown in Fig. 1.1 when such a best-practice frontier is characterized by multiple performance metrics. In DEA, performance metrics are classified as "inputs" and "outputs". See Sect. 1.3 for detailed discussion on DEA inputs and outputs.

According to Cooper et al. (2011a):

> DEA is a relatively new "data oriented" approach for evaluating the performance of a set of peer entities called Decision Making Units (DMUs) which convert multiple inputs into multiple outputs. The definition of a DMU is generic and flexible. Recent years have seen a great variety of applications of DEA for use in evaluating the performances of many different kinds of entities engaged in many different activities in many different contexts in many different countries. These DEA applications have used DMUs of various forms to evaluate the performance of entities, such as hospitals, US Air Force wings, universities, cities, courts, business firms, and others, including the performance of countries, regions, etc. Because it requires very few assumptions, DEA has also opened up possibilities for use in cases which have been resistant to other approaches because of the complex (often unknown) nature of the relations between the multiple inputs and multiple outputs involved in DMUs.

Throughout the book, we use decision making units (DMUs) to represent business operations or processes. Each DMU is evaluated based upon a set of multiple performance measures that are classified as "inputs" and "outputs".

For now, suppose we have a set of observations on n DMUs. Each observation consists of values of performance measures related to a DMU$_j$ ($j=1, \ldots, n$). The selected set of performance measures are classified as m inputs x_{ij} ($i=1, 2, \ldots, m$) and s outputs y_{rj} ($r=1, 2, \ldots, s$).

DEA uses linear programming techniques to identify the (empirical) efficient frontier or best-practice frontier for these n observations. The following two properties ensure that we can develop a piecewise linear approximation to the efficient frontier and the area dominated by the frontier, as shown in Fig. 1.1.

Property 1.1 *Convexity*. $\sum_{j=1}^{n} \lambda_j x_{ij}$ $(i = 1, 2, \ldots, m)$ and $\sum_{j=1}^{n} \lambda_j y_{rj}$ $(r = 1, 2, \ldots, s)$ are possible input and output levels achievable by the DMU$_j$, where λ_j $(j = 1, \ldots, n)$ are nonnegative scalars such that $\sum_{j=1}^{n} \lambda_j = 1$.

Property 1.2 *Inefficiency*. The same y_{rj} can be obtained by using $\hat{x}_{ij}$, where $\hat{x}_{ij} \geq x_{ij}$ (i.e., the same output levels can be achieved by using more inputs); The same x_{ij} can be used to achieve $\hat{y}_{rj}$, where $\hat{y}_{rj} \leq y_{rj}$ (i.e., the same input levels can be used to achieve less outputs).

Consider Fig. 1.1 where total supply chain cost and supply chain response time represent two inputs. Applying Property 1.1 to S1, S2, and S3 yields the piecewise linear approximation to the curve shown in Fig. 1.1. Applying both properties expands the line segments S1S2 and S2S3 into the area dominated by the curve.

For specific x_i $(i = 1, 2, \ldots, m)$ and y_i $(r = 1, 2, \ldots, s)$, we have

$$\begin{cases} \sum_{j=1}^{n} \lambda_j x_{ij} \leq x_i & i = 1, 2, \ldots, m \\ \sum_{j=1}^{n} \lambda_j y_{rj} \geq y_r & r = 1, 2, \ldots, s \\ \sum_{j=1}^{n} \lambda_j = 1 \end{cases} \tag{1.1}$$

The next step is to estimate the empirical (piecewise linear) efficient frontier characterized by (1.1). DEA uses linear programming to implicitly estimate the tradeoffs inherent in the empirical efficient frontier. DEA introduced by Charnes et al. (1978) has been proven an effective tool in identifying such empirical frontiers and in evaluating relative "efficiency".

Here "efficiency" is a generic term that can represent a variety of cases depending on a particular set of DMUs and a set of associated performance measures. For example, if performance measures are inputs and outputs of a production process, then DEA "efficiency" is a "production efficiency". If performance measures are quality indicators, then DEA "efficiency" yields a composite quality measure.

In fact, in addition to be used as an estimate of "production efficiency", DEA is a "balanced benchmarking" (Sherman and Zhu 2013) that examines performance in multiple criteria and helps organizations to test their assumptions about performance, productivity, and efficiency. Under general benchmarking, the DEA score may no longer be referred to as "production efficiency". In this case, we may wish to refer to the DEA score as a form of "overall performance" of an organization. Such "overall performance" can appear in the form of composite measure that aggregates individual indicators (inputs and outputs) via a DEA model. For example, composite measures (DEA scores) of quality allow senior leaders to better benchmark their organization's performance against other high-performing organizations (Shwartz et al. 2009). The DEA inputs and outputs represent more than the "inputs" and "outputs" under the notion of production process, and DEA is more than an efficiency measure under the notion of production process. In the next section, we will discuss what constitute DEA inputs and outputs.

DEA was designed to measure the relative efficiency where market prices are not available (see, e.g., Charnes et al. 1981; Johnson and Zhu 2002). However, by its ability to model multiple-input and multiple-output relationships without *a priori* underlying functional form assumption, DEA has also been widely applied to other areas. See Liu et al. (2013) for a comprehensive survey on DEA applications from 1978 to 2010. These authors identify the top five DEA applications areas as banking, health care, agriculture and farm transportation, and education, and the applications with the highest growth momentum recently as energy, environment, and finance.

Such previous DEA studies provide useful managerial information on improving the performance. In particular, DEA is an excellent tool for improving the productivity of service businesses (Sherman and Zhu 2006).

In the current book, we present various DEA approaches that can be used in identifying best-practice frontier and further in performance evaluation and benchmarking. Other recommended readings include several DEA handbooks such as Cooper et al. (2011a), Cook and Zhu (2014), and Zhu (2015).

1.3 Performance Metrics Classified as Inputs and Outputs

DEA requires that performance measures or metrics be classified into inputs and outputs. Whether it is the researcher, the practitioner or the student, the use of the DEA methodology gives rise to an important question before proceeding to a DEA analysis:

> What are the outputs and inputs to be used to characterize the performance of those DMUs?

As discussed in Cook et al. (2014), in the literature, DEA is generally introduced as a mathematical programming approach for measuring relative efficiencies of DMUs, when multiple inputs and multiple outputs are present. While the concept of inputs and outputs is well understood, it is often the case that researchers/users take the notion for granted, and little attention tends to be paid to insuring that the selected measures properly reflect the "process" under study. As a case in point, the original DEA model of Charnes et al. (1978, 1981), involving the study of school districts in Texas, was developed in a ratio form of Outputs/Inputs, but the authors provide little in the way of rationalization in regard to appropriate variables (inputs and outputs) for studying student performance. This is not to imply that the variables used were not appropriate for the problem at hand, but rather it serves to illustrate that the paper, like many of those that followed over the past three decades, was primarily focused on methodological development. One gets the sense in much of the literature that there is little need to spend time laboring over how a process actually works. After all, in a production or service process, inputs and outputs are generally clearly defined. For example, the number of employees and profits are obvious examples of an input and an output, respectively.

As pointed out by Cook et al. (2014), although DEA has a strong link to production theory in economics, the tool here is intended for benchmarking, where a set of measures is selected to benchmark the performance of manufacturing and service operations. In the circumstance of benchmarking, the efficient DMUs, as defined by DEA, may not necessarily form a "production frontier", but rather lead to a "best-practice frontier". For example, if one benchmarks the performance of computers, it is natural to consider different features (screen size and resolution, memory size, process speed, hard disk size, and others). One would then have to classify these features into "inputs" and "outputs" in order to apply a proper DEA analysis. However, these features may not actually represent inputs and outputs at all, in the standard notion of production. In fact, if one examines the benchmarking literature, other terms, such as "indicators" and "outcomes", are used. The issue now becomes one of how to classify these performance measures into inputs and outputs, for use in DEA.

One could use a different notions, e.g., "Category I" and "Category II" measures for DEA "inputs" and "output", respectively. Since "inputs" and "outputs" have been used as the standard notions in DEA, the current book will not make such a change of notions. The reader should, however, understand that DEA "inputs" and "output" are just classifications of performance measures.

In general, DEA minimizes "inputs" and maximizes "outputs"; in other words, smaller levels of the former and larger levels of the latter represent better performance or efficiency. This can then be a rule for classifying factors under these two headings. There are, however, exceptions to this; for example, pollutants from a production process are outputs, yet higher levels of these indicate worse performance. There are DEA models that deal with such *undesirable* outputs (see, e.g., Seiford and Zhu 2002; Liu et al. 2010).

In certain circumstances, a factor can play a dual role of input and output simultaneously. For example, when evaluating the efficiencies of a set of universities, if one considers the numbers of Ph.D. students trained as outcomes from the education process, then this factor can rightly be viewed as an output. At the same time, however, Ph.D. students assist in carrying out research, and can therefore be viewed as a resource, hence an input to the process. See Cook et al. (2006). In such cases, the user must clearly define the purpose of benchmarking so that such performance measures can be classified as inputs or outputs. In some situations, the DMUs may have internal structures, e.g., a two-stage process. For example, banks generate deposits as an output in the first stage, and then the deposits are used as an input to generate profit in the second stage. In this case, "deposits" is treated as both output (from the first stage) and input (to the second stage).

In summary, if the underlying DEA problem represents a form of "production process", then "inputs" and "outputs" can often be more clearly identified. The resources used or required are usually the inputs and the outcomes are the outputs. If, however, the DEA problem is a general benchmarking problem, then the inputs are usually the "less-the-better" type of performance measures and the outputs are usually the "more-the-better" type of performance measures. The latter case is particularly relevant to the situations where DEA is employed as a MCDM (multiple criteria decision making) tool (see, e.g., Belton and Vickers 1993; Doyle and Green 1993; Stewart 1996).

Cook et al. (2014) also point out that we can also have a mixed use of ratio data, percentage data, and raw data, as inputs and outputs. Interested reader is referred to Cook et al. (2014) for other issues prior to choosing a DEA model.

1.4 Number of DMUs vs Number of Inputs and Outputs

It is well known that large numbers of inputs and outputs compared to the number of DMUs may diminish the discriminatory power of DEA. A suggested "rule of thumb" is that the number of DMUs be at least twice the number of inputs and outputs combined (see Golany and Roll 1989). Banker et al. (1989) on the other hand state that the number of DMUs should be at least three times the number of inputs and outputs combined. However, such a rule is neither imperative, nor does it have a statistical basis, but rather is often imposed for convenience. Otherwise, it is true that one loses discrimination power. It is not suggested, however, that such a rule is one that must be satisfied. There are situations where a significant number of DMUs are in fact efficient. In some cases the population size is small and does not permit one to add actual DMUs beyond a certain point. However, if the user wishes to reduce the number or proportion of efficient DMUs, various DEA models can help; for example, weight restrictions may be useful in such cases.

Cook et al. (2014) point out that while in statistical regression analysis, sample size can be a critical issue, as it tries to estimate the average behavior of a set of DMUs, DEA when used as a benchmarking tool, focuses on individual DMU performance. In that sense, the size of the sample or the number of DMUs under evaluation may be immaterial. For example, if there are only 10 firms in a particular market and if a large number of inputs and outputs have to be used if deemed necessary by the management, then the benchmarking results obtained from DEA can still be of value. One fact remains, namely that whatever form the production frontier takes, it is beyond the best practice frontier. It is also true that if one adds an additional DMU to an existing set, that DMU will be either inefficient or efficient. In the former case, the best practice frontier does not shift, and nothing new is learned about the production frontier. In the latter situation, the frontier may shift closer to the actual (but unknown) production frontier.

In summary, DEA is not a form of regression model, but rather it is a frontier-based linear programming-based optimization technique. It is meaningless to apply a sample size requirement to DEA, which should be viewed as a benchmarking tool focusing on individual performance. It is likely that a significant portion of DMUs will be deemed as efficient, if there are “too many” inputs and outputs given the number of DMUs. If the goal is to obtain fewer efficient DMUs, then one can use weight restrictions or other DEA approaches to reduce the number of efficient DMUs. See, e.g., Allen et al. (1997), Cooper et al. (2011b), Thompson et al. (1990), Chaps. 4–7 and 12.

1.5 Measuring and Managing Performance

If the baseball world of “moneyball” has taught us anything, it is that we cannot always trust our eyes. An overweight pitcher with an unorthodox delivery might provide exceptional value to a team, whereas an outfielder with a great throwing arm and fast bat speed might be a relatively poor investment. Similarly, savvy business managers know that their intuition can often be misleading, if not downright incorrect. *Employees who seem to work the least can often be the most productive. Business units that boast high profitability can sometimes be the least efficient.* This precisely describes what DEA does in performance benchmarking.

In a Sloan Management Review article by Sherman and Zhu (2013), the authors describe DEA as a “balanced benchmarking” that enables companies to benchmark and locate best practices that are not visible through other commonly used management methodologies. Today, DEA can be utilized by anyone with Microsoft *Excel* (the current book offers a version of DEAFrontier software—a Microsoft Excel Add-In), but it was not always so easy.

As documented in Sherman and Zhu (2013), DEA provides managers with a sophisticated mechanism to assess the performance of different service providers—comparing, for example, the London and Tokyo offices of a global advertising agency—by going well beyond the crude metrics and ratios such as profitability and account billings per employee. From the results of DEA, a company can identify its least efficient offices or business units, and it can assess the magnitude of the inefficiency and investigate potential paths for improvement that the analysis has identified. Moreover, executives can study the top-performing units, identify the best practices, and transfer that valuable knowledge throughout the organization to enhance performance. Lastly, DEA enables companies to test their assumptions, particularly before implementing cost-cutting initiatives that might inadvertently be counter-productive.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_1) contains supplementary material, which is available to authorized users.

References

Allen, R., Athanassopoulos, A., Dyson, R. G., & Thanassoulis, E. (1997). Weights restrictions and value judgments in data envelopment analysis: Evolution, development and future directions. *Annals of Operations Research, 73,* 13–34.

Banker, R. D., Charnes, A., Cooper, W. W., Swarts, J., & Thomas, D. (1989). An introduction to data envelopment analysis with some of its models and their uses. *Research in Government and Nonprofit Accounting, 5,* 125–163.

Belton, V., & Vickers, S. P. (1993). Demystifying DEA: A visual interactive approach based on multi criteria analysis. *Journal of the Operational Research Society, 44*(9), 883–896.

Camp, R. C. (1995). *Business process benchmarking, finding and implementing best practices*. Milwaukee: ASQC Quality Press.

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring efficiency of decision making units. *European Journal of Operational Research, 2,* 429–444.

Charnes, A., Cooper, W. W., & Rhodes, E. (1981). Evaluating program and managerial efficiency: An application of data envelopment analysis to program follow through. *Management Science, 27*(6), 668–697.

Cook, W. D., & Zhu, J. (2014). *Data envelopment analysis—A handbook of modeling internal structures and networks*. New York: Springer.

Cook, W. D., Green, R., & Zhu, J. (2006). Dual—Role factors in DEA. *IIE Transactions, 38,* 1–11.

Cook, W. D., Tone, K., & Zhu, J. (2014). Data envelopment analysis: Prior to choosing a model. *OMEGA, 44,* 1–4.

Cooper, W. W., Seiford, L. M., & Zhu, J. (2011a). *Handbook on data envelopment analysis* (2nd ed.). New York: Springer.

Cooper, W. W., Ruiz, J. L., & Sirvent, I. (2011b). Choice and uses of DEA weights. In W. W. Cooper, L. M. Seiford, & J. Zhu (Eds.), *Handbook on data envelopment analysis*. New York: Springer.

Doyle, J., & Green, R. (1993). Data envelopment analysis and multiple criteria decision making. *Omega, 21*(6), 713–715.

Golany, B., & Roll, Y. (1989). An application procedure for DEA. *OMEGA, 17*(3), 237–250.

Johnson, S. A., & Zhu, J. (2002). Identifying top applicants in recruiting using data envelopment analysis. *Socio-Economic Planning Sciences, 37,* 125–139.

Liu, W. B., Meng, W., Li, X. X., & Zhang, D. Q. (2010). DEA models with underirable inputs and outputs. *Annals of Operations Research, 173,* 177–194.

Liu, J. S., Lu, L. Y. Y., Lu, W.-M., & Lin, B. J. Y. (2013). A survey of DEA applications. *OMEGA, 41*(5), 893–902.

Seiford, L. M., & Zhu, J. (2002). Modeling undesirable factors in efficiency evaluation. *European Journal of Operational Research, 142*(1), 16–20.

Stewart, T. J. (1996). Relationships between data envelopment analysis and multicriteria decision analysis. *The Journal of the Operational Research Society, 47*(5), 654–665.

Sherman, H. D. (1984). Improving the productivity of service businesses. *Sloan Management Review, 35*(3), 11–23.

Sherman, D. H., & Zhu, J. (2006). *Service productivity management: Improving service performance using data envelopment analysis.* Boston: Springer Science.

Sherman, H. D., & Zhu, J. (2013). Analyzing performance in service organizations. *Sloan Management Review, 54*(4, Summer 2013), 37–42.

Shwartz, M., Burgess, J. F., Jr., & Berlowitz, D. (2009). Benefit-of-the-doubt approaches for calculating a composite measure of quality. *Health Services and Outcomes Research Methodology, 9,* 234–251.

Thompson, R. G., Langemeir, L. N., Lee, C., Lee, E., & Thrall, R. M. (1990). The role of multiplier bounds in efficiency analysis with application to Kanssa farming. *Journal of Econometrics, 1*(2), 93–108.

Zhu, J. (2015). *Data envelopment analysis—A handbook of models and methods.* New York: Springer.

Chapter 2
Envelopment DEA Models

2.1 Introduction

This chapter presents some basic DEA models that are used to determine the best-practice frontier characterized by (Sect. 1.1) in Chap. 1. These models are called envelopment models, because the identified best-practice frontier envelops all the observations (DMUs). The shapes of best-practice (or efficient) frontiers obtained from these models can be associated with the concept of Returns-to-Scale (RTS) which will be discussed in details in Chap. 13. This is because the best-practice (or efficient) frontiers can be viewed as exhibiting of various types of RTS. However, if the inputs and outputs are not related to a "production function", RTS concept cannot be applied. Under such cases, RTS is merely used to refer to different shapes of frontiers.

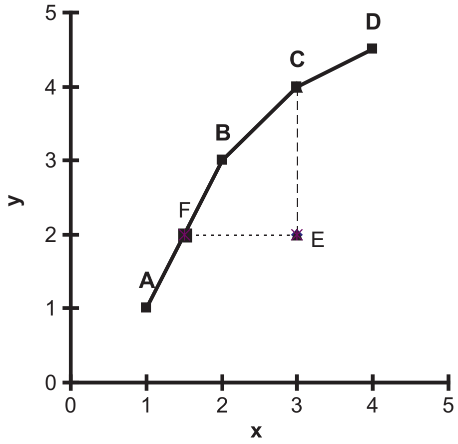
Consider Fig. 2.1 where we have 5 DMUs (A, B, C, D, and E) with one input and one output. One possible best-practice frontier consists of DMUs A, B, C, and D. AB exhibits increasing RTS (IRS), B exhibits constant RTS (CRS), and BC and CD exhibit decreasing RTS (DRS). As a result, this best-practice frontier is called Variable RTS (VRS) frontier.

DMU E is not efficient (or best-practice), because it uses too much input and/or it does not produce enough output. In fact, there are two ways to improve the performance of E. One is to reduce its input to reach F on the frontier, and the other to increase its output to reach C on the frontier. As a result, DEA models will have two orientations: input-oriented and output-oriented.

Input-oriented models are used to test if a DMU under evaluation can reduce its inputs while keeping the outputs at their current levels. Output-oriented models are used to test if a DMU under evaluation can increase its outputs while keeping the inputs at their current levels.

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_2,

Fig. 2.1 Variable returns-to-scale (VRS) frontier



2.2 Variable Returns-to-Scale (VRS) Model

The following DEA model is an input-oriented model where the inputs are minimized and the outputs are kept at their current levels (Banker et al. 1984)

$$\begin{aligned}
&\theta^* = \min \theta \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le \theta x_{io} \quad i = 1,2,\dots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} \quad r = 1,2,\dots,s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \ge 0 \quad j = 1,2,\dots,n.
\end{aligned} \tag{2.1}$$

where DMU_o represents one of the n DMUs under evaluation, and x_{io} and y_{ro} are the ith input and rth output for DMU_o, respectively.

Since $\theta=1$ is a feasible solution to (1.2), the optimal value to (2.1), $\theta^* \le 1$. If $\theta^* = 1$, then the current input levels cannot be reduced (proportionally), indicating that DMU_o is on the frontier. Otherwise, if $\theta^* < 1$, then DMU_o is dominated by the frontier. θ^* represents the (input-oriented) efficiency score of DMU_o.

Consider a simple numerical example shown in Table 2.1 where we have five DMUs (supply chain operations). Within a week, each DMU generates the same profit of \$ 2,000 with a different combination of supply chain cost and response time.

Table 2.1 Supply chain operations within a week

DMU	Cost ($ 100)	Response time (days)	Profit ($ 1,000)
1	1	5	2
2	2	2	2
3	4	1	2
4	6	1	2
5	4	4	2

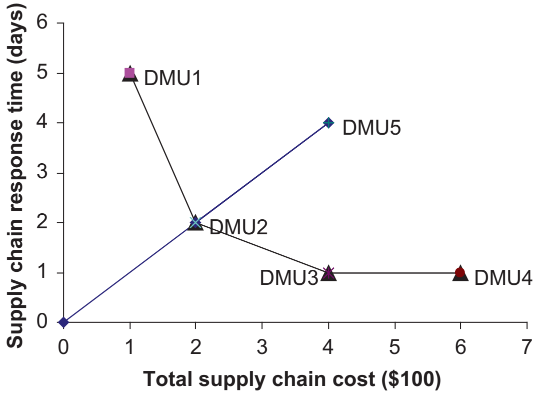


Fig. 2.2 Five supply chain operations

Figure 2.2 presents the five DMUs and the piecewise linear frontier. DMUs 1, 2, 3, and 4 are on the frontier. If we calculate model (2.1) for DMU5,

$$
\begin{aligned}
&\text{Min } \theta \\
&\text{Subject to} \\
&1\,\lambda_1 + 2\lambda_2 + 4\lambda_3 + 6\lambda_4 + 4\lambda_5 \le 4\theta \\
&5\,\lambda_1 + 2\lambda_2 + 1\lambda_3 + \ 1\lambda_4 + 4\lambda_5 \le 4\theta \\
&2\,\lambda_1 + 2\lambda_2 + 2\lambda_3 + 2\lambda_4 + 2\lambda_5 \ge 2 \\
&\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 = 1 \\
&\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 \ge 0
\end{aligned}
$$

we obtain a set of unique optimal solutions of $\theta^* = 0.5$, $\lambda_2^* = 1$ and $\lambda_j^* = 0\ (j \ne 2)$, indicating that DMU2 is the benchmark for DMU5, and DMU5 should reduce its cost and response time to the amounts used by DMU2.

Now, if we calculate model (2.1) for DMU4, we obtain $\theta^* = 1$, $\lambda_4^* = 1$, and $\lambda_j^* = 0\ (j \ne 4)$, indicating that DMU4 is on the frontier. However, Fig. 2.2 indicates that DMU4 can still reduce its total supply chain cost by $ 200 to reach DMU3. This individual input reduction is called input slack.

In fact, both input and output slack values may exist in model (2.1). Usually, after calculating (2.1), we have

$$\begin{cases} s_i^- = \theta^* x_{io} - \sum_{j=1}^{n} \lambda_j x_{ij} & i = 1,2,...,m \\ s_r^+ = \sum_{j=1}^{n} \lambda_j y_{rj} - y_{ro} & r = 1,2,...,s \end{cases} \tag{2.2}$$

where s_i^- and s_r^+ represent input and output slacks, respectively. An alternate optimal solution of $\theta^* = 1$ and $\lambda_3^* = 1$ exists when we calculate model (2.1) for DMU4. This leads to $s_1^- = 2$ for DMU4. However, if we obtain $\theta^* = 1$ and $\lambda_4^* = 1$ from model (2.1), we have all zero slack values. i.e., because of possible multiple optimal solutions, (2.2) may not yield all the non-zero slacks.

Therefore, we use the following linear programming model to determine the possible non-zero slacks after (2.1) is solved.

$$\begin{aligned} & \max \sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+ \\ & \text{subject to} \\ & \sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = \theta^* x_{io} \qquad i = 1,2,...,m; \\ & \sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \qquad r = 1,2,...,s; \\ & \sum_{j=1}^{n} \lambda_j = 1 \\ & \lambda_j \geq 0 \qquad j = 1,2,...,n. \end{aligned} \tag{2.3}$$

For example, applying (2.3) to DMU4 yields

$$\begin{aligned} & \text{Max } s_1^- + s_2^- + s_1^+ \\ & \text{Subject to} \\ & 1\lambda_1 + 2\lambda_2 + 4\lambda_3 + 6\lambda_4 + 4\lambda_5 + s_1^- = 6\theta^* = 6 \\ & 5\lambda_1 + 2\lambda_2 + 1\lambda_3 + 1\lambda_4 + 4\lambda_5 + s_2^- = \theta^* = 1 \\ & 2\lambda_1 + 2\lambda_2 + 2\lambda_3 + 2\lambda_4 + 2\lambda_5 - s_1^+ = 2 \\ & \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 = 1 \\ & \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, s_1^-, s_2^-, s_1^+ \geq 0 \end{aligned}$$

with optimal slacks of $s_1^{-*} = 2$, $s_2^{-*} = s_1^{+*} = 0$.

DMU_o is efficient if and only if $\theta^* = 1$ and $s_i^{-*} = s_r^{+*} = 0$ for all *i* and *r*. DMU_o is weakly efficient if $\theta^* = 1$ and $s_i^{-*} \neq 0$ and (or) $s_r^{+*} \neq 0$ for some *i* and *r*. In Fig. 2.2, DMUs 1, 2, and 3 are efficient, and DMU 4 is weakly efficient.

Definition 2.1 The slacks obtained by (2.3) are called DEA slacks. Or specifically, slacks calculated from a second-stage DEA calculation are called DEA slacks.

In fact, models (2.1) and (2.3) represent a two-stage DEA process involved in the following DEA model.

$$
\begin{aligned}
&\min \theta - \varepsilon\left(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+\right) \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = \theta x_{io} \qquad i = 1,2,\dots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \qquad r = 1,2,\dots,s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \ge 0 \qquad j = 1,2,\dots,n.
\end{aligned} \tag{2.4}
$$

The presence of the non-Archimedean ε in the objective function of (2.4) effectively allows the minimization over θ to preempt the optimization involving the slacks, s_i^- and s_r^+. Thus, (2.4) is calculated in a two-stage process with maximal reduction of inputs being achieved first, via the optimal θ^* in (2.1); then, in the second stage, movement onto the efficient frontier is achieved via optimizing the slack variables in (2.3). It is incorrect if one attempts to solve model (2.4) in a single model/stage by specifying an ε value in the objective function of (2.4).

In fact, the presence of weakly efficient DMUs is the cause of multiple optimal solutions. Thus, if weakly efficient DMUs are not present, the second stage calculation (2.3) is not necessary, and we can obtain the slacks using (2.2). However, priori to calculation, we usually do not know whether weakly efficient DMUs are present or not.

Note that the frontier determined by model (2.1) exhibits variable returns to scale (VRS). Therefore, model (2.1) is called input-oriented VRS envelopment model. (see Chap. 13 for a detailed discussion on DEA and Returns-to-Scale (RTS).)

The output-oriented VRS envelopment model can be expressed as

$$
\begin{aligned}
&\max \phi + \varepsilon\left(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+\right) \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \qquad i = 1,2,\dots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = \phi y_{ro} \qquad r = 1,2,\dots,s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \ge 0 \qquad j = 1,2,\dots,n.
\end{aligned} \tag{2.5}
$$

Model (2.5) is also calculated in a two-stage process. (One should never try to solve model (2.5) in a single model by specifying an ε value in the objective function of (2.5)).

First, we calculate ϕ^* by ignoring the slacks, namely,

$$\begin{aligned}
&\max \phi \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge \phi y_{ro} \quad r = 1,2,\ldots,s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \ge 0 \qquad\qquad j = 1,2,\ldots,n.
\end{aligned}$$

Then we optimize the slacks by fixing the ϕ^* in the following linear programming problem.

$$\begin{aligned}
&\max \sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+ \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = \phi^* y_{ro} \quad r = 1,2,\ldots,s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \ge 0 \qquad\qquad j = 1,2,\ldots,n.
\end{aligned} \tag{2.6}$$

DMU_o is efficient if and only if $\phi^* = 1$ and $s_i^{-*} = s_r^{+*} = 0$ for all i and r. DMU_o is weakly efficient if $\phi^* = 1$ and $s_i^{-*} \neq 0$ and (or) $s_r^{+*} \neq 0$ for some i and r. If weakly efficient DMUs are not present, then we need not to calculate (2.6), and we can obtain the slacks via

$$\begin{cases}
s_i^- = x_{io} - \sum_{j=1}^{n} \lambda_j x_{ij} & i = 1,2,\ldots,m \\
s_r^+ = \sum_{j=1}^{n} \lambda_j y_{rj} - \phi^* y_{ro} & r = 1,2,\ldots,s
\end{cases}$$

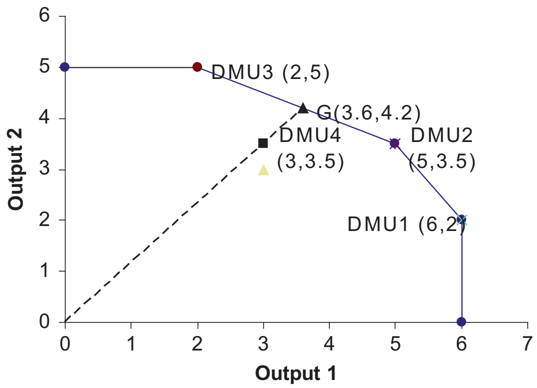


Fig. 2.3 Output efficient frontier

Figure 2.2 shows an input efficient frontier when outputs are fixed at their current levels. Similarly, we can obtain an output efficient frontier when inputs are fixed at their current levels. Consider the four DMUs shown in Fig. 2.3 where we have two outputs.

In Fig. 2.3, DMUs 1, 2 and 3 are efficient. If we calculate model (2.5) for DMU4, we have

$$
\begin{aligned}
&\text{Max}\,\phi\\
&\text{Subject to}\\
&\lambda_1+\lambda_2+\lambda_3+\lambda_4 \le 1\\
&6\lambda_1 + 5\lambda_2 + 2\lambda_3 + 3\lambda_4 \ge 3\phi\\
&2\lambda_1 + 3.5\lambda_2 + 5\lambda_3 + 3.5\lambda_4 \ge 3.5\phi\\
&\lambda_1+\lambda_2+\lambda_3+\lambda_4 = 1\\
&\lambda_1,\lambda_2,\lambda_3,\lambda_4 \ge 0
\end{aligned}
$$

The optimal solution is $\phi^*=1.2$, $\lambda_2^* = 8/15$, and $\lambda_3^* = 7/15$. i.e., DMU4 is inefficient and is compared to G in Fig. 2.3, or DMU4 should increase its two output levels to G. In this case, if we calculate model (2.6), all slacks will be zero.

2.3 DEA Slacks

In this section, we provide two numerical examples to further show the concept of DEA slacks calculated from the second stage DEA calculation in model (2.3) or (2.6).

Consider Fig. 2.4 with one input and one output and input-oriented VRS model. Any DMUs in the shaded area will have an output DEA slack after it is moved onto the VRS frontier by input reduction. For example, DMU H is moved onto G which is a frontier point. However, we can still increase G's output to point A.

Fig. 2.4 DEA slack under input orientation

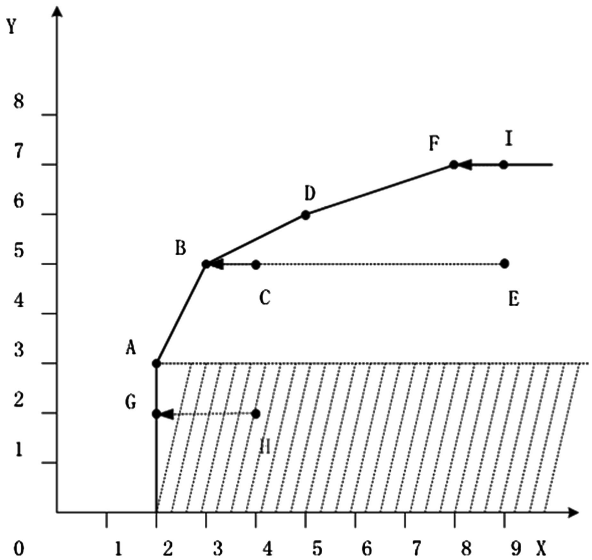
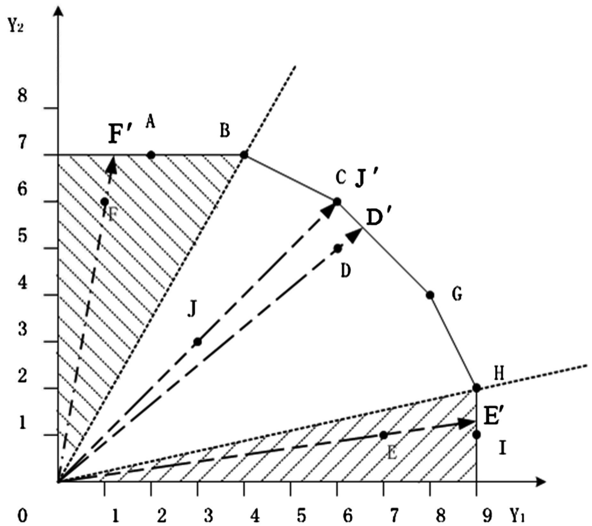


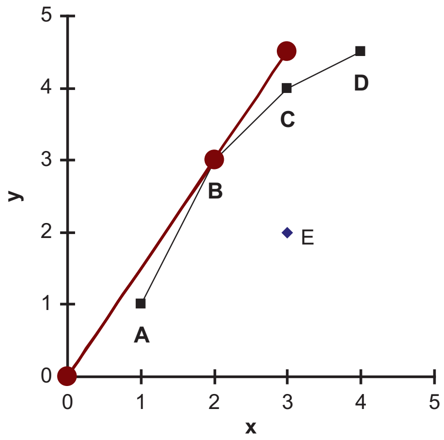
Fig. 2.5 DEA slack under output orientation



DMUs I and E are inefficient under the input-oriented VRS model. Note that DMU I is actually on the VRS frontier, and is a weakly efficient DMU under the output-oriented VRS model. Because under model (2.5), the efficiency score for DMU I is equal to one. Yet, DMU I can further reduce its input to point F.

We next consider Fig. 2.5 where we have two outputs. Under the output-oriented VRS model, all DMUs (e.g., E and F) within the shaded areas will have (non-zero) DEA slack values. DMUs A and I are weakly efficient. DMUs J and D are

Fig. 2.6 CRS frontier



inefficient and do not have DEA slacks, as their outputs are proportionally increased to J′ and D′, respectively.

2.4 Other Envelopment Models

The constraint on $\sum_{j=1}^{n}\lambda_j$ in model (2.1) actually determines the RTS type of an efficient frontier. If we remove $\sum_{j=1}^{n}\lambda_j = 1$ from models (2.1) and (2.5), we obtain CRS (Constant RTS) envelopment models where the frontier exhibits CRS (Charnes et al. 1978). Figure 2.6 shows a CRS frontier—ray OB. Based upon this CRS frontier, only B is efficient.

If we replace $\sum_{j=1}^{n}\lambda_j = 1$ with $\sum_{j=1}^{n}\lambda_j \le 1$, then we obtain non-increasing RTS (NIRS) envelopment models. In Fig. 2.7, the NIRS frontier consists of DMUs B, C, D and the origin.

If we replace $\sum_{j=1}^{n}\lambda_j = 1$ with $\sum_{j=1}^{n}\lambda_j \ge 1$, then we obtain non-decreasing RTS (NDRS) envelopment models. In Fig. 2.8, the NDRS frontier consists of DMUs, A, B, and the section starting with B on ray OB.

Table 2.2 summarizes the envelopment models with respect to the orientations and frontier types. The last row presents the efficient target (DEA projection) of a specific DMU under evaluation.

The interpretation of the envelopment model results can be summarized as

I. If $\theta^* = 1$ or $\phi^* = 1$, then the DMU under evaluation is a frontier point. i.e., there is no other DMUs that are operating more efficiently than this DMU. Otherwise, if $\theta^* < 1$ or $\phi^* > 1$, then the DMU under evaluation is inefficient. i.e., this DMU can either increase its output levels or decrease its input levels.
II. The left-hand-side of the envelopment models is usually called the "Reference Set", and the right-hand-side represents a specific DMU under evaluation. The

Fig. 2.7 NIRS frontier

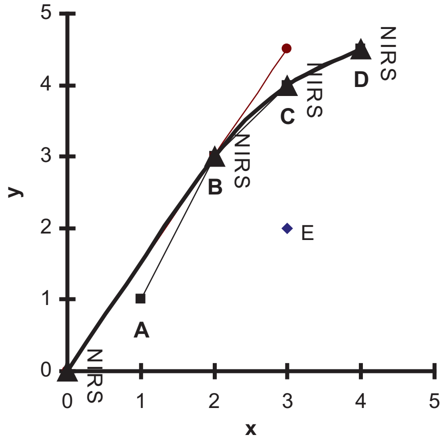
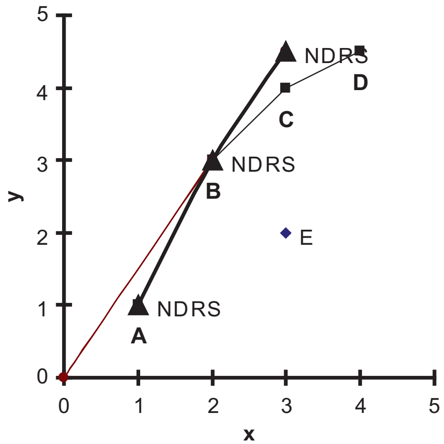


Fig. 2.8 NDRS frontier



non-zero optimal λ_j^* represent the benchmarks for a specific DMU under evaluation. The Reference Set provides coefficients (λ_j^*) to define the hypothetical efficient DMU. The Reference Set or the efficient target shows how inputs can be decreased and outputs increased to make the DMU under evaluation efficient.

III. The "Efficient Target" in Table 2.2 is a result of two stage DEA calculation. However, sometimes a DEA user may ignore the second stage slack calculation and is only interested in the efficiency scores. In that case, $\theta^* x_{io}$ or $\phi^* y_{ro}$ can be regarded as Target on the frontier.

Table 2.2 Envelopment models

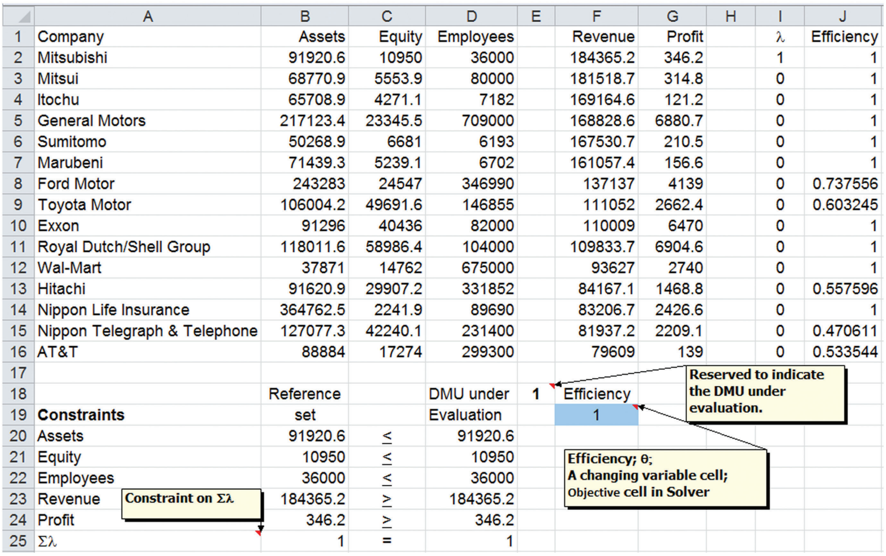
Frontier type	Input-oriented	Output-oriented
CRS	$\min \theta - \varepsilon(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+)$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = \theta x_{io} \quad i = 1,2,\ldots,m;$ $\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \quad r = 1,2,\ldots,s;$ $\lambda_j \geq 0 \quad j = 1,2,\ldots,n.$	$\max \phi + \varepsilon(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+)$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \quad i = 1,2,\ldots,m;$ $\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = \phi y_{ro} \quad r = 1,2,\ldots,s;$ $\lambda_j \geq 0 \quad j = 1,2,\ldots,n.$
VRS	Add $\sum_{j=1}^{n} \lambda_j = 1$	
NIRS	Add $\sum_{j=1}^{n} \lambda_j \leq 1$	
NDRS	Add $\sum_{j=1}^{n} \lambda_j \geq 1$	
Efficient target	$\begin{cases} \hat{x}_{io} = \theta^* x_{io} - s_i^{-*} & i = 1,2,\ldots,m \\ \hat{y}_{ro} = y_{ro} + s_r^{+*} & r = 1,2,\ldots,s \end{cases}$	$\begin{cases} \hat{x}_{io} = x_{io} - s_i^{-*} & i = 1,2,\ldots,m \\ \hat{y}_{ro} = \varphi^* y_{ro} + s_r^{+*} & r = 1,2,\ldots,s \end{cases}$

Table 2.3 Fortune global 500 companies. (Source: Fortune magazine 1995)

Company	Assets	Equity	Employees	Revenue	Profit
Mitsubishi	91920.6	10950	36000	184365.2	346.2
Mitsui	68770.9	5553.9	80000	181518.7	314.8
Itochu	65708.9	4271.1	7182	169164.6	121.2
General motors	217123.4	23345.5	709000	168828.6	6880.7
Sumitomo	50268.9	6681	6193	167530.7	210.5
Marubeni	71439.3	5239.1	6702	161057.4	156.6
Ford motor	243283	24547	346990	137137	4139
Toyota motor	106004.2	49691.6	146855	111052	2662.4
Exxon	91296	40436	82000	110009	6470
Royal Dutch/Shell group	118011.6	58986.4	104000	109833.7	6904.6
Wal-Mart	37871	14762	675000	93627	2740
Hitachi	91620.9	29907.2	331852	84167.1	1468.8
Nippon life insurance	364762.5	2241.9	89690	83206.7	2426.6
Nippon telegraph & telephone	127077.3	42240.1	231400	81937.2	2209.1
AT&T	88884	17274	299300	79609	139

2.5 Envelopment Models in Spreadsheets

Table 2.3 presents 15 companies from the top Fortune Global 500 list in 1995. We have three inputs: (1) number of employees, (2) assets ($ millions), and (3) equity ($ millions), and two outputs: (1) profit ($ millions), and (2) revenue ($ millions).



	A	B	C	D	E	F	G	H	I	J
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	Efficiency
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		1	1
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0	1
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	1
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0	1
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0	1
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	1
8	Ford Motor	243283	24547	346990		137137	4139		0	0.737556
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	0.603245
10	Exxon	91296	40436	82000		110009	6470		0	1
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	1
12	Wal-Mart	37871	14762	675000		93627	2740		0	1
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	0.557596
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0	1
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0	0.470611
16	AT&T	88884	17274	299300		79609	139		0	0.533544
17										
18		Reference		DMU under	1	Efficiency				
19	**Constraints**	set		Evaluation		1				
20	Assets	91920.6	≤	91920.6						
21	Equity	10950	≤	10950						
22	Employees	36000	≤	36000						
23	Revenue	184365.2	≥	184365.2						
24	Profit	346.2	≥	346.2						
25	Σλ	1	=	1						

Fig. 2.9 Input-oriented VRS envelopment spreadsheet model

2.5.1 *Input-Oriented VRS Envelopment Spreadsheet Model*

The input-oriented VRS envelopment model (model 2.1) requires 15 calculations—one for each company (DMU). We illustrate how to formulate this efficiency evaluation problem in a spreadsheet, and then illustrate how Excel Solver can be used to calculate the efficiency scores for the 15 companies.

We begin by organizing the data in Table 2.3 in a spreadsheet (see Fig. 2.9). A spreadsheet model of an envelopment model contains the following four major components: (1) cells for the decision variables (e.g., λ_j and θ); (2) cell for the objective function (efficiency) (e.g., θ); (3) cells containing formulas for computing the DEA reference set (the left-hand-side of the constraints) $\left(\sum_{j=1}^{n} \lambda_j x_{ij}, \sum_{j=1}^{n} \lambda_j y_{rj}, \text{ and } \sum_{j=1}^{n} \lambda_j\right)$; and (4) cells containing formulas for computing the DMU under evaluation (right-hand-sided of the constraints) (e.g., θx_{io} and y_{ro}).

In Fig. 2.9, cells I2 through I16 represent λ_j ($j=1$, 2, …, 15). Cell F19 represents the efficiency score θ which is the objective function.

For the DEA reference set (left-hand-side of the envelopment model), we enter the following formulas that calculate the weighted sums of inputs and outputs across all DMUs, respectively.

Cell B20 =SUMPRODUCT(B2:B16,$I$2:$I$16)
Cell B21 =SUMPRODUCT(C2:C16,$I$2:$I$16)
Cell B22 =SUMPRODUCT(D2:D16,$I$2:$I$16)
Cell B23 =SUMPRODUCT(F2:F16,$I$2:$I$16)
Cell B24 =SUMPRODUCT(G2:G16,$I$2:$I$16)

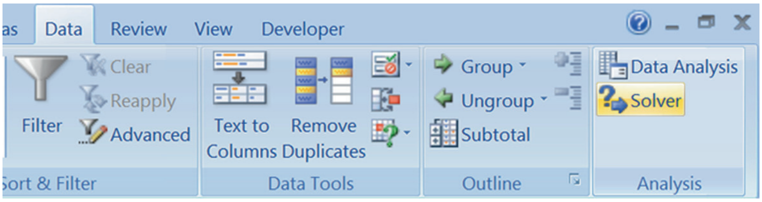


Fig. 2.10 Excel solver

For the DMU under evaluation (DMU1: Mitsubishi), we enter the following formulas into cells D20:D24.

Cell D20 =$F$19*INDEX(B2:B16,E18,1)
Cell D21 =$F$19*INDEX(C2:C16,E18,1)
Cell D22 =$F$19*INDEX(D2:D16,E18,1)
Cell D23 =INDEX(F2:F16,E18,1)
Cell D24 =INDEX(G2:G16,E18,1)

Finally, we enter the formula for $\sum_{j=1}^{n} \lambda_j = 1$ into cells B25 (=SUM(I2:I16)) and D25 (=1), respectively.

Cell E18 is reserved to indicate the DMU under evaluation. The function INDEX(array,row number,column number) returns the value in the specified row and column of the given array. Because cell E18 contains the current value of 1, the INDEX function in cell D23 returns the value in first row and first column of the Revenue array F2:F16 (or the value in cell F2, the Revenue output for DMU1). When the value in cell E18 changes from 1 to 15, the INDEX functions in cells D20:D24 return the input and output values for a specific DMU under evaluation. This feature becomes obvious and useful when we provide the Visual Basic for Applications (VBA) code to automate the DEA computation.

2.5.2 Using Solver

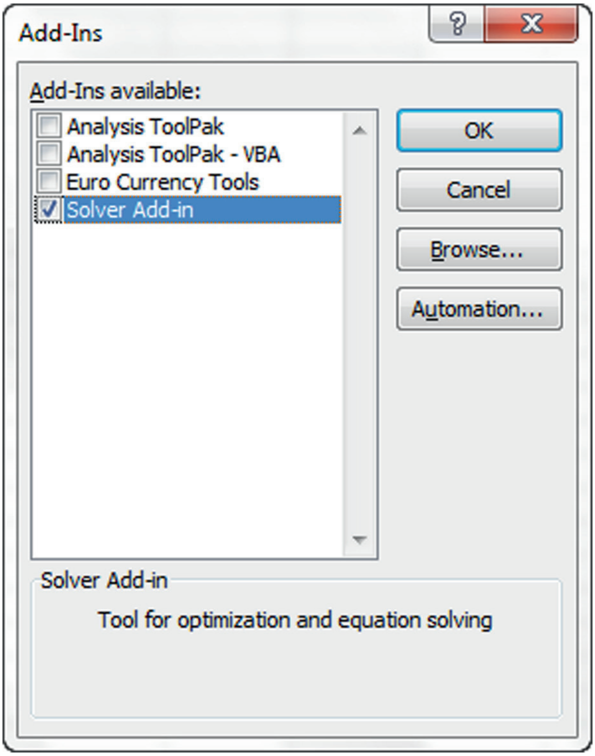
After the DEA model is set up in the spreadsheet, we can use Solver to find the optimal solutions. First, we need to invoke Solver in Excel by using the **Solver** command which is available in the **Data** tab as shown in Fig. 2.10.

If Solver does not exist, you need to do the followings:

1. Click **File** tab, and then click **Options**.
2. Click **Add-Ins**, and then in the **Manage** box, select **Excel Add-ins**.
3. Click **Go**.
4. In the **Add-Ins available** box, select the **Solver Add-in** check box, and then click **OK**. (see Fig. 2.11)

Now, you should see the Solver Parameters dialog box shown in Fig. 2.12.

Fig. 2.11 Solver add-in



2.5.3 Setting the Objective Cell and Changing Variable Cells

Set Objective cell indicates the objective function cell in the spreadsheet, and whether its value should be maximized or minimized. In our case, the objective cell is the DEA efficiency represented by cell F19, and its value should be minimized, because we use the input-oriented VRS envelopment model (2.1) (see Fig. 2.13).

Changing Variable Cells represent the decision variables in the spreadsheet. In our case, they represent the λ_j ($j=1,2, \ldots, 15$) and θ, and should be cells I2:I16 and F19, respectively (see Fig. 2.13).

2.5.4 Adding Constraints and Selecting Solving Method

Constraints represent the constraints in the spreadsheet. In our case, they are determined by cells B20:B25 and D20:D25. For example, click the Add button shown in Fig. 2.13, you will see the Add Constraint dialog box shown in Fig. 2.14.

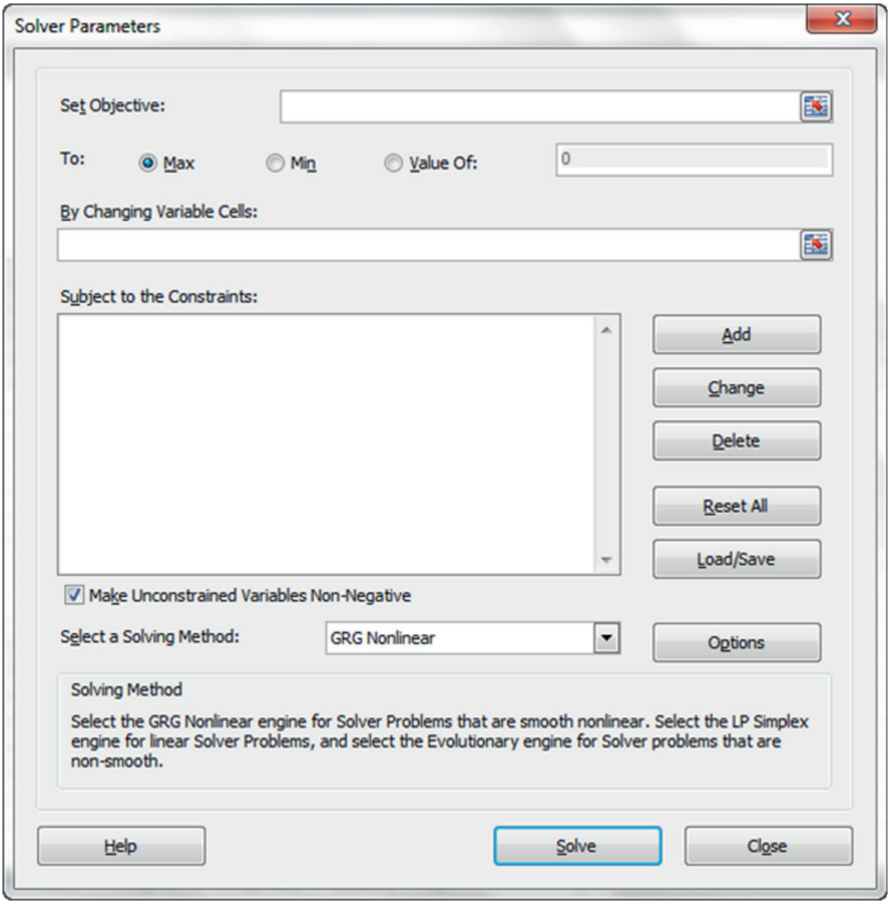


Fig. 2.12 Solver parameters dialog box

In the spreadsheet model shown in Fig. 2.9, we have six constraints. The "Cell Reference" corresponds to the DEA Reference Set, and "Constraint" corresponds to the DMU under evaluation. The first three constraints are related to the three inputs (see Fig. 2.14). Click the Add button to add additional constraints (output constraints and $\sum_{j=1}^{n} \lambda_j = 1$), and click the OK button when you have finished adding the constraints. The set of the constraints are shown in Fig. 2.15.

Note that λ_j and θ are all non-negative. This can be achieved by checking the option of "Make Unconstrained Variables Non-Negative". Since DEA models are linear models, we should select "Simplex LP" in solving method option, as shown in Fig. 2.15.

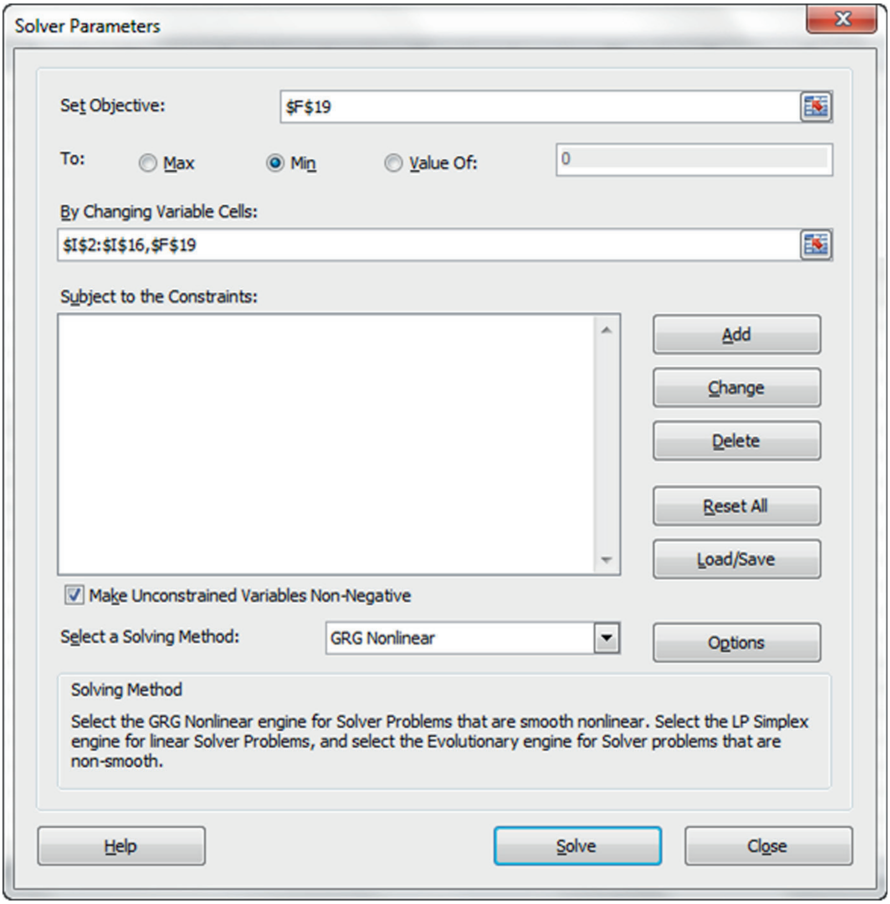
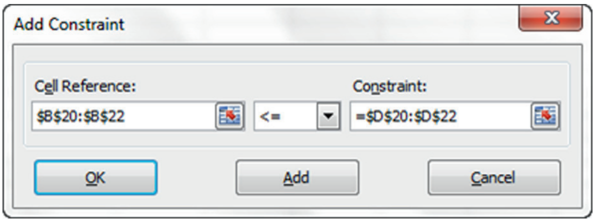


Fig. 2.13 Setting objective cell and changing cells

Fig. 2.14 Adding constraints



2.5.5 *Solving the Model*

Now, we have successfully set up the Solver Parameters dialog box, as shown in Fig. 2.15. Click the Solve button to solve the model. When Solver finds an optimal solution, it displays the Solver Results dialog box, as shown in Fig. 2.16.

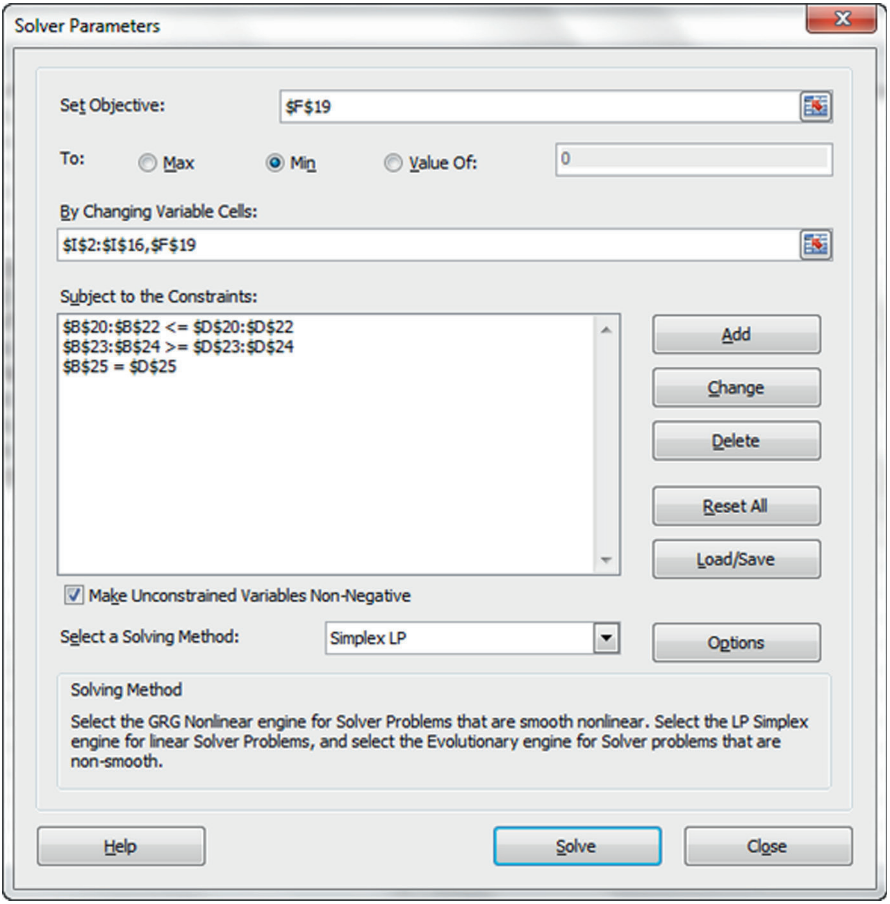


Fig. 2.15 Solver parameters for input-oriented VRS envelopment model

2.5.6 Automating the DEA Calculation

To complete the analysis for the remaining 14 companies, one needs to manually change the value in cell E18 to 2, 3, …, 15 and use Solver to re-optimize the spreadsheet model for each company and record the efficiency scores (in column J, for instance). When the number of DMUs becomes large, the manual process is apparently cumbersome. Note that exactly the same Solver settings will be used to find the optimal solutions for the remaining DMUs. This allows us to write a simple VBA code to carry out the process automatically.

Before we write the VBA code, we need to set a reference to Solver Add-In in Visual Basic (VB) Editor. Otherwise, VBA will not recognize the Solver functions and you will get a “Sub or function not defined” error message.

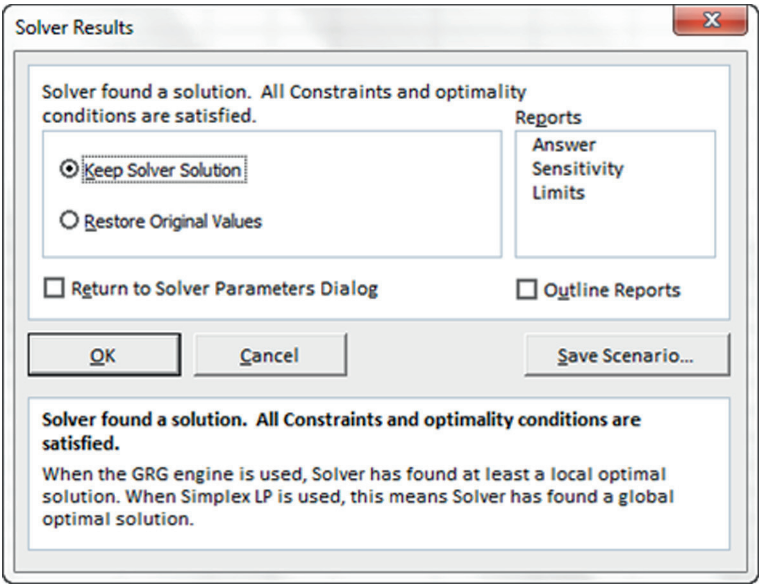


Fig. 2.16 Solver results dialog box

We may follow the following procedure to set the reference. Enter the VB Editor by using the Developer Tab. One must configure Excel to show the Developer tab because it does not appear by default. To show the Developer tab, follow the following steps.

1. On the **File** tab, choose the **Options** button
2. In the **Options** dialog box, choose the **Customize Ribbo**n button
3. In the list of main tabs, select the **Developer** check box
4. Choose the **OK** button to close the **Options** dialog box

Open the Tools/References menu in the VB Editor. This brings up a list of references. One of these should be Solver (**Solver.xlam)** (see Fig. 2.17). To add the reference, simply check its box.

After the Solver reference is added, we should see "Reference to Solver.xlam" under the "References" in the VBA Project Explorer window shown in Fig. 2.18. (The file "envelopment spreadsheet.xlsm" contains the spreadsheet model.)

Next, select the Insert/Module menu item in the VB Editor (Fig. 2.19). This action will add a Module (e.g., Module1) into the Excel file. (You can change the name of the inserted module in the Name property of the module.)

Now, we can insert the VBA code into the Module1. Type "Sub DEA()" in the code window. This generates a VBA procedure called DEA which is also the Macro name (see Fig. 2.21). Figure 2.20 shows the VBA code for automating the DEA calculation.

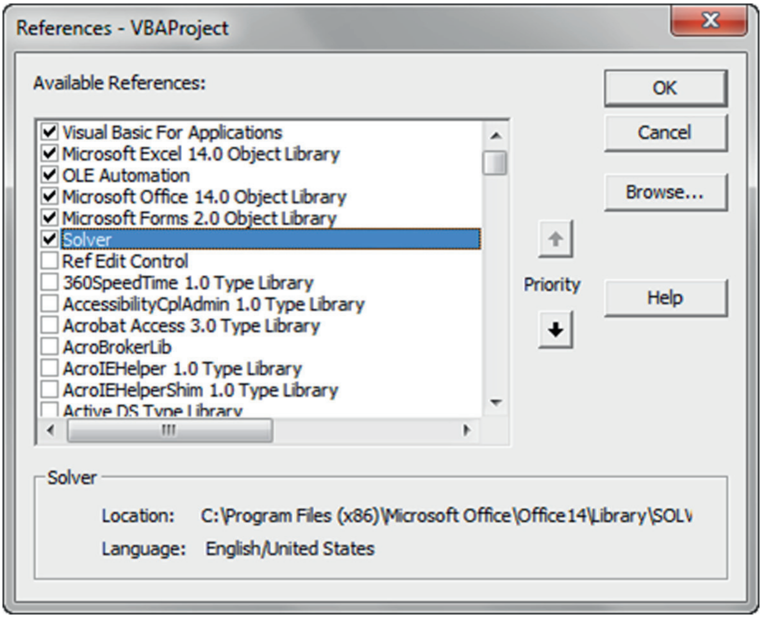


Fig. 2.17 Adding reference to solver Add-In

Fig. 2.18 Reference to solver add-in in VBA project

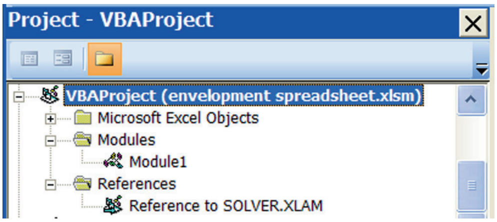
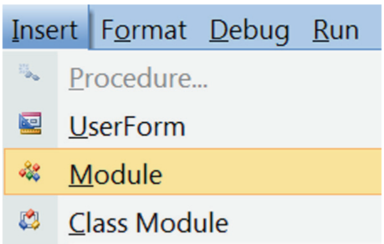


Fig. 2.19 Insert a module



The Macro statement "SolverSolve UserFinish:=True" tells the Solver to solve the DEA problem without displaying the Solver Results dialog box. The "Offset(*rowOffset*, *columnOffset*)" property takes two arguments that correspond to the relative position from the upper-left cell of the specified Range. When we evaluate the first DMU, i.e., DMUNo = 1, Range("J1").Offset(1,0) refers to cell J2. The state-

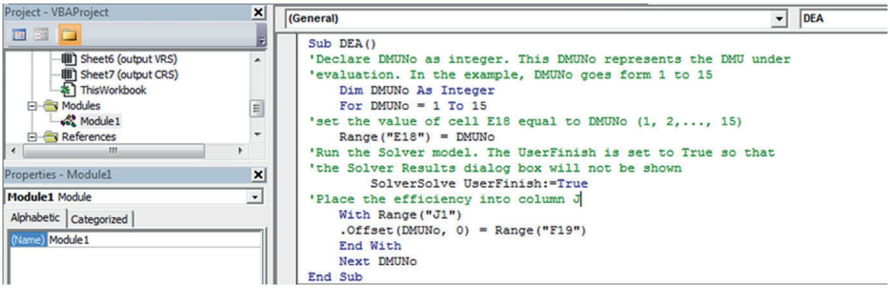


Fig. 2.20 VBA Code for input-oriented VRS envelopment model

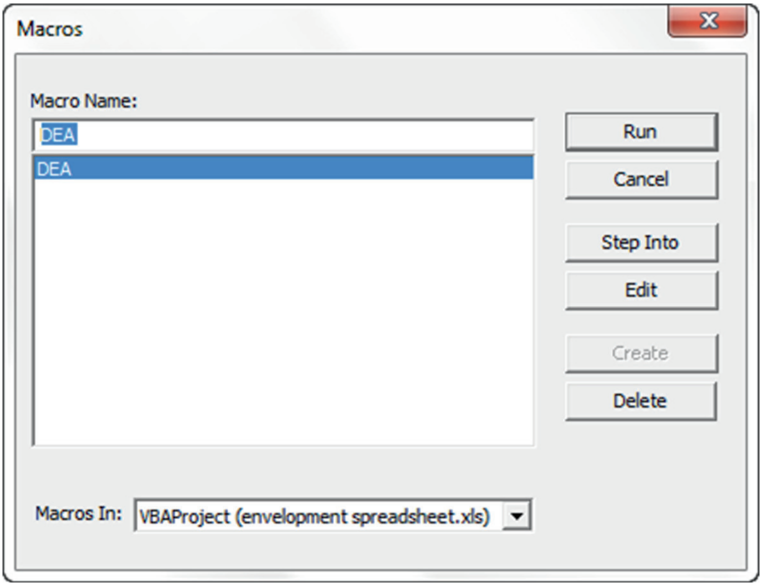


Fig. 2.21 Run "DEA" macro

ments "With Range("J1") and ".Offset(DMUNo, 0)=Range("F19") take the optimal objective function value (efficiency score) in cell F19 and place it in cell J "DMUNo" (that is, cell J2, J3, …, J16).

Enter the Run Macro dialog box by pressing *Alt-F8* key combination (or using the Developer/Macros tab). You should see "DEA", as shown in Fig. 2.21. Select "DEA" and then click the Run button. This action will generate the efficiency scores (cells J2:J16) for the 15 companies, as shown in Fig. 2.22.

Ten companies are efficient (on the VRS frontier). For the inefficient companies, the non-zero optimal λ_j indicate the benchmarks. For example, the efficiency score for AT&T is 0.53354 and the benchmarks for AT&T are Sumitoma ($\lambda_5=0.77$ in cell I6) and Wal-Mart ($\lambda_{11}=0.23$ in cell I12).

	A	B	C	D	E	F	G	H	I	J
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	Efficiency
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		0	1
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0	1
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	1
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0	1
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0.77	1
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	1
8	Ford Motor	243283	24547	346990		137137	4139		0	0.737556
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	0.603245
10	Exxon	91296	40436	82000		110009	6470		0	1
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	1
12	Wal-Mart	37871	14762	675000		93627	2740		0.23	1
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	0.557596
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0	1
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0	0.470611
16	AT&T	88884	17274	299300		79609	139		0	0.533544
17										
18		Reference		DMU under	15	Efficiency				
19	**Constraints**	set		Evaluation		0.533544				
20	Assets	47423.482	≤	47423.482						
21	Equity	8535.6544	≤	9216.4308						
22	Employees	159689.58	≤	159689.58						
23	Revenue	150569.21	≥	79609						
24	Profit	791.04056	≥	139						
25	Σλ	1	=	1						

Fig. 2.22 Input-oriented VRS envelopment efficiency

The previous macro "DEA" does not record the optimal λ_j in the worksheet. This can be done by the adding a VBA procedure named "DEA_1" into the existing module.

```
Sub DEA_1()
'Declare DMUNo as integer. This DMUNo represents the DMU under
'evaluation. In the example, DMUNo goes form 1 to 15
    Dim DMUNo As Integer
    For DMUNo = 1 To 15
'set the value of cell E18 equal to DMUNo (1, 2,..., 15)
    Range("E18") = DMUNo
'Run the Solver model. The UserFinish is set to True so that
'the Solver Results dialog box will not be shown
    SolverSolve UserFinish:=True
'Place the efficiency into column J
    Range("J" & DMUNo + 1) = Range("F19")
'Select the cells containing the optimal lambdas
    Range("I2:I16").Select
'copy the selected lambdas and paste them to row "DMUNo+1"
'(that is row 2, 3, ..., 16) starting with column K
Selection.Copy
Range("K" & DMUNo + 1).Select
Selection.PasteSpecial Paste:=xlPasteValues, Transpose:=True
Next DMUNo
End Sub
```

	A	B	C	D	E	F	G	H	I	J
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	Efficiency
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		0	1
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0	1
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	1
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0	1
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0.7705	1
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	1
8	Ford Motor	243283	24547	346990		137137	4139		0	0.737556
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	0.603245
10	Exxon	91296	40436	82000		110009	6470		0	1
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	1
12	Wal-Mart	37871	14762	675000		93627	2740		0.2295	1
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	0.557596
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0	1
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0	0.470611
16	AT&T	88884	17274	299300		79609	139		0	0.533544
17						Reserved to indicate the DMU under evaluation.				
18		Reference		DMU under	15					
19	**Constraints**	set		Evaluation						
20	Assets	47423.482	=	47423.482		0	slack; changing cell			
21	Equity	9216.4308	=	9216.4308		680.7764	slack; changing cell			
22	Employees	159689.58	=	159689.58		1.75E-09	slack; changing cell			
23	Revenue	79609	=	79609		70960.21	slack; changing cell			
24	Profit	139	=	139		652.0406	slack; changing cell			
25	Σλ	1	=	1		72293.02	Objective function Sum of slacks			

Fig. 2.23 Second-stage slack spreadsheet model

In the Run Macro dialog box, select "DEA_1" and then click the Run button. The procedure "DEA_1" will record both the efficiency scores and the related optimal values on λ_j ($j=1,2, \ldots, 15$) (see file "envelopment spreadsheet.xlsm").

2.5.7 Second-Stage Slack Calculation

Based upon the efficiency scores and the optimal values on λ_j ($j=1,2, \ldots, 15$), we now calculate the slack values using model (2.3).

Figure 2.23 shows the spreadsheet model for calculating the slacks after the efficiency scores are obtained. This spreadsheet model is built upon the spreadsheet model shown in Fig. 2.9 with efficiency scores reported in column J.

Cells F20:F24 are reserved for input and output slacks (changing cells). The formulas for cells B25 and D25 remain unchanged. The formulas for Cells B20:B24 are changed to

Cell B20 =SUMPRODUCT(B2:B16,$I$2:$I$16)+F20
Cell B21 =SUMPRODUCT(C2:C16,$I$2:$I$16)+F21
Cell B22 =SUMPRODUCT(D2:D16,$I$2:$I$16)+F22
Cell B23 =SUMPRODUCT(F2:F16,$I$2:$I$16)–F23
Cell B24 =SUMPRODUCT(G2:G16,$I$2:$I$16)–F24

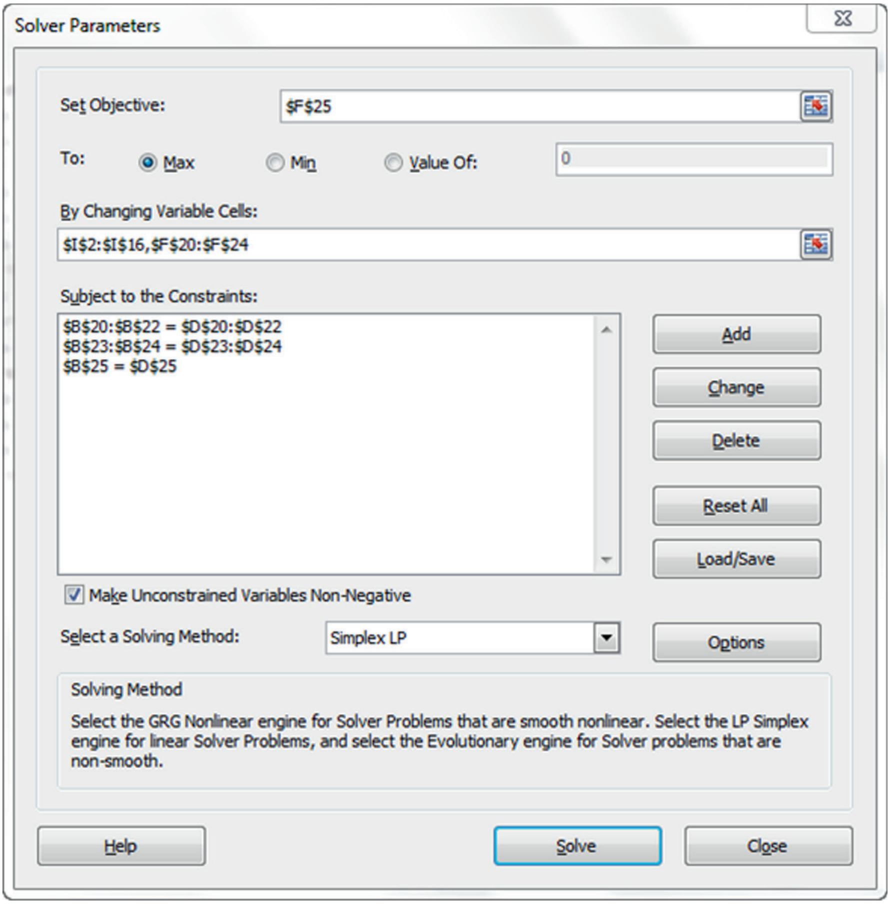


Fig. 2.24 Solver parameters for calculating slacks

The formulas for cells D21:D24 are

Cell D20 =INDEX(J2:J16,E18,1)*INDEX(B2:B16,E18,1)
Cell D21 =INDEX(J2:J16,E18,1)*INDEX(C2:C16,E18,1)
Cell D22 =INDEX(J2:J16,E18,1)*INDEX(D2:D16,E18,1)
Cell D23 =INDEX(F2:F16,E18,1)
Cell D24 =INDEX(G2:G16,E18,1)

After the Solver parameters are set up, as shown in Fig. 2.24, the VBA procedure "DEASlack" is inserted into the existing module to automate the slack calculations for the 15 companies (see file "envelopment spreadsheet.xlsm").

```
Sub DEASlack()
'Declare DMUNo as integer. This DMUNo represents the DMU under
'evaluation. In the example, DMUNo goes form 1 to 15
    Dim DMUNo As Integer
    For DMUNo = 1 To 15
'set the value of cell E18 equal to DMUNo (1, 2,..., 15)
    Range("E18") = DMUNo
'Run the Slack Solver model
    SolverSolve UserFinish:=True
'Select the cells containing the slacks
    Range("F20:F24").Select
'copy the selection (slacks) and paste it to row "DMUNo+1"
'(that is, row 2,3, ...,16) starting column L
Selection.Copy
Range("L" & DMUNo + 1).Select
Selection.PasteSpecial Paste:=xlPasteValues, Transpose:=True
Next DMUNo
End Sub
```

2.5.8 Other Input-Oriented Envelopment Spreadsheet Models

Figs. 2.9 and 2.15 represent the input-oriented VRS envelopment model. By changing the constraint of $\sum_{j=1}^{n}\lambda_j$, we immediately obtain other input-oriented envelopment models (Fig. 2.25).

For example, if we select \$B\$25 = \$D\$25 and click the Delete button in Fig. 2.15 (i.e., we remove $\sum_{j=1}^{n}\lambda_j = 1$), we obtain the Solver parameters for the input-oriented CRS envelopment model, as shown in Fig. 2.26.

If we click the Change button, and replace \$B\$25 = \$D\$25 with \$B\$25 ≤ \$D\$25 (or \$B\$25 ≥ \$D\$25), we obtain the input-oriented NIRS (or NDRS) envelopment model.

During this process, the spreadsheet shown in Fig. 2.9 and the VBA procedures remain unchanged. For example, if we run the Macro “DEA” for the input-oriented CRS envelopment model, we have the CRS efficiency scores shown in Fig. 2.26. Seven DMUs are on the CRS efficient frontier.

2.6 Output-Oriented Envelopment Spreadsheet Models

We next consider the output-oriented envelopment models. The spreadsheet model should be similar to the one in Fig. 2.9, but with a different set of formulas for the DMU under evaluation. Figure 2.27 shows a spreadsheet for the output-oriented VRS envelopment model.

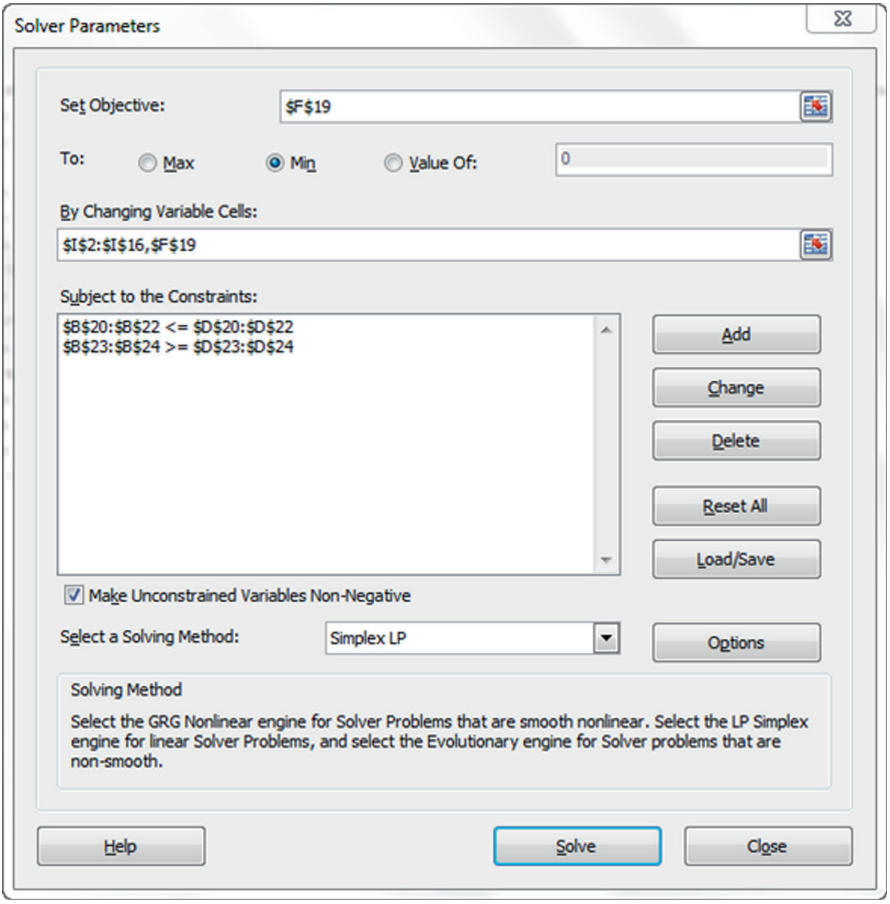


Fig. 2.25 Solver parameters for input-oriented CRS envelopment model

To make the spreadsheet more understandable, we use "range names" in the formulas. Select a range that needs to be named, and then type the desirable range name in the upper left "name box" in the Excel. This "name box" is just above the column A heading (see Fig. 2.27). For example, we select cells B2:D16 containing the inputs, and then type "InputUsed" in the "name box" (see Fig. 2.27). An alternative way is to use the Insert/Name/Define menu item. We can then refer to the inputs by using "InputUsed" in stead of cells B2:D16.

We name the cells F2:G16 containing the outputs as "OutputProduced". We also name the changing cells I2:I16 and F19 "Lambdas" and "Efficiency", respectively. As a result, the formulas on $\sum_{j=1}^{n} \lambda_j$ can be expressed as Cell B25 =SUM(Lambdas), and the formulas for the DEA reference set can be expressed as

Cell B20 = SUMPRODUCT(INDEX(InputUsed,0,1),Lambdas)

	A	B	C	D	E	F	G	H	I	J
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	Efficiency
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		0	0.662832
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0	1
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	1
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0	1
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0.47	1
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	0.971967
8	Ford Motor	243283	24547	346990		137137	4139		0	0.737166
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	0.524558
10	Exxon	91296	40436	82000		110009	6470		0	1
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	0.841424
12	Wal-Mart	37871	14762	675000		93627	2740		0.01	1
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	0.386057
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0	1
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0	0.348578
16	AT&T	88884	17274	299300		79609	139		0	0.270382
17										
18		Reference		DMU under	15	Efficiency				
19	**Constraints**	set		Evaluation		0.270382				
20	Assets	24032.613	≤	24032.613						
21	Equity	3338.6447	≤	4670.5747						
22	Employees	12922.984	≤	80925.264						
23	Revenue	79609	≥	79609						
24	Profit	139	≥	139						
25	Σλ	0.4817465								

Fig. 2.26 Input-oriented CRS envelopment efficiency

InputUsed f_x 91920.6

	A	B	C	D	E	F	G	H	I	J	K
1	Company	Assets	Equity	Employees		Revenue	Profit		λ		
2	Mitsubishi	91920.6	10950	36000		184365.2	1		0.66		
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0.31		
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0		
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0.03		
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0		
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0		
8	Ford Motor	243283	24547	346990		137137	4139		0		
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0		
10	Exxon	91296	40436	82000		110009	6470		0		
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0		
12	Wal-Mart	37871	14762	675000		93627	2740		0		
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0		
14	Nippon Life Insurance	364762.5	2241.9	89690		[illegible]	[illegible]		0		
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		[illegible]	[illegible]		0		
16	AT&T	88884	17274	299300		[illegible]	[illegible]		0		
17											
18		Reference		DMU under	15	Efficiency					
19	**Constraints**	set		Evaluation		2.298629					
20	Assets	88884	≤	88884							
21	Equity	9699.2872	≤	17274							
22	Employees	71229.375	≤	299300							
23	Revenue	182991.59	≥	182991.59							
24	Profit	319.50949	≥	319.50949							
25	Σλ	1									
26											

Name cells I2:I16 "Lambdas"
Reserved to indicate the DMU under evaluation.
Efficiency; ϕ; A changing variable cell; Target cell in Solver
Constraint on Σλ The cell is named "SumLambda"

Fig. 2.27 Output-oriented VRS envelopment spreadsheet model

Cell B21 = SUMPRODUCT(INDEX(InputUsed,0,2),Lambdas)
Cell B22 = SUMPRODUCT(INDEX(InputUsed,0,3),Lambdas)
Cell B23 = SUMPRODUCT(INDEX(OutputProduced,0,1),Lambdas)
Cell B24 = SUMPRODUCT(INDEX(OutputProduced,0,2),Lambdas)

Note that we use "0" for the "row number" in the INDEX function. This returns the whole column in the specified array in the INDEX function. For example, INDEX(InputUsed,0,1) returns the first input across all DMUs in cells B2:B16.

We assign a range name of "DMU" to cell E18, the cell representing the DMU under evaluation. The formulas for the DMU under evaluation then can be expressed as

Cell D20 =INDEX(InputUsed,DMU,1)
Cell D21 =INDEX(InputUsed,DMU,2)
Cell D22 =INDEX(InputUsed,DMU,3)
Cell D23 = Efficiency*INDEX(OutputProduced,DMU,1)
Cell D24 = Efficiency*INDEX(OutputProduced,DMU,2)

The cells "B20:B22", "B23:B24", "B25", "D20:D22", "D23:D24" are named as "ReferenceSetInput", "ReferenceSetOutput", "SumLambda", "DMUInput", and "DMUOutput", respectively. Based upon these range names, we obtain the Solver parameters shown in Fig. 2.28. Since it is an output-oriented envelopment model, "Max" is selected to maximize the efficiency (ϕ).

We can still apply the previous Excel macros ("DEA" or "DEA_1") to this spreadsheet model shown in Fig. 2.27 with the Solver parameters shown in Fig. 2.28. We next present an alternative approach to automate the DEA calculation.

First, turn on the ActiveX Controls by clicking the Developer/Insert tab. Then click the Command Button (ActiveX Controls) icon on the Insert tab, and drag it on to your worksheet (see Fig. 2.29).

While the command button is selected, you can change its properties by clicking the Properties tab. For example, setting the TakeFocusOnClick to False leaves the worksheet selection unchanged when the button is clicked. You can change the name of this Command Button by changing the Caption property with "Output-oriented VRS" (see Fig. 2.30).

Double click the command button. This should launch the VB Editor and bring up the code window for the command button's click event. Insert the statements shown in Fig. 2.31.

The word Private before the macro name "CommandButton1_Click" means that this macro will not appear in the Run Macro dialog box. The macro is only available to the worksheet "output VRS" containing the model (see Fig. 2.31).

In the macro, we introduce three variables, NDMUs, NInputs, and NOutputs, representing the number of DMUs, inputs and outputs, respectively. In the current example, NDMUs = 15, NInputs = 3, and NOutputs = 2. For a different set of DMUs, set these variables to different values, and the macro should still work.

Close the VB Editor and click the Design Mode tab to turn off design mode. The selection handles disappear from the command button. The macro runs when you click the command button. Figure 2.32 shows the results (see file "envelopment spreadsheet.xlsm").

In a similar manner, we can set up other output-oriented envelopment spreadsheet models. For example, if we remove "SumLambda=1" from Fig. 2.28, we obtain the Solver parameters for output-oriented CRS envelopment model.

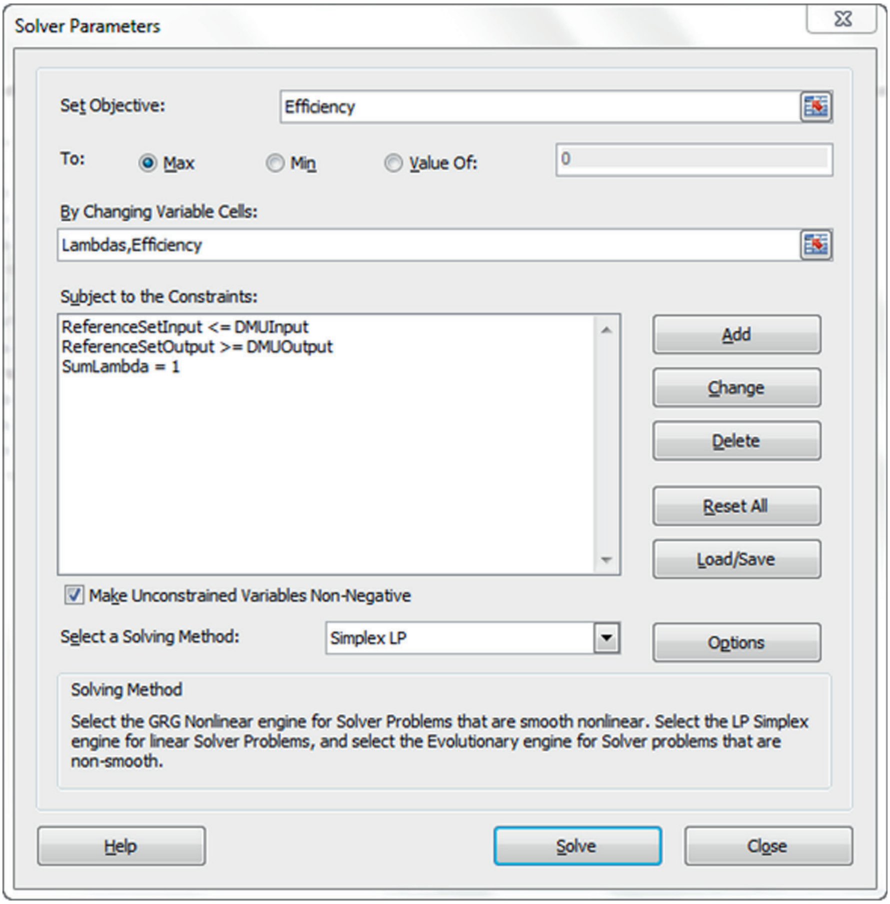


Fig. 2.28 Solver parameters for output-oriented VRS envelopment model

If one wants to use the macros established for the input-oriented envelopment spreadsheet models, one can proceed as follows using the Button (Form Controls).

First, we click the Developer tab and then select the button (the first item on the Insert tab) (see Fig. 2.29). Drag the button onto your worksheet containing the output-oriented CRS envelopment spreadsheet and the Solver parameters. You will immediately be asked to assign a macro to this button. Select “DEA_1”. At this point, the button is selected. You may also want to change the caption on the button to “Output-oriented CRS’, for example (see Fig. 2.33). To run the selected macro, you have to deselect the button by clicking anywhere else on the worksheet. You can always assign a different macro to the button by right-clicking on the button and selecting “Assign Macro”. Figure 2.34 shows the output-oriented CRS efficiency scores.

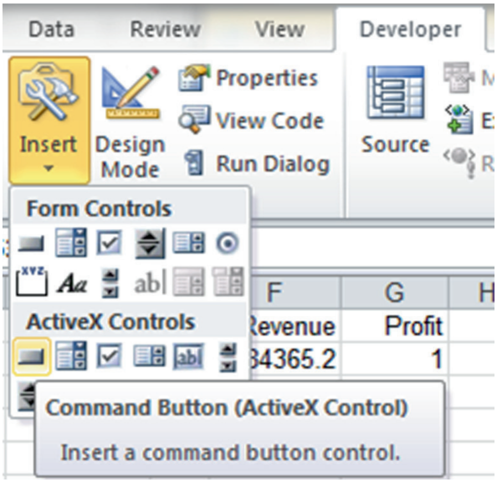


Fig. 2.29 Adding command button

2.7 Using OpenSolver

The standard built-in Excel Solver has its limitation with respect to how many changing cell it can have, for example. In our envelopment models, the total number of changing cells need to be reserved in an Excel sheet is the total number of DMUs (lambdas) plus one (efficiency score). If that number exceeds the standard Excel capacity, one needs to upgrade the Solver product (www.solver.com).

There is an open source linear and integer optimizer for Microsoft Excel called "OpenSolver" that can be used as an alternative to the standard Excel Solver. The OpenSolver is freely available at "opensolver.org". OpenSolver is an Excel VBA add-in that extends the standard Excel Solver with Linear Programming solver. OpenSolver is being developed by Andrew Mason in the Department of Engineering Science at the University of Auckland, and Iain Dunning.

There is no need to change the spreadsheets/models built with the standard Excel Solver. OpenSolver can use the existing Excel Solver parameters. For our DEA models in the book, it is recommended that the user builds the DEA models using the standard Excel Solver, and then use the OpenSolver to solve the DEA models.

We here use the input-oriented VRS model (Fig. 2.9) as an example to show how to use OpenSolver. OpenSolver is loaded under the Data tab as shown in Fig. 2.35.

Note the DEA model is already built using Excel Solver as shown in Fig. 2.9 and 2.15 (Excel Solver Parameters). If we click the Model option in OpenSolver, we will see that all the model parameters have been captured from the Excel Solver, as shown in Fig. 2.36.

If we click the "Solve" option, the same efficiency score will be obtained. However, we should use the VBA to automate the DEA calculation process. To do that, we first need to add a reference to OpenSolver in the VBA Editor, as shown in Fig. 2.37.

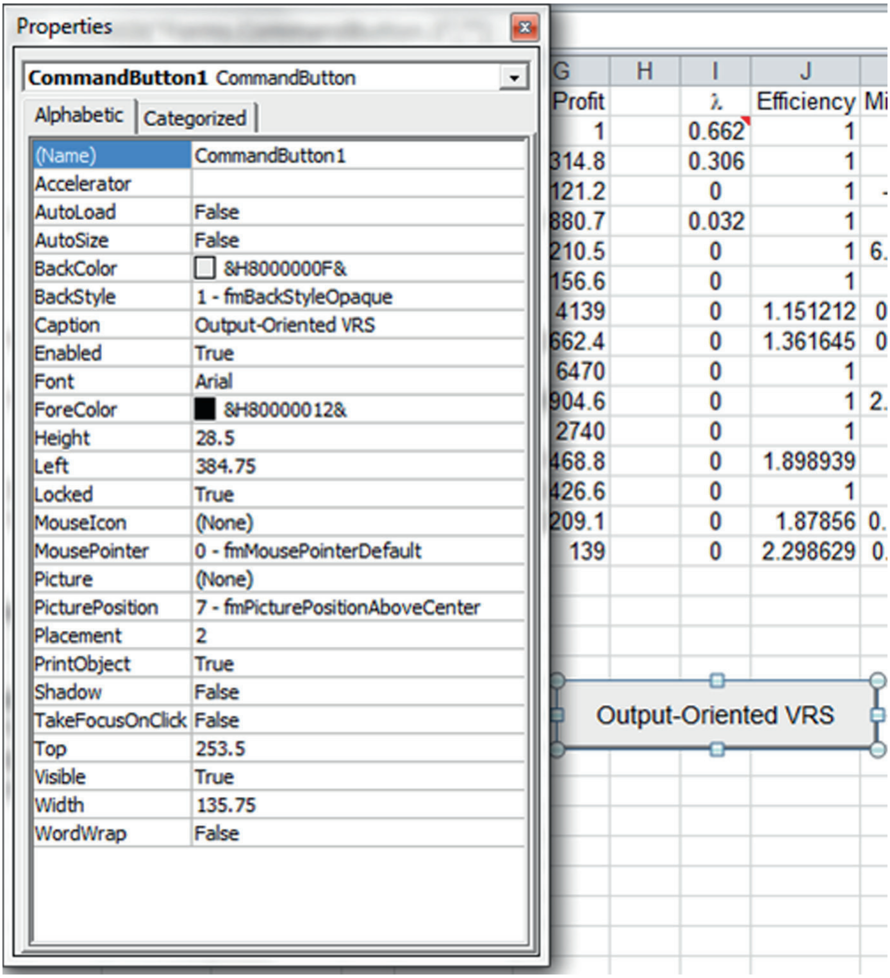


Fig. 2.30 Changing command button properties

Now, in the VBA code "DEA", we need to replace

"SolverSolve UserFinish:=True" with
"RunOpenSolver False".

See file "envelopment_OpenSolver.xlsm" for other VBA codes under OpenSolver.

Interested user should visit opensolver.org for additional information on how to use OpenSolver and its use with VBA. In the reminder of the book, we will not show how to solve DEA models with OpenSolver. This is because once the model is set up using Excel Solver, the user can following the above steps to execute the same model using OpenSolver.

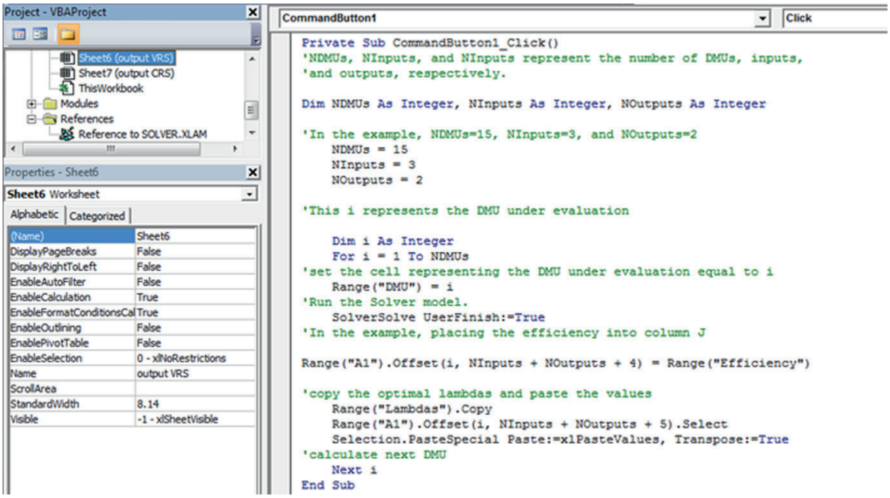


Fig. 2.31 VBA code for output-oriented VRS envelopment model

	A	B	C	D	E	F	G	H	I	J
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	Efficiency
2	Mitsubishi	91920.6	10950	36000		184365.2	1		0.66	1
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0.31	1
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	1
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0.03	1
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0	1
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	1
8	Ford Motor	243283	24547	346990		137137	4139		0	1.151212
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	1.361645
10	Exxon	91296	40436	82000		110009	6470		0	1
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	1
12	Wal-Mart	37871	14762	675000		93627	2740		0	1
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	1.898939
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0	1
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0	1.87856
16	AT&T	88884	17274	299300		79609	139		0	2.298629
17										
18		Reference		DMU under	15	Efficiency				
19	**Constraints**	set		Evaluation		2.298629				
20	Assets	88884	≤	88884						
21	Equity	9699.2872	≤	17274			Output-Oriented VRS			
22	Employees	71229.375	≤	299300						
23	Revenue	182991.59	≥	182991.59						
24	Profit	319.50949	≥	319.50949						
25	Σλ	1								

Fig. 2.32 Output-oriented VRS envelopment efficiency

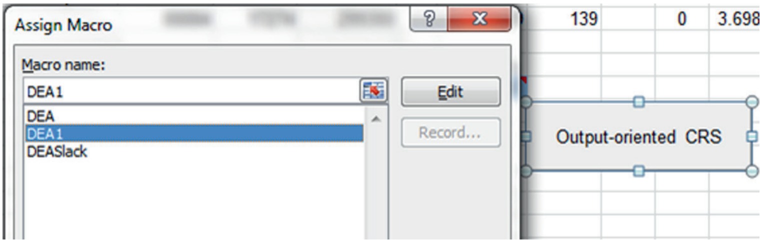


Fig. 2.33 Adding a button with macro

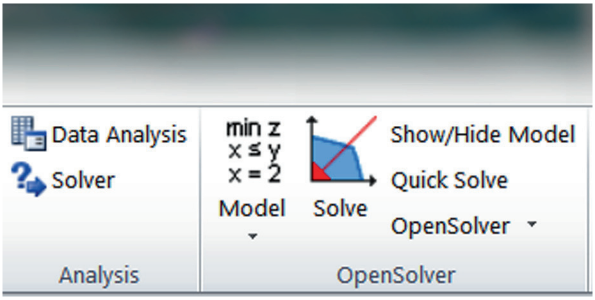
	A	B	C	D	E	F	G	H	I	J
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	Efficiency
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		0	1.508679
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0	1
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	1
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0	1
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		1.73	1
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	1.028842
8	Ford Motor	243283	24547	346990		137137	4139		0	1.356546
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	1.906368
10	Exxon	91296	40436	82000		110009	6470		0	1
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	1.188462
12	Wal-Mart	37871	14762	675000		93627	2740		0.05	1
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	2.590289
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0	1
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0	2.8688
16	AT&T	88884	17274	299300		79609	139		0	3.698474
17										
18		Reference		DMU under	15	Efficiency				
19	**Constraints**	set		Evaluation		3.698474				
20	Assets	88884	≤	88884						
21	Equity	12347.891	≤	17274			Output-oriented CRS			
22	Employees	47795.324	≤	299300						
23	Revenue	294431.83	≥	294431.83						
24	Profit	514.08791	≥	514.08791						
25	Σλ	1.7817269								

Fig. 2.34 Output-oriented CRS envelopment efficiency

2.8 Solving DEA Using DEAFrontier Software

One can solve the envelopment DEA models using the spreadsheets and Excel Solver (or OpenSolver) We now demonstrate how to solve the above DEA models using a special version of *DEAFrontier* software supplied with the book. See www.deafrontier.net for full the version of the *DEAFrontier* software. This version of *DEAFrontier* is an Add-In for Microsoft® Excel and uses the Excel Solver. This version of software requires Excel 2007–2013 and can solve up to 30 DMUs with unlimited number of inputs and outputs (subject to the capacity of the standard Excel Solver). To install the software, copy the file "DEAFrontier.xlam" to your hard drive.

Fig. 2.35 OpenSolver



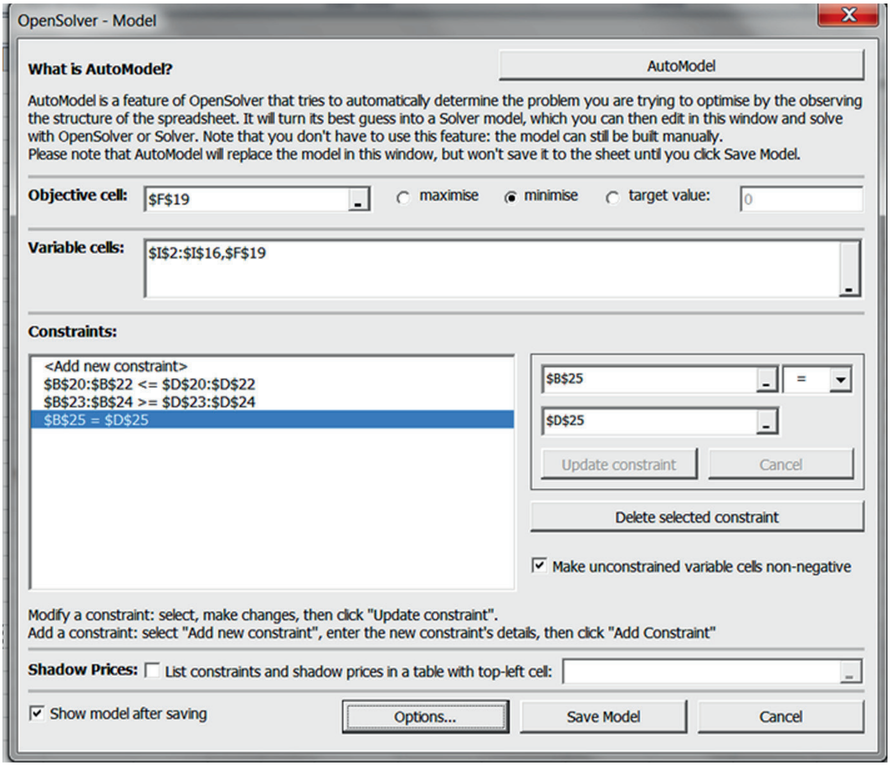


Fig. 2.36 OpenSolver model

If you run the DEAFrontier and get an error message (as shown in Fig. 2.38), this error message means that the Excel Solver is not found by the *DEAFrontier* software. To correct this, please use the following steps:

Step 1: Open Excel

Step 2: Load Excel Solver so that the Excel Solver parameters dialog box (Fig. 2.13) is displaced.

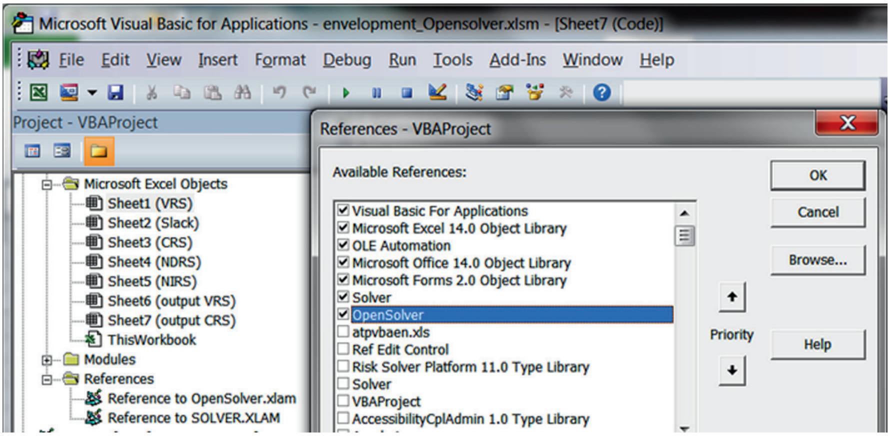


Fig. 2.37 Reference to OpenSolver in VBA project

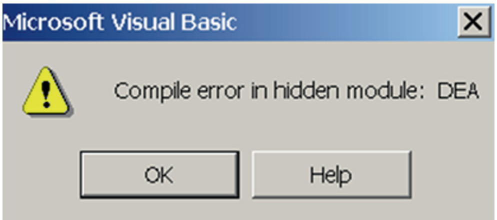


Fig. 2.38 DEAFrontier error message

Step 3: Close the Excel Solver parameters dialog box.

Step 4: Load the DEA software by opening "DEAFrontier.xlam".

To locate the DEA Menu, select the Add-Ins tab and navigate to the DEA menu option, as shown in Fig. 2.39

2.8.1 Data Sheet Format

The Excel sheets for storing the DEA data (inputs and outputs for DMUs) must have the same format as shown in Figs. 2.40 and 2.41. Leave one blank column between the input and output data. No blank columns and rows are allowed within the input and output data.

Negative or non-numerical data are deemed as invalid data. The software checks if the data are in valid form before the calculation. If the data sheet contains negative or non-numerical data, the software will quit and locate the invalid data (see Fig. 2.42).

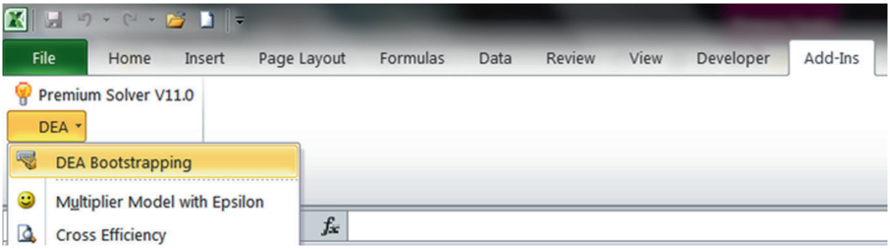


Fig. 2.39 DEAFrontier menu

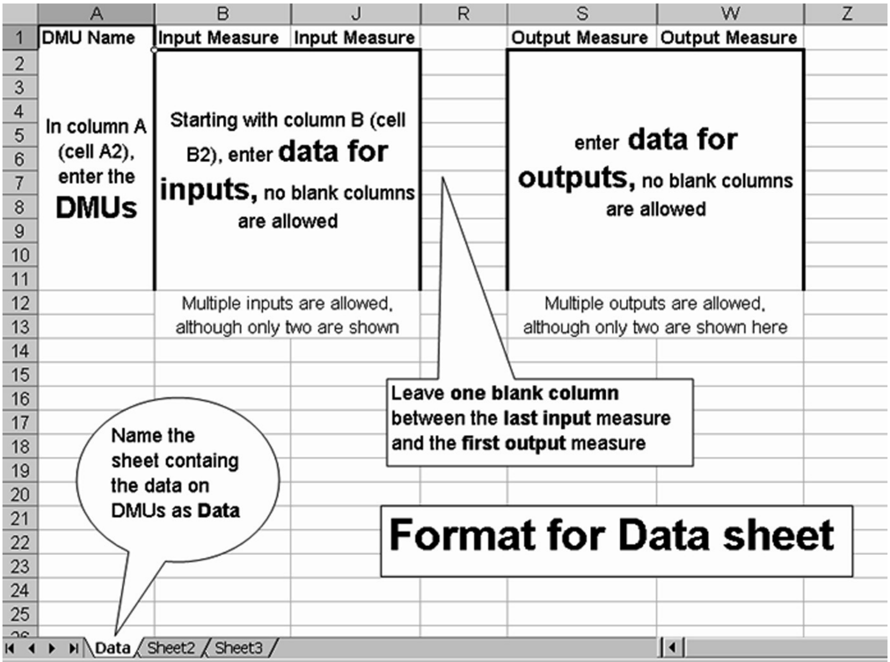
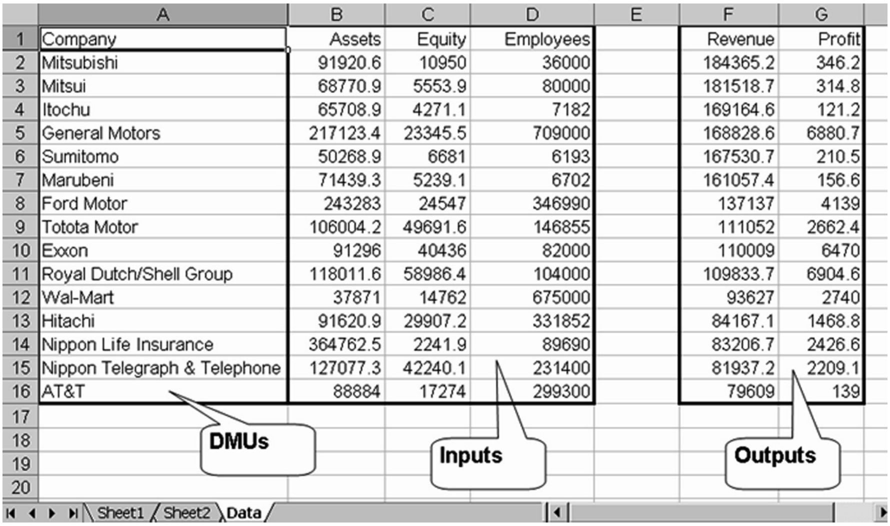


Fig. 2.40 Data sheet format

2.8.2 *Envelopment Models*

To run the envelopment models in Table 2.1, select the "Envelopment Model" menu item. You will be prompted with a form for selecting the sheet storing the DEA data, model orientation, and frontier type, as shown in Fig. 2.43.



	A	B	C	D	E	F	G
1	Company	Assets	Equity	Employees		Revenue	Profit
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8
4	Itochu	65708.9	4271.1	7182		169164.6	121.2
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7
6	Sumitomo	50268.9	6681	6193		167530.7	210.5
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6
8	Ford Motor	243283	24547	346990		137137	4139
9	Totota Motor	106004.2	49691.6	146855		111052	2662.4
10	Exxon	91296	40436	82000		110009	6470
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6
12	Wal-Mart	37871	14762	675000		93627	2740
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1
16	AT&T	88884	17274	299300		79609	139

Fig. 2.41 Example data sheet

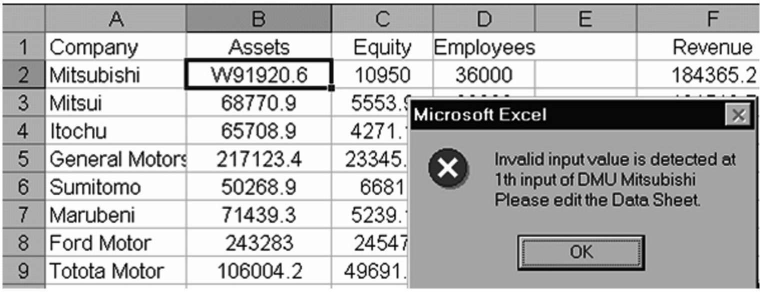


Fig. 2.42 Invalid data

The software performs a two-stage DEA calculation. First, the efficiency scores are calculated, and the efficiency scores and benchmarks (λ_j^*) are reported in the "Efficiency" sheet. At the same time, a "Slack" sheet and a "Target" sheet are generated based upon the efficiency scores and the λ_j^*.

Then you will be asked whether you want to perform the second-stage calculation, i.e., fixing the efficiency scores and calculating the DEA slacks (see Fig. 2.44). If Yes, then the slack and target sheets will be replaced by new ones which report the DEA slacks and the efficient targets defined in Table 2.1.

For example, Fig. 2.45 reports the efficiency results for the input-oriented CRS envelopment model. Column A reports the DMU No. Column B reports the DMU names. Column C reports the efficiency scores (it also indicates the type of DEA

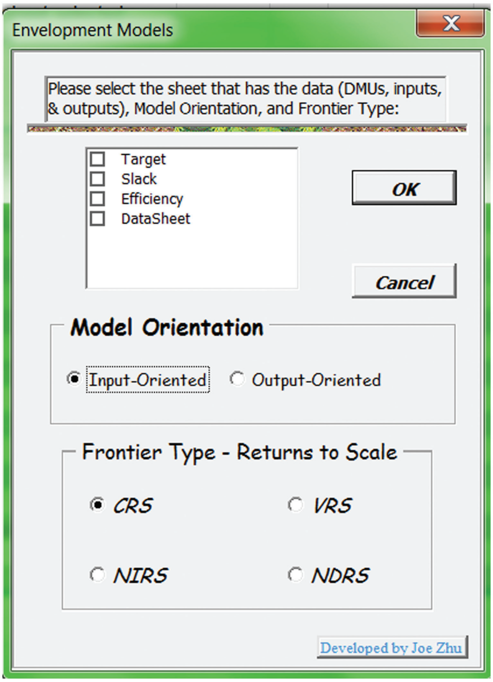


Fig. 2.43 Envelopment models

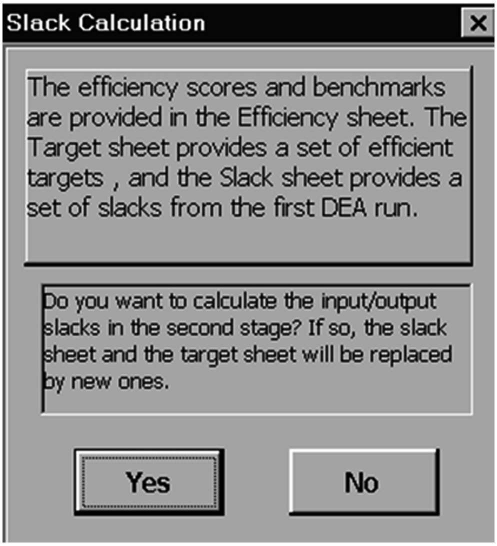


Fig. 2.44 Second stage DEA slack calculation

	A	B	C	D	E	F	G	H	I	J
1	**Inputs**		**Outputs**				The data sheet selected			
2	Assets		Revenue				DataSheet			
3	Equity		Profit							
4	Employees									
5										
6			Input-Oriented							
7			CRS							
8	DMU No.	DMU Name	Efficiency	Σλ	RTS	**Benchmarks**				
9	1	Mitsubishi	0.66283	1.102	Decreasing		0.184	Itochu		0.017
10	2	Mitsui	1.00000	1.000	Constant		1.000	Mitsui		
11	3	Itochu	1.00000	1.000	Constant		1.000	Itochu		
12	4	General Motors	1.00000	1.000	Constant		1.000	General Motors		
13	5	Sumitomo	1.00000	1.000	Constant		1.000	Sumitomo		
14	6	Marubeni	0.97197	0.956	Increasing		0.547	Itochu		0.408
15	7	Ford Motor	0.73717	0.998	Increasing		0.253	Itochu		0.305
16	8	Totota Motor	0.52456	0.819	Increasing		0.382	Sumitomo		0.371
17	9	Exxon	1.00000	1.000	Constant		1.000	Exxon		
18	10	Royal Dutch/Shell Group	0.84142	1.067	Decreasing		1.067	Exxon		
19	11	Wal-Mart	1.00000	1.000	Constant		1.000	Wal-Mart		
20	12	Hitachi	0.38606	0.627	Increasing		0.312	Sumitomo		0.145
21	13	Nippon Life Insurance	1.00000	1.000	Constant		1.000	Nippon Life Insurance		
22	14	Nippon Telegraph & Telephone	0.34858	0.619	Increasing		0.012	General Motors		0.247
23	15	AT&T	0.27038	0.482	Increasing		0.467	Sumitomo		0.015
24										

Target / Slack / Efficiency / DataSheet

Fig. 2.45 CRS DEA results

models used). Column D reports the optimal $\sum \lambda_j^*$ which is used to identify the RTS classifications reported in column E (see Chap. 16) for discussions on RTS). Sheet "Efficiency" also reports the benchmark DMUs along with the optimal λ_j^*. The data sheet selected is reported in cell G2.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_2) contains supplementary material, which is available to authorized users.

References

Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Science, 30,* 1078–1092.

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring efficiency of decision making units. *European Journal of Operational Research, 2,* 429–444.

Chapter 3
Multiplier DEA Model

3.1 Multiplier Models in Spreadsheets

The dual linear programming problems to the envelopment models are called multiplier models as shown in Table 3.1.

The dual variables v_i and μ_r are called multipliers. A DMU is on the frontier if and only if $\sum_{r=1}^{s} \mu_r y_{ro} + \mu = 1$ (or $\sum_{i=1}^{m} v_i x_{io} + v = 1$) in optimality. The ε in the envelopment model essentially requires that v_i and μ_r are positive in the multiplier models. The constraint $\sum_{i=1}^{m} v_i x_{io} = 1$ (or $\sum_{r=1}^{s} \mu_r y_{ro} = 1$) is known as a normalization constraint. In DEA, the weighted input and output of $\sum_{i=1}^{m} v_i x_{ij}$ and $\sum_{r=1}^{s} \mu_r y_{rj}$ are called virtual input and virtual output, respectively. See Seiford and Thrall (1990) for a detailed discussion on these models.

Note that $\mu_r, v_i \geq \varepsilon$. This set of constraints ensures that a DMU with an efficiency score of one must be efficient. If a DMU's efficiency score equals one with non-zero slacks in an envelopment model, then this DMU must have a score less than one in the above related multiplier model (with ε). That is, if we impose $\mu_r, v_i \geq \varepsilon$ in the multiplier models, the two-stage process in the envelopment models is automatically carried out in the calculation. However, note that ε is a very small positive value and usually is set equal to 10^{-6}, and such choice does not always work. It is also possible that the multiplier model can be infeasible because the ε is not correctly selected.

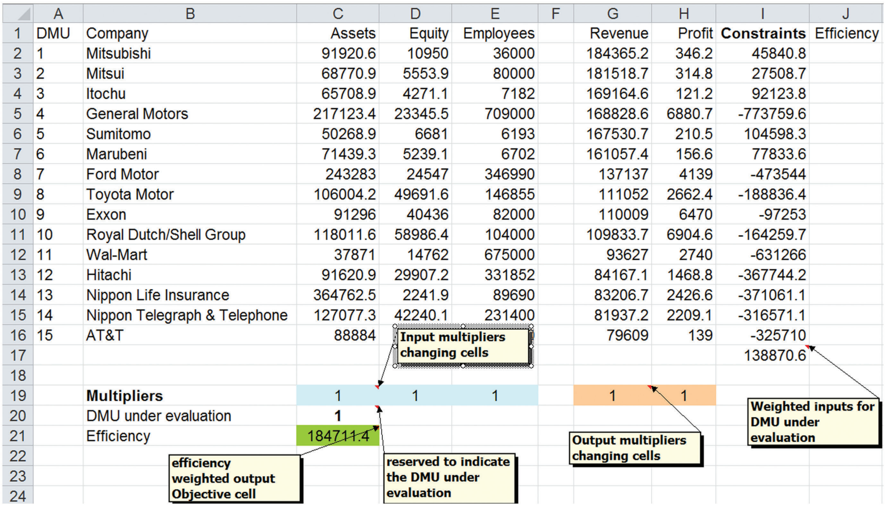
Figure 3.1 presents the input-oriented CRS multiplier spreadsheet model. We name the cells C2:E16 containing the inputs as "InputUsed" and the cells G2:H16 containing the outputs as "OutputProduced". Cells C19:E19 and G19:H19 are reserved for the decision variables—input and output multipliers, and are named "InputMultiplier" and "OutputMultiplier", respectively. Cells A2:A16 are reserved for DMU numbers which are used in the formulas in cells I2:I16

Cell I2 contains the formula "= SUMPRODUCT(OutputMultiplier, INDEX (OutputProduced,A2,0))-SUMPRODUCT(InputMultiplier,INDEX (InputUsed,A2,0))" which represents the difference between weighted output and weighted input for DMU1. This value will be set as non-negative in the Solver parameters.

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213, DOI 10.1007/978-3-319-06647-9_3,

Table 3.1 Multiplier models

Frontier type	Input-oriented	Output-oriented
	$\max \sum_{r=1}^{s} \mu_r y_{ro} + \mu$ subject to $\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} + \mu \le 0$ $\sum_{i=1}^{m} v_i x_{io} = 1$ $\mu_r, v_i \ge 0(\varepsilon)$	$\min \sum_{i=1}^{m} v_i x_{io} + v$ subject to $\sum_{i=1}^{m} v_i x_{ij} - \sum_{r=1}^{s} \mu_r y_{rj} + v \ge 0$ $\sum_{r=1}^{s} \mu_r y_{ro} = 1$ $\mu_r, v_i \ge 0(\varepsilon)$
CRS	where $\mu = 0$	where $v = 0$
VRS	where μ free	where v free
NIRS	where $\mu \le 0$	where $v \ge 0$
NDRS	where $\mu \ge 0$	where $v \le 0$



	A	B	C	D	E	F	G	H	I	J
1	DMU	Company	Assets	Equity	Employees		Revenue	Profit	**Constraints**	Efficiency
2	1	Mitsubishi	91920.6	10950	36000		184365.2	346.2	45840.8	
3	2	Mitsui	68770.9	5553.9	80000		181518.7	314.8	27508.7	
4	3	Itochu	65708.9	4271.1	7182		169164.6	121.2	92123.8	
5	4	General Motors	217123.4	23345.5	709000		168828.6	6880.7	-773759.6	
6	5	Sumitomo	50268.9	6681	6193		167530.7	210.5	104598.3	
7	6	Marubeni	71439.3	5239.1	6702		161057.4	156.6	77833.6	
8	7	Ford Motor	243283	24547	346990		137137	4139	-473544	
9	8	Toyota Motor	106004.2	49691.6	146855		111052	2662.4	-188836.4	
10	9	Exxon	91296	40436	82000		110009	6470	-97253	
11	10	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6	-164259.7	
12	11	Wal-Mart	37871	14762	675000		93627	2740	-631266	
13	12	Hitachi	91620.9	29907.2	331852		84167.1	1468.8	-367744.2	
14	13	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6	-371061.1	
15	14	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1	-316571.1	
16	15	AT&T	88884				79609	139	-325710	
17									138870.6	
18										
19		**Multipliers**	1	1	1		1	1		
20		DMU under evaluation	1							
21		Efficiency	184711.4							
22										
23										
24										

Fig. 3.1 Input-oriented CRS Multiplier spreadsheet model

The function INDEX(array,row number,0) returns the entire row in the array. For example, the value for cell A2 is one, therefore INDEX(OutputProduced,A2,0) returns all the outputs for DMU1, i.e., cells G2:H2.

The formula in cell I2 is then copied into cells I3:I16. Cells I2:I16 are named "ConstraintDMUj".

The formula for cell I17 is "= SUMPRODUCT (InputMultiplier, INDEX (InputUsed,DMU,0))", where DMU is a range name for cell C20, indicating the

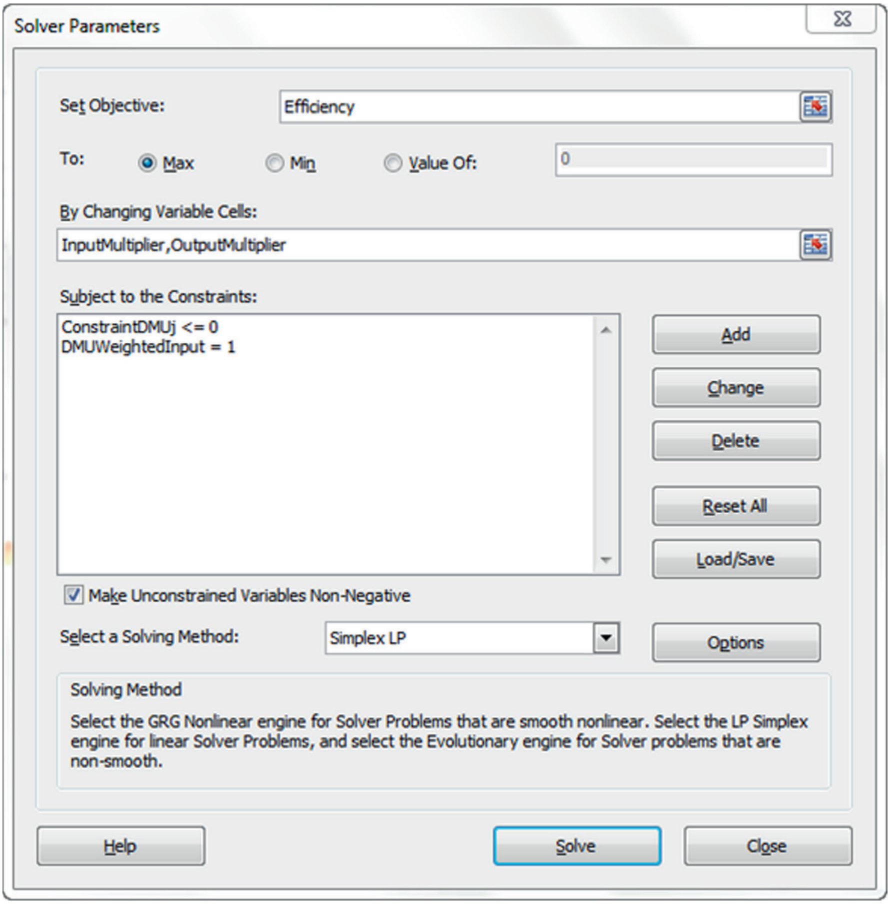


Fig. 3.2 Solver parameters for input-oriented CRS multiplier model

DMU under evaluation. The value of cell I17 will be set equal to one in the Solver parameters. Cell I17 is named “DMUWeightedInput”.

The objective cell is C21 which represents the efficiency—weighted output for the DMU under evaluation. The cell C21 is named “Efficiency”. Its formula is “= SUMPRODUCT(OutputMultiplier,INDEX(OutputProduced, DMU,0))”.

Note that initial values of one are entered into the cells for the multipliers. As a result, some of the constraints are violated, and the value in cell C21 (efficiency) is greater than one. However, once the Solver solves, these values will be replaced by optimal solutions.

Figure 3.2 shows the Solver parameters for the spreadsheet model in Fig. 3.1.

Figure 3.3 shows the optimal solutions for DMU1 with an efficiency of 0.66283. To calculate the CRS efficiencies for the remaining DMUs, we insert a VBA procedure “MultiplierCRS” to automate the computation, as shown in Fig. 3.4. Note that the name of the module is changed to “MultiplierDEA”. This VBA procedure

Efficiency f_x =SUMPRODUCT(OutputMultiplier, INDEX(OutputProduced,DMU,0))

	A	B	C	D	E	F	G	H	I	J
1	DMU	Company	Assets	Equity	Employees		Revenue	Profit	**Constraints**	Efficiency
2	1	Mitsubishi	91920.6	10950	36000		184365.2	346.2	-0.4069718	0.66283
3	2	Mitsui	68770.9	5553.9	80000		181518.7	314.8	-0.1572998	1
4	3	Itochu	65708.9	4271.1	7182		169164.6	121.2	-0.1720259	1
5	4	General Motors	217123.4	23345.5	709000		168828.6	6880.7	-1.5931016	1
6	5	Sumitomo	50268.9	6681	6193		167530.7	210.5	0	1
7	6	Marubeni	71439.3	5239.1	6702		161057.4	156.6	-0.2619527	0.97197
8	7	Ford Motor	243283	24547	346990		137137	4139	-2.1077375	0.73717
9	8	Toyota Motor	106004.2	49691.6	146855		111052	2662.4	-0.7118903	0.52456
10	9	Exxon	91296	40436	82000		110009	6470	-0.3901587	1
11	10	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6	-0.6730777	0.84142
12	11	Wal-Mart	37871	14762	675000		93627	2740	0	1
13	12	Hitachi	91620.9	29907.2	331852		84167.1	1468.8	-0.6894804	0.38606
14	13	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6	-3.7255062	1
15	14	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1	-1.064743	0.34858
16	15	AT&T	88884	17274	299300		79609	139	-0.7296182	0.27038
17									1.03416363	
18										
19		**Multipliers**	1.125E-05	0	0		3.32E-06	4E-05		
20		DMU under evaluation	**1**							
21		Efficiency	0.6271918							

Fig. 3.3 Input-oriented CRS multiplier efficiency

```
Sub MultiplierCRS()
Dim NDMUs As Integer, NInputs As Integer, NOutputs As Integer
    NDMUs = 15
    NInputs = 3
    NOutputs = 2
    Dim i As Integer
    For i = 1 To NDMUs
    Range("DMU") = i
SolverSolve UserFinish:=True
'record the efficiency scores
    Range("A1").Offset(i, NInputs + NOutputs + 4) = Range("Efficiency")
'record the optimal multipliers
    Range("InputMultiplier").Copy
    Range("A1").Offset(i, NInputs + NOutputs + 5).Select
    Selection.PasteSpecial Paste:=xlPasteValues
    Range("OutputMultiplier").Copy
    Range("A1").Offset(i, 2 * NInputs + NOutputs + 6).Select
    Selection.PasteSpecial Paste:=xlPasteValues

    Next i
End Sub
```

Fig. 3.4 VBA code for input-oriented CRS multiplier model

	A	B	C	D	E	F	G	H	I	J
1	DMU	Company	Assets	Equity	Employees		Revenue	Profit	**Constraints**	Efficiency
2	1	Mitsubishi	91920.6	10950	36000		184365.2	346.2	-0.4469263	
3	2	Mitsui	68770.9	5553.9	80000		181518.7	314.8	-0.2104174	
4	3	Itochu	65708.9	4271.1	7182		169164.6	121.2	-0.1636977	
5	4	General Motors	217123.4	23345.5	709000		168828.6	6880.7	-1.9048865	
6	5	Sumitomo	50268.9	6681	6193		167530.7	210.5	-1.11E-16	
7	6	Marubeni	71439.3	5239.1	6702		161057.4	156.6	-0.2242861	
8	7	Ford Motor	243283	24547	346990		137137	4139	-2.1108427	
9	8	Toyota Motor	106004.2	49691.6	146855		111052	2662.4	-0.6178272	
10	9	Exxon	91296	40436	82000		110009	6470	-0.4493419	
11	10	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6	-0.7365683	
12	11	Wal-Mart	37871	14762	675000		93627	2740	0	
13	12	Hitachi	91620.9	29907.2	331852		84167.1	1468.8	-0.5018292	
14	13	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6	-3.3467539	
15	14	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1	-0.8575795	
16	15	AT&T	88884	17274	299300		79609	139	-0.4664565	
17									0.9804698	
18									free variable	0.53354
19		**Multipliers**	1.059E-05	0	1.963E-07		0	0	0.53354352	0
20		DMU under evaluation	1							
21		Efficiency	0.5335435							

Fig. 3.5 Input-oriented VRS multiplier spreadsheet model

works for other sets of DMUs when setting the "NDMUs", "NInputs", and "NOutputs" equal to proper values. In the current example, this VBA procedure takes the efficiency in cell C21 and places it into cells J2:J16, and also takes the optimal multipliers and places them into cells K2:M16 and O2:P16 for 15 DMUs. Select and run the macro "MultiplierCRS" in the Run Macro dialog box will generate the efficiency results. You may also create a button in Forms toolbar and assign macro "MultiplierCRS" to the button (see file "multiplier.xlsm").

Spreadsheets for other multiplier models can be set up in a similar manner. For example, Fig. 3.5 shows a spreadsheet model for the input-oriented VRS multiplier model.

Because we have a decision variable that is free in sign, we need to introduce two variables in cells I19 and J19. The free variable in the VRS multiplier model is represented by cell J18 with a formula of "=I19-J19". In the Solver parameters, cells I19 and J19 (not cell J18) along with cells C19:E19 and G19:H19 are changing cells.

The formula for cell I2 is

Cell I2 =SUMPRODUCT(G2:H2,$G$19:$H$19)- SUMPRODUCT(C2:E2, $C$19:$E$19)+$I$19-$J$19

Cells for the multipliers and free variables are used as absolute references indicated by the dollar sign. This allows us to copy the formula in cell I2 to cells I3:I16. Figure 3.6 shows the Solver parameters for the input-oriented VRS multiplier spreadsheet model.

Insert the VBA procedure "MultiplierVRS" shown in Fig. 3.7 into the existing module "MultiplierDEA". The macro records the efficiency score in cells J2:J16, optimal free variable in cells K2:K16, and optimal multipliers in cells L2:N16 and P2:Q16 for 15 DMUs (see file "multiplier.xlsm").

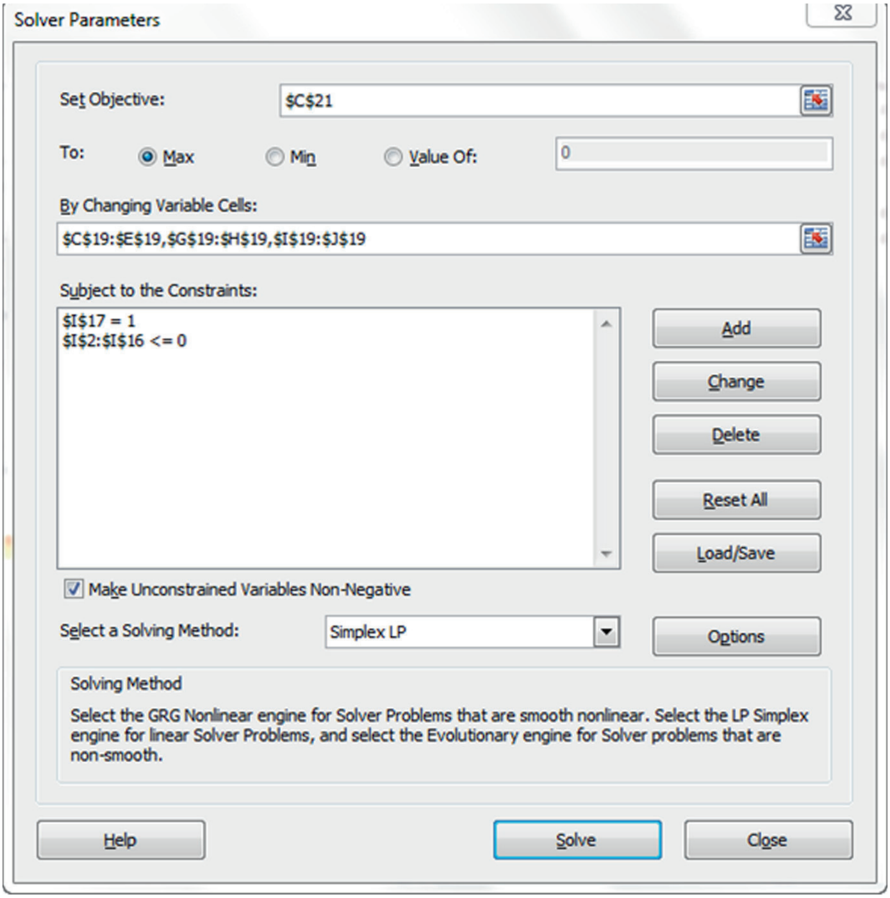


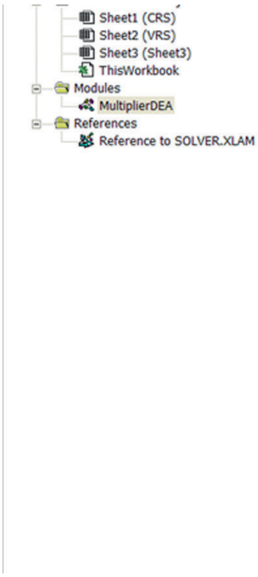
Fig. 3.6 Solver parameters for input-oriented CRS multiplier model

3.2 Weight Restrictions in Multiplier Models

In the DEA literature, a number of approaches have been proposed to introduce additional restrictions on the values that the multipliers can assume.

Some of the techniques for enforcing these additional restrictions include imposing bounds on ratios of multipliers (Thompson et al. 1990), appending multiplier inequalities (Wong and Beasley 1990), and requiring multipliers to belong to given closed cones (Charnes et al. 1989), among others.

We here present the assurance region (AR) approach of Thompson et al. (1990). To illustrate the AR approach, suppose we wish to incorporate additional inequality constraints of the following form into the multiplier DEA models as given in Table 2.1:




```
Sub MultiplierVRS()
'Declare i as integer. This i represents the DMU under
'evaluation. In the example, i goes form 1 to 15
    Dim i As Integer
    For i = 1 To 15
'set the value of cell C20 equal to i (1, 2,..., 15)

    Range("C20") = i
'Run the Solver model. The UserFinish is set to True so that
'the Solver Results dialog box will not be shown

    SolverSolve UserFinish:=True
'Place the efficiency in cell C20 into column J

    Range("J" & i + 1) = Range("C21")
'Select the cells containing the optimal multipliers

    Range("C19:H19").Select
'copy the selected multipliers and paste them to row "i+1"
'(that is row 2, 3, ..., 16) starting with column L

    Selection.Copy
    Range("L" & i + 1).Select
    Selection.PasteSpecial Paste:=xlPasteValues
'copy the value of free variable in cell J18 into column K

    Range("K" & i + 1) = Range("J18")
    Next
End Sub
```

Fig. 3.7 VBA Code for the input-oriented VRS multiplier model

$$\begin{aligned} &\alpha_i \le \frac{v_i}{v_{i_o}} \le \beta_i, \qquad i = 1,\ldots,m \\ &\delta_r \le \frac{\mu_r}{\mu_{r_o}} \le \gamma_r, \qquad r = 1,\ldots,s \end{aligned} \qquad \text{(AR)}$$

Here, v_{i_o} and μ_{r_o} represent multipliers which serve as "numeraires" in establishing the upper and lower bounds represented here by α_i, β_i, and by δ_r, γ_r for the multipliers associated with inputs $i=1, \ldots, m$ and outputs $r=1, \ldots, s$ where $\alpha_{i_o} = \beta_{i_o} = \delta_{r_o} = \gamma_{r_o} = 1$. The above constraints are called Assurance Region (AR) constraints as in Thompson et al. (1990).

Uses of such bounds are not restricted to prices. For example, Zhu (1996) uses an assurance region approach to establish bounds on the weights obtained from uses of Analytic Hierarchy Processes in Chinese textile manufacturing in order to reflect how the local government in measuring the textile manufacturing performance.

For example, we can include the following AR constraints

$$1 \le \frac{v_{Employee}}{v_{Assets}} \le 2.5$$

$$1.5 \le \frac{v_{Employee}}{v_{Equity}} \le 3$$

$$3 \le \frac{\mu_{MarketValue}}{\mu_{Revenue}} \le 4$$

The first AR constraint indicates that Employee input should be at most 2.5 times as important as the Assets input, but at least as important as the Assets input.

It is noted that the AR constraints in the above form are non-linear, however, they can be converted into linear restrictions, namely

$$\begin{aligned} &\alpha_i v_{i_o} \le v_i \le \beta_i v_{i_o}, \qquad i = 1,\ldots,m \\ &\delta_r \mu_{r_o} \le \mu_r \le \gamma_r \mu_{r_o}, \quad r = 1,\ldots,s \end{aligned}$$

or

$$\begin{aligned} &\alpha_i v_{i_o} \le v_i, \qquad i = 1,\ldots,m \\ &v_i \le \beta_i v_{i_o}, \qquad i = 1,\ldots,m \\ &\delta_r \mu_{r_o} \le \mu_r, \qquad r = 1,\ldots,s \\ &\mu_r \le \gamma_r \mu_{r_o}, \qquad r = 1,\ldots,s \end{aligned}$$

We next incorporate $1 \le \frac{v_{Employee}}{v_{Assets}} \le 2.5$ into the CRS multiplier model shown in Fig. 3.3. The following two additional constraints are needed

$$1 v_{Assets} \le v_{Employee} \text{ and } v_{Employee} \le 2.5 v_{Assets}$$

Cells G22:G23 contains the left-hand-side of the above two constraints and cells I22:I23 contains the right-hand-side of the above two constraints, as shown in Fig. 3.8. In the Solver parameters, we need to add these two additional constraints, as shown in Fig. 3.9.

	A	B	C	D	E	F	G	H	I	J
1	DMU	Company	Assets	Equity	Employees		Revenue	Profit	**Constraints**	Efficiency
2	1	Mitsubishi	91920.6	10950	36000		184365.2	346.2	-0.210733	0.60413
3	2	Mitsui	68770.9	5553.9	80000		181518.7	314.8	-0.0746789	0.82717
4	3	Itochu	65708.9	4271.1	7182		169164.6	121.2	-2.248E-15	1
5	4	General Motors	217123.4	23345.5	709000		168828.6	6880.7	-0.1888734	0.90829
6	5	Sumitomo	50268.9	6681	6193		167530.7	210.5	-0.0400331	1
7	6	Marubeni	71439.3	5239.1	6702		161057.4	156.6	-0.0386433	0.88263
8	7	Ford Motor	243283	24547	346990		137137	4139	-0.4430373	0.69786
9	8	Toyota Motor	106004.2	49691.6	146855		111052	2662.4	-1.1801741	0.37262
10	9	Exxon	91296	40436	82000		110009	6470	0	1
11	10	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6	-0.551944	0.83648
12	11	Wal-Mart	37871	14762	675000		93627	2740	-0.6204026	0.52316
13	12	Hitachi	91620.9	29907.2	331852		84167.1	1468.8	-1.0360741	0.28055
14	13	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6	5.6621E-15	1
15	14	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1	-1.197011	0.31726
16	15	AT&T	88884	17274	299300		79609	139	-0.8792355	0.12076
17									1	
18										
19		**Multipliers**	1.172E-06	3.2E-05	1.172E-06		1.15E-06	0.0002		
20		DMU under evaluation	**15**							
21		Efficiency	0.1207645			AR				
22						1st	1.17E-06	<=	1.1715E-06	
23						2nd	1.17E-06	<=	2.9288E-06	
24		CRS AR								

Fig. 3.8 CRS AR multiplier model

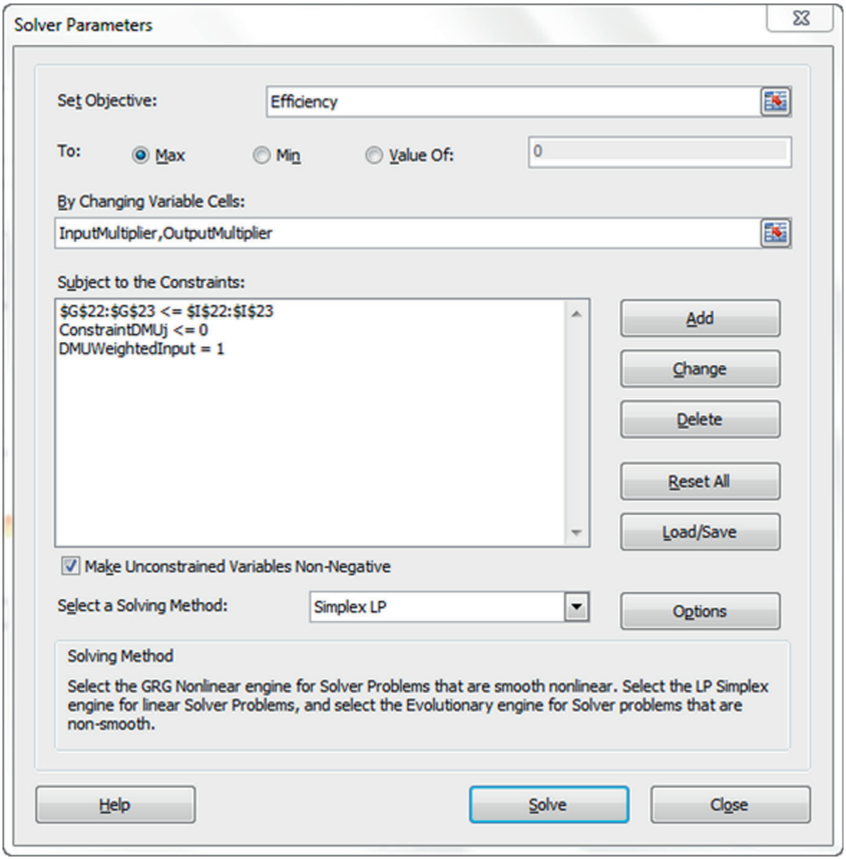


Fig. 3.9 Solver parameters for CRS AR model

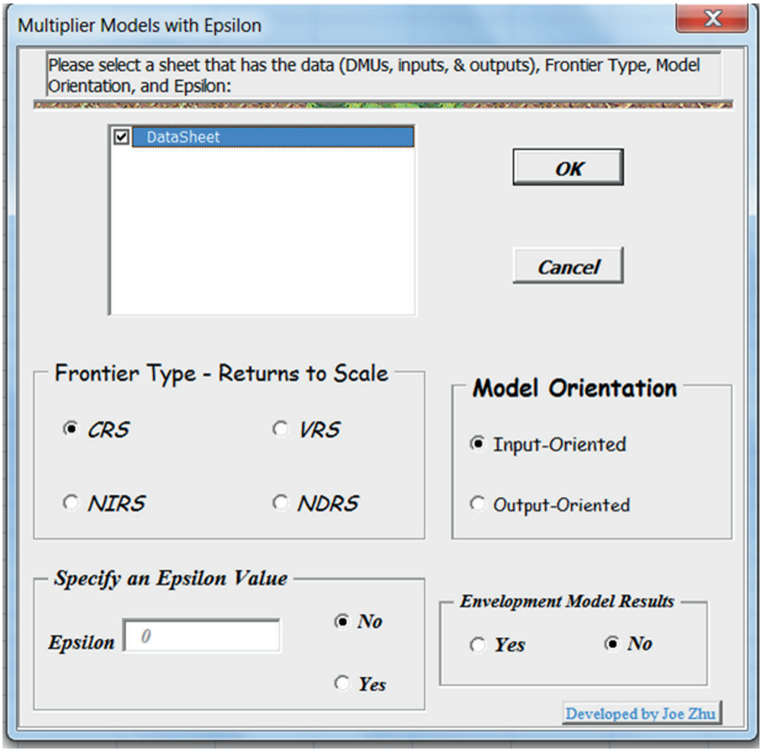


Fig. 3.10 Multiplier model

3.3 Solving Multiplier Models Using DEAFrontier

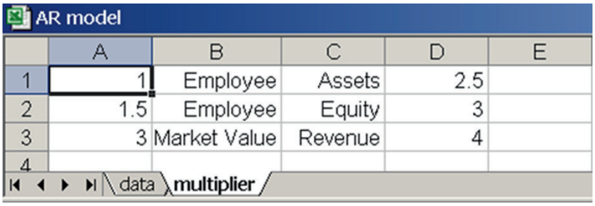
To run the multiplier models, select the "Multiplier Model with Epsilon" menu item. You will be prompted with a form for selecting the models presented in Table 3.1. As shown in Fig. 3.10, the default ε value = 0. The user can specify its own non-zero ε. The results are reported in a sheet named "Efficiency Report".

To run the AR model, we need to set up the sheet "Multiplier" which contains the ARs. For example, if we want to include the following ARs

$$1 \le \frac{v_{Employee}}{v_{Assets}} \le 2.5$$

$$1.5 \le \frac{v_{Employee}}{v_{Equity}} \le 3$$

$$3 \le \frac{\mu_{MarketValue}}{\mu_{Revenue}} \le 4$$



AR model

	A	B	C	D	E
1	1	Employee	Assets	2.5	
2	1.5	Employee	Equity	3	
3	3	Market Value	Revenue	4	
4					

Fig. 3.11 Restrictions (AR) on multipliers

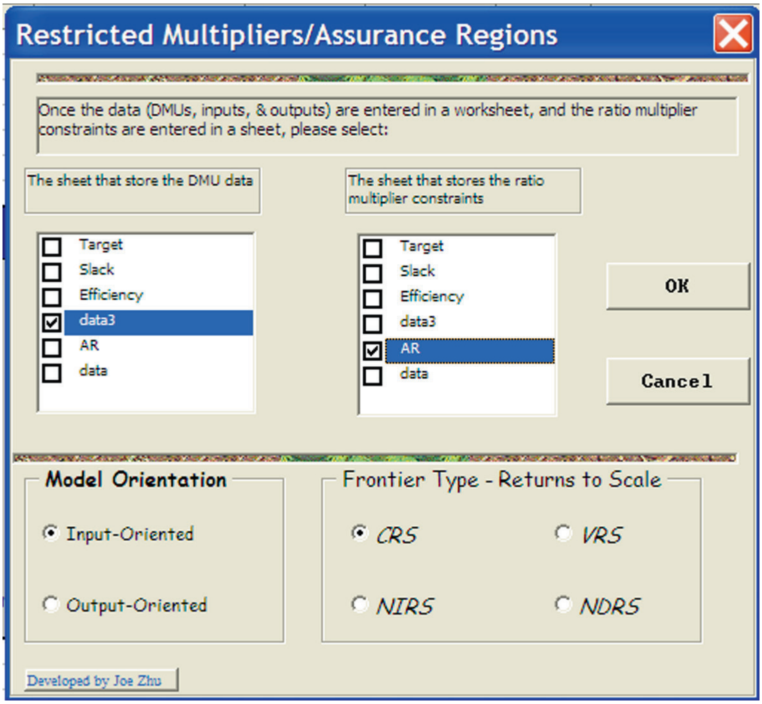


Fig. 3.12 Restricted multiplier model

then the data in the "Multiplier" sheet should be entered as shown in the following Fig. 3.11.

To avoid any errors, we suggest copying and pasting the input and output names from the "data" sheet when you enter the information into the "Multiplier" sheet. If the input (output) names in the two sheets do not match, the program will stop.

Once the sheets for the DMU data and multiplier restrictions are set up, select the "Restricted Multipliers" menu item and you will be prompted to choose a data sheet, an AR sheet, and DEA model, as shown in Fig. 3.12. Figure 3.13 shows the results of the input-oriented CRS multiplier model with the above ARs.

Note that you can also add ARs that link the input and output multipliers for the "Restricted Multipliers". Note also that if the ARs are not properly specified, then

	A	B	C	D	E	F	G	H
1	**Inputs**		**Outputs**				The data sheet selected	
2	Assets		Revenue				data3	
3	Equity		Profit					
4	Employees							
5								
6			Input-Oriented					
7			CRS	Optimal Multipliers				
8	DMU No.	DMU Name	Efficiency	Assets	Equity	Employees	Revenue	Profit
9	1	Mitsubishi	0.49699	0.00001	0.00000	0.00001	0.00000	0.00001
10	2	Mitsui	0.43370	0.00001	0.00000	0.00001	0.00000	0.00001
11	3	Itochu	0.85381	0.00001	0.00002	0.00003	0.00001	0.00002
12	4	General Motors	0.07544	0.00000	0.00000	0.00000	0.00000	0.00000
13	5	Sumitomo	1.00000	0.00001	0.00001	0.00004	0.00001	0.00002
14	6	Marubeni	0.76195	0.00001	0.00002	0.00003	0.00000	0.00001
15	7	Ford Motor	0.09166	0.00000	0.00000	0.00000	0.00000	0.00000
16	8	Totota Motor	0.15745	0.00000	0.00000	0.00000	0.00000	0.00001
17	9	Exxon	0.25360	0.00001	0.00000	0.00001	0.00000	0.00001
18	10	Royal Dutch/Shell Group	0.19825	0.00000	0.00000	0.00000	0.00000	0.00001
19	11	Wal-Mart	0.05236	0.00000	0.00000	0.00000	0.00000	0.00000
20	12	Hitachi	0.07347	0.00000	0.00000	0.00000	0.00000	0.00000
21	13	Nippon Life Insurance	0.07373	0.00000	0.00000	0.00000	0.00000	0.00000
22	14	Nippon Telegraph & Telephone	0.08494	0.00000	0.00000	0.00000	0.00000	0.00000
23	15	AT&T	0.07256	0.00000	0.00000	0.00000	0.00000	0.00000
24								

data3 | Efficiency Report | AR | Sheet3

Fig. 3.13 Restricted multiplier model results

the related DEA model may be infeasible. If that happens, the program will return a value "−9999" for the efficiency score.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_3) contains supplementary material, which is available to authorized users.

References

Charnes, A., Cooper, W. W., Wei, Q. L., & Huang, Z. M. (1989). Cone ratio data envelopment analysis and multi-objective programming. *International Journal of Systems Science, 20*(7), 1099–1118.

Seiford, L. M., & Thrall, R. M. (1990). Recent developments in DEA: The mathematical programming approach to frontier analysis. *Journal of Econometrics, 46,* 7–38.

Thompson, R. G., Langemeier, L. N., Lee, C. T., Lee, E., Thrall, R. M. (1990). The role of multiplier bounds in efficiency analysis with application to Kansas farming. *Journal of Econometrics, 46,* 93–108.

Wong, Y.-H. B., & Beasley, J. E. (1990). Restricting weight flexibility in data envelopment analysis. *Journal of Operational Research Society, 41/9,* 829–835.

Zhu, J. (1996). DEA/AR analysis of the 1988–1989 performance of the Nanjing Textile Cooperation. *Annals of Operations Research, 66,* 311–335.

Chapter 4
DEA Cross Efficiency

4.1 Introduction

While DEA has been proven an effective approach in identifying the best practice frontiers, its flexibility in weighting multiple inputs and outputs and its nature of self-evaluation have been criticized. The cross efficiency method is developed as a DEA extension to rank DMUs (Sexton et al. 1986) with the main idea being to use DEA to do peer evaluation, rather than in pure self-evaluation mode. Cross efficiency has been further investigated by Doyle and Green (1994). There are mainly two advantages for cross-evaluation method. It provides an ordering among DMUs and it eliminates unrealistic weight schemes without requiring the elicitation of weight restrictions from application area experts (e.g., Anderson et al. 2002).

Cross efficiency evaluation has been used in various applications, e.g., efficiency evaluations of nursing homes (Sexton et al. 1986), R&D project selection (Oral et al. 1991), preference voting (Green et al. 1996), and others. However, as noted in Doyle and Green (1994), the non-uniqueness of the DEA optimal weights/multipliers possibly reduces the usefulness of cross efficiency. Specifically, cross efficiency scores obtained from the original DEA are generally not unique, and depend on which of the alternate optimal solutions to the DEA linear programs is used. Sexton et al. (1986) and Doyle and Green (1994) propose to use a secondary goal to deal with the non-unique DEA solutions. They developed aggressive (benevolent) model formulations to identify optimal weights that not only maximize the efficiency of a particular DMU under evaluation, but also minimize (maximize) the average efficiency of other DMUs

In the current chapter, we will present the standard DEA cross efficiency method. We then discuss several approaches that are developed to address the non-uniqueness of the cross efficiency. They include the game cross efficiency by Liang et al. (2008) and maximum cross efficiency (Cook and Zhu 2014) based upon a set of log-linear DEA models. We will also show how to use the *DEAFrontier* software on these different cross efficiency approaches.

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_4,

4.2 Cross Efficiency

Suppose we have a set of n DMUs and each DMU_j have s different outputs and m different inputs. We denote the ith input and rth output of $DMU_j\,(j = 1, 2, \ldots, n)$ as $x_{ij}\,(i = 1, \ldots, m)$ and $y_{rj}\,(r = 1, \ldots, s)$, respectively. Cross efficiency is often calculated as a two-phase process. The first phase is calculated using the CRS DEA model of Charnes et al. (1978).

Suppose DMU_d is under evaluation by the CRS model (Charnes et al. 1978). Then its efficiency score is determined by the following DEA model

$$
\begin{aligned}
&Max\ \ E_{dd} = \frac{\sum_{r=1}^{s} u_{rd}\, y_{rd}}{\sum_{i=1}^{m} v_{id}\, x_{id}} \\
&s.t.\ \ E_{dj} = \frac{\sum_{r=1}^{s} u_{rd}\, y_{rj}}{\sum_{i=1}^{m} v_{id}\, x_{ij}} \le 1,\ j = 1, 2, \ldots, n. \\
&\qquad u_{rd} \ge 0, r = 1, \ldots, s. \\
&\qquad v_{id} \ge 0, i = 1, \ldots, m.
\end{aligned}
\tag{4.1}
$$

where v_{id} and u_{rd} represent ith input and rth output weights for DMU_d.

The cross efficiency of DMU_j, using the weights that DMU_d has chosen in model (4.1), is then:

$$
E_{dj} = \frac{\sum_{r=1}^{s} u_{rd}^{*}\, y_{rj}}{\sum_{i=1}^{m} v_{id}^{*}\, x_{ij}},\ d, j = 1, 2, \ldots, n
\tag{4.2}
$$

where (*) denotes optimal values in model (4.1). For DMU_j ($j=1, 2, \ldots, n$), an average of all E_{dj} ($d=1, 2, \ldots, n$),

$$
\overline{E_j} = \frac{1}{n} \sum_{d=1}^{n} E_{dj},
\tag{4.3}
$$

referred to as the *cross efficiency score* for DMU_j.

We should point out that each individual E_{dj} is called cross efficiency and the average defined in (4.3) is also called cross efficiency in the DEA literature. In general, "cross efficiency" refers to the average defined in (4.3), not the individual scores defined in (4.2).

	Rated DMU						
Rating DMU	1	2	3	4	5	6	Averaged appraisal of peers
1	$\mathbf{E_{11}}$	E_{12}	E_{13}	E_{14}	E_{15}	E_{16}	A_1
2	E_{21}	$\mathbf{E_{22}}$	E_{23}	E_{24}	E_{25}	E_{26}	A_2
3	E_{31}	E_{32}	$\mathbf{E_{33}}$	E_{34}	E_{35}	E_{36}	A_3
4	E_{41}	E_{42}	E_{43}	$\mathbf{E_{44}}$	E_{45}	E_{46}	A_4
5	E_{51}	E_{52}	E_{53}	E_{54}	$\mathbf{E_{55}}$	E_{56}	A_5
6	E_{61}	E_{62}	E_{63}	E_{64}	E_{65}	$\mathbf{E_{66}}$	A_6
	$\bar{E}_1$	$\bar{E}_2$	$\bar{E}_3$	$\bar{E}_4$	$\bar{E}_5$	$\bar{E}_6$	
	Averaged appraisal by peers (peer appraisal)						

Fig. 4.1 Cross Efficiency Matrix (Doyle and Green 1994)

DEA model (4.1) is a DEA model in efficiency ratio form and is equivalent to the input-oriented CRS multiplier model

$$\begin{aligned}
&\max E_{dd} = \sum_{r=1}^{s} u_{rd} y_{rd} \\
&\text{Subject to} \\
&\sum_{i=1}^{m} v_{id} x_{id} = 1 \\
&\sum_{r=1}^{s} u_{rd} y_{rj} - \sum_{i=1}^{m} v_{id} x_{ij} \le 0 \qquad j = 1, \ldots., n \\
&u_{rd}, v_{id} \ge 0
\end{aligned} \tag{4.4}$$

Due to the fact that the above cross efficiency is based upon input-oriented models, cross efficiency scores are not greater than one.

We here briefly illustrate the concept of cross efficiency by adopting the cross efficiency matrix from Doyle and Green (1994). In Fig. 4.1, we have six DMUs. E_{dj} is the (cross) efficiency of DMU_j based upon a set of DEA weights calculated for DMU_d. This set of DMU weights gives the best efficiency score for DMU_d under evaluation by a DEA model, and E_{dd} (in the leading diagonal) is the DEA efficiency for DMU_d. The cross efficiency for a given DMU_j is defined as the arithmetic average down column j, given by $\bar{E}_j$. (We point out that in Doyle and Green (1994), the efficiency score for DMU k is not included as part of the average.)

Obviously, E_{dj} ($d \neq j$) and $\bar{E}_j$ are not unique due to multiple optimal DEA weights/multipliers. As a result of this non uniqueness, the cross efficiency concept has been criticized as unreliable.

Note that the above discussion is based upon input-orientation. Similarly, we can use output-oriented models to calculate cross efficiency. In this case, E_{dj} in (4.2) based upon output-oriented DEA model becomes

$$E_{dj} = \frac{\sum_{i=1}^{m} v_{id}^{*} x_{ij}}{\sum_{r=1}^{s} u_{rd}^{*} y_{rj}} \tag{4.5}$$

where v_{id}^{*} and u_{rd}^{*} are optimal values in the following output-oriented model when DMU_d is under evaluation

$$\begin{aligned} \min E_{dd} &= \frac{\sum_{i=1}^{m} v_{id} x_{id}}{\sum_{r=1}^{s} u_{rd} y_{rd}} \\ s.t. \quad E_{dj} &= \frac{\sum_{i=1}^{m} v_{id} x_{ij}}{\sum_{r=1}^{s} u_{rd} y_{rj}} \geq 1, j = 1, 2, \ldots, n \\ v_{id} &\geq 0, i = 1, \ldots, m \\ u_{rd} &\geq 0, r = 1, \ldots, s. \end{aligned} \tag{4.6}$$

The above model (4.6) is equivalent to the output-oriented CRS multiplier model:

$$\begin{aligned} & Min \sum_{r=1}^{m} v_{id} x_{id} \\ & \text{subject to} \\ & \sum_{i=1}^{m} v_{id} x_{ij} - \sum_{r=1}^{s} u_{rd} y_{rj} \geq 0, j = 1, 2, \ldots, n \\ & \sum_{i=1}^{s} u_{rd} y_{rd} = 1 \\ & v_{id} \geq 0, i = 1, \ldots, m \\ & u_{rd} \geq 0, r = 1, \ldots, s \end{aligned} \tag{4.7}$$

Note that under output-oriented case, like the output-oriented CRS score, all cross efficiency scores are not less than one. The output-oriented DEA cross efficiency score can be defined in a similar manner as in (4.3).

Finally, the above discussion is based upon CRS. Similar developments can be obtained under VRS, NIRS, and NIRS. We, however, point out that negative cross efficiency scores can be obtained under non-CRS conditions.

	A	B	L	M	N	O	P	Q
1	DMU	Company	Assets	Equity	Employees		Revenue	Profit
2	1	Mitsubishi	5.55E-06	4.38E-05	2.83E-07		3.07E-06	0.000277
3	2	Mitsui	9.26E-06	6.54E-05	0		5.06E-06	0.000257
4	3	Itochu	1.09E-05	6.6E-05	0		5.91E-06	0
5	4	General Motors	7.14E-07	2.06E-05	5.13E-07		0	0.000145
6	5	Sumitomo	1.99E-05	0	0		5.97E-06	0
7	6	Marubeni	0	8.61E-05	8.19E-05		4.58E-06	0.001498
8	7	Ford Motor	8.96E-07	2.3E-05	6.25E-07		8.48E-07	0.00015
9	8	Toyota Motor	9.15E-06	0	2.06E-07		2.64E-06	8.67E-05
10	9	Exxon	7.59E-07	2.19E-05	5.46E-07		0	0.000155
11	10	Royal Dutch/Shell Group	0	0	9.62E-06		0	0.000122
12	11	Wal-Mart	6.3E-06	4.38E-05	1.7E-07		0	0.000365
13	12	Hitachi	1.01E-05	0	2.27E-07		2.92E-06	9.57E-05
14	13	Nippon Life Insurance	2.02E-06	5.84E-05	1.46E-06		0	0.000412
15	14	Nippon Telegraph & Telephone	2.34E-06	1.58E-05	1.49E-07		1.2E-06	0.000113
16	15	AT&T	1.13E-05	0	0		3.32E-06	4.19E-05

Fig. 4.2 Optimal multipliers

	A	B	C	D	E	F	G	H	I	J	K
20	DMU	Company	1	2	3	4	5	6	7	8	9
21	1	Mitsubishi	**0.662832**	0.996737	1	1	1	0.857933	0.622139	0.384642	0.926578
22	2	Mitsui	0.652396	**1**	1	0.740936	1	0.852272	0.455342	0.294387	0.635435
23	3	Itochu	0.630959	0.959659	**1**	0.255003	1	0.845149	0.189457	0.147906	0.177351
24	4	General Motors	0.162465	0.22362	0.127097	**1**	0.173128	0.140163	0.701457	0.329368	1
25	5	Sumitomo	0.601826	0.791994	0.772485	0.233316	**1**	0.67647	0.169141	0.314346	0.361561
26	6	Marubeni	0.350136	0.18522	1	0.1844	1	**0.971967**	0.223603	0.275786	1
27	7	Ford Motor	0.583677	0.840123	1	1	0.856905	0.848077	**0.737166**	0.370936	1
28	8	Toyota Motor	0.609941	0.78561	0.759576	0.489288	1	0.670854	0.314145	**0.524558**	1
29	9	Exxon	0.162465	0.22362	0.127097	1	0.173128	0.140163	0.701457	0.329368	**1**
30	10	Royal Dutch/Shell Group	0.12188	0.049872	0.213878	0.122997	0.430785	0.29614	0.151178	0.22977	1
31	11	Wal-Mart	0.118627	0.166455	0.073434	1	0.125829	0.083949	0.566335	0.33849	1
32	12	Hitachi	0.609941	0.78561	0.759576	0.489288	1	0.670854	0.314145	0.524558	1
33	13	Nippon Life Insurance	0.162465	0.22362	0.127097	1	0.173128	0.140163	0.701457	0.329368	1
34	14	Nippon Telegraph & Telephone	0.659905	0.969928	0.971875	1	1	0.838291	0.6281	0.411703	1
35	15	AT&T	0.606472	0.796695	0.767302	0.347831	1	0.674081	0.229933	0.403083	0.620149
36											
37		**cross efficiency**	0.446399	0.599918	0.646628	0.657537	0.72886	0.580435	0.447004	0.347218	0.848072

Fig. 4.3 Cross efficiency calculation

4.3 Cross Efficiency in Spreadsheets

The cross efficiency can actually be calculated directly from the spreadsheet for multiplier CRS model, as shown in Fig. 4.3, where the efficiency scores and related optimal multipliers have been calculated, as shown in Fig. 4.2. In Fig. 4.2, optimal multipliers obtained from model (4.4) (input-oriented CRS multiplier model) are reported in columns L to N and columns P and Q. Using these multipliers and (4.2), we can calculate cross efficiency scores for these 15 companies. In this case, we do not need to modify and/or build any new Solver parameters.

Figure 4.3 shows a partial cross efficiency matrix. The user is recommended to refer to the Excel file "cross efficiency.xlsm" for detailed information, including the formulas used. For example,

	A	B	C	D	E	F	G	H	I
1	**Inputs**		**Outputs**				The data sheet selected		
2	Assets		Revenue				Sheet1		
3	Equity		Profit						
4	Employees								
5									
6			*Input-Oriented*						
7			*CRS*	*Cross Efficiency Matrix*					
8	*DMU No.*	*DMU Name*	*Cross Efficiency*	*DMU1*	*DMU2*	*DMU3*	*DMU4*	*DMU5*	*DMU6*
9	1	Mitsubishi	0.43830	0.66283	0.99674	1.00000	1.00000	1.00000	0.85793
10	2	Mitsui	0.60879	0.61749	1.00000	1.00000	1.00000	0.91361	0.84444
11	3	Itochu	0.65913	0.53796	0.95649	1.00000	1.00000	0.77072	0.82021
12	4	General Motors	0.77559	0.10899	0.18251	0.08705	1.00000	0.11031	0.09429
13	5	Sumitomo	0.70363	0.66177	0.96928	1.00000	1.00000	1.00000	0.86002
14	6	Marubeni	0.58743	0.35014	0.18522	1.00000	0.18440	1.00000	0.97197
15	7	Ford Motor	0.50713	0.58368	0.84012	1.00000	1.00000	0.85691	0.84808
16	8	Totota Motor	0.35214	0.60994	0.78561	0.75958	0.48929	1.00000	0.67085
17	9	Exxon	0.89972	0.16247	0.22362	0.12710	1.00000	0.17313	0.14016
18	10	Royal Dutch/Shell Group	0.68892	0.12188	0.04987	0.21388	0.12300	0.43079	0.29614
19	11	Wal-Mart	0.75217	0.11863	0.16646	0.07343	1.00000	0.12583	0.08395
20	12	Hitachi	0.27370	0.60994	0.78561	0.75958	0.48929	1.00000	0.67085
21	13	Nippon Life Insurance	0.56134	0.16247	0.22362	0.12710	1.00000	0.17313	0.14016
22	14	Nippon Telegraph & Telephone	0.27684	0.65990	0.96993	0.97188	1.00000	1.00000	0.83829
23	15	AT&T	0.13677	0.60647	0.79670	0.76730	0.34783	1.00000	0.67408

Fig. 4.4 Cross efficiency

Cell C21
=SUMPRODUCT(INDEX($G$2:$H$16,C20,0),INDEX($P$2:$Q$16, $A$21,0))/SUMPRODUCT(INDEX($C$2:$E$16,C20,0),INDEX($L$2:$N$16, $A$21,0)) and is copied into Cells D21:Q21.

Cell C22 has a slightly different formula
=SUMPRODUCT(INDEX($G$2:$H$16,C20,0),INDEX($P$2:$Q$16, $A$22,0))/SUMPRODUCT(INDEX($C$2:$E$16,C20,0),INDEX($L$2:$N$16, $A$22,0)), and is copied into Cells D21:Q21.

In Fig. 4.3, the leading diagonal shows the original CRS efficiency scores. Cells C37:Q37 calculate cross efficiency scores based upon (4.3). In this particular case, DMU9 (Exxon) has the highest cross efficiency score of 0.848, followed by DMU5 (Suitomo).

To obtain the cross efficiency scores using *DEAFrontier* software, the user selects the "Cross Efficiency" menu item. This function will generate the cross efficiency scores ("Cross Efficiency Report", as shown in Fig. 4.4) along with results generated by the associated DEA multiplier model.

4.4 Game Cross Efficiency

As we can see from Fig. 4.3 and Fig. 4.4, the standard DEA cross efficiency scores are not unique. In this section, we present the Game Cross Efficiency developed by Liang et al. (2008) to address the non-uniqueness issue.

As pointed out by Liang et al. (2008), in many DEA applications, some form of direct or indirect competition may exist among the DMUs under evaluation. Certainly any setting where DMUs compete for scarce funds, competition is present by definition. R&D project proposals submitted by different departments in an organization can be viewed as DMUs, and subjected to a DEA analysis. These proposals are clearly competing for available funds. Candidates in a preferential election setting can be looked upon as DMUs, and competition is obviously present. An academic applying for research grants is in competition with other academics. Participants in organized sporting events such as the Olympic games, constitute competitive DMUs. When DMUs are viewed as players in a game, cross efficiency scores may be viewed as payoffs, and each DMU may choose to take a non-cooperative game stance to the extent that it will attempt to maximize its (worst possible) payoff.

The idea of game cross efficiency can be presented as follows. For *each* competing DMU_j, a multiplier bundle is determined that optimizes the efficiency score for *j*, with the additional constraint that the resulting score for DMU *d* should be at or above DMU *d*'s estimated best performance, *in a cross-efficiency sense*. In game cross efficiency case, rather than using the ideal score for DMU_d, we strive to use a score which will actually be representative of its final measure of performance. The problem, of course, arises that we will not know this best performance score for *d* until the best performances of all other DMUs are known as well. To combat this "chicken and egg" phenomenon, Liang et al. (2008) adopt an iterative approach that leads to an equilibrium.

4.4.1 Input-oriented Game Cross Efficiency

To make these ideas more concrete, suppose that in a game sense, one player DMU_d is given an efficiency score α_d, and that another player DMU_j then tries to maximize its own efficiency, subject to the condition that α_d cannot be decreased. We define the *game cross efficiency* for DMU_j relative to DMU_d as

$$\alpha_{dj} = \frac{\sum_{r=1}^{s} u_{rj}^{d} y_{rj}}{\sum_{i=1}^{m} v_{ij}^{d} x_{ij}}, \; d = 1,\ 2, \ldots, n \tag{4.8}$$

where u_{rj}^d and v_{ij}^d are optimal weights in the following model (4.9). The subscript *dj* is intended to indicate that DMU_j is permitted only to choose weights that will not deteriorate the currently estimated efficiency of DMU_d. The difference between (4.2) and (4.8) is that weights in (4.8) are not necessarily optimal, but rather are a feasible solution to the CRS multiplier model (4.4). Such a definition allows DMUs to choose (negotiate) a set of weights, (hence a form of cross efficiency scores), that are best for all of the DMUs. So, in this sense, we adopt a non-cooperative game approach.

To calculate the game *d*-cross efficiency defined in (4.8), we consider the following model for each DMU_j

$$
\begin{aligned}
& Max \sum_{r=1}^{s} u_{rj}^{d} y_{rj} \\
& \text{subject to} \\
& \sum_{i=1}^{m} v_{ij}^{d} x_{il} - \sum_{r=1}^{s} u_{rj}^{d} y_{rl} \geq 0, l = 1, 2, ..., n \\
& \sum_{i=1}^{m} v_{ij}^{d} x_{ij} = 1 \\
& \alpha_d \times \sum_{i=1}^{m} v_{ij}^{d} x_{id} - \sum_{r=1}^{s} u_{rj}^{d} y_{rd} \leq 0 \\
& v_{ij}^{d} \geq 0, i = 1, ..., m \\
& u_{rj}^{d} \geq 0, r = 1, ..., s
\end{aligned}
\tag{4.9}
$$

where $\alpha_d \leq 1$ is a parameter. This model (4.9) is very similar to the CRS multiplier model (4.4), except for the additional constraint of $\alpha_d \times \sum_{i=1}^{m} v_{ij}^{d} x_{id} - \sum_{r=1}^{s} u_{rj}^{d} y_{rd} \leq 0$ which ensures that the (cross) efficiency score of DMU_d cannot be less than α_d.

This α_d initially takes the value given by the average original cross efficiency of DMU_d. When the algorithm converges, this α_d becomes the best (average) game-cross efficiency score. Model (4.9) is referred to as the *DEA game d-cross efficiency model*. Note that model (4.9) maximizes the efficiency of DMU_j, under the condition that the efficiency of a given DMU_d, is not less than a given value (α_d). Thus, the efficiency of DMU_j is further constrained by the requirement that the ratio efficiency of DMU_d is not less than its original average cross efficiency.

For each DMU_j, model (4.9) is solved *n* times, once for each $d=1, \ldots, n$. Note that for each *d*, at optimality, $\sum_{i=1}^{m} v_{ij}^{d} x_{ij} = 1$ holds for DMU_j ($j=1, 2, \ldots, n$). Therefore, for each DMU_j, the optimal value to model (4.9) actually represents a game cross efficiency with respect to DMU_d(*d*-game cross efficiency), as defined in (4.8). We have

Definition 4.1 Let $u_{rj}^{d*}(\alpha_d)$ be an optimal solution to model (4.9). For each DMU_j, $\alpha_j = \frac{1}{n} \sum_{d=1}^{n} \sum_{r=1}^{s} u_{rj}^{d*}(\alpha_d) y_{rj}$ is called the input-oriented (average) *game cross efficiency* for DMU_j.

Note that the average game cross efficiency no longer represents a regular DEA cross efficiency value. Liang et al. (2008) show that optimal game cross efficiency scores constitute a Nash Equilibrium point.

Table 4.1 Game cross efficiency example

	X_1	X_2	X_3	Y_1	Y_2	CRS efficiency	Game cross efficiency[a]
DMU_1	7	7	7	4	4	0.6857	0.63813
DMU_2	5	9	7	7	7	1	0.97638
DMU_3	4	6	5	5	7	1	1
DMU_4	5	9	8	6	2	0.8571	0.79833
DMU_5	6	8	5	3	6	0.8571	0.66659

[a] In the algorithm, we set $\delta = 0.001$

We now present the procedure for determining the *best* average input-oriented game-cross efficiency for DMU_j, as described in Liang et al. (2008).

Algorithm **Step 1:** Solve model (4.4) and obtain a set of original DEA cross efficiency scores $\overline{E_d}$ defined in (4.3). Let $t = 1$ and $\alpha_d = \alpha_d^1 = \overline{E_d}$.

Step 2: Solve model (4.9). Let $\alpha_j^2 = \frac{1}{n}\sum_{d=1}^{n}\sum_{r=1}^{s} u_{rj}^{d*}(\alpha_d^1) y_{rj}$ or in a general format,

$$\alpha_j^{t+1} = \frac{1}{n}\sum_{d=1}^{n}\sum_{r=1}^{s} u_{rj}^{d*}(\alpha_d^t) y_{rj}. \tag{4.10}$$

where $u_{rj}^{d*}(\alpha_d^t)$ represents optimal value of u_{rj}^d in model (4.9) when $\alpha_d = \alpha_d^t$.

Step 3: If $|\alpha_j^{t+1} - \alpha_j^t| \geq \delta$ for some j, where ε is a specified small positive value, then let $\alpha_d = \alpha_d^{t+1}$ and go to Step 2. If $|\alpha_j^{t+1} - \alpha_j^t| < \delta$ for all j, then stop. α_j^{t+1} is the best average game-cross efficiency given to DMU_j. (In calculation, we can set $\delta = 0.001$, for example.)

In Step 1, the $\overline{E_d}$ represent traditional (average) cross efficiency scores for $DMU_d, d = 1, 2, \ldots, n$, and are the initial values for α_d (denoted as α_d^1) in model (4.9). Although the cross efficiency scores may not be unique, Liang et al. (2008) show that any initial values for α_d (or any traditional cross efficiency scores), will lead to unique game-cross efficiency scores. When the algorithm stops, since $\sum_{r=1}^{s} u_{rj}^{d*}(\alpha_d^t) y_{rj}$ is the optimal value to model (4.9), $\alpha_j^{t+1} = \frac{1}{n}\sum_{d=1}^{n}\sum_{r=1}^{s} u_{rj}^{d*}(\alpha_d^t) y_{rj}$, $t \geq 1$ is unique. Also, the notation $\alpha_d = \alpha_d^t$, $t \geq 1$, given in Step 2, means that in model (4.9) α_d is replaced with α_d^t. Step 3 is used to indicate when to terminate the process of executing model (4.9).

To illustrate the game efficiency model using spreadsheets, we consider the numerical example in Liang et al. (2008) where we have five DMUs, with three inputs X_1, X_2, X_3 and two outputs Y_1, Y_2. Table 4.1 presents the data along with the results.

In the algorithm, we use the regular cross efficiency as the starting point for our game cross efficiency scores. Table 4.2 shows a cross efficiency matrix along with cross efficiency scores shown in column 2.

Table 4.2 Input-oriented cross efficiency matrix

	CRS	Cross efficiency matrix				
DMU	*Cross efficiency*	*DMU1*	*DMU2*	*DMU3*	*DMU4*	*DMU5*
DMU1	0.54531	0.68571	0.93333	1.00000	0.80000	0.45000
DMU2	0.86286	0.57143	1.00000	1.00000	0.85714	0.42857
DMU3	1.00000	0.48980	0.66667	1.00000	0.19048	0.64286
DMU4	0.57667	0.57143	1.00000	1.00000	0.85714	0.42857
DMU5	0.56143	0.40816	0.71429	1.00000	0.17857	0.85714

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		X_1	X_2	X_3		Y_1	Y_2			Constraints		α		average α t
2	DMU_1	7	7	7		4	4			-0.828571429		0.54531		0.666922643
3	DMU_2	5	9	7		7	7			-0.4		0.86286		1
4	DMU_3	4	6	5		5	7			0		1.00000		1
5	DMU_4	5	9	8		6	2			-1.314285714		0.57667		0.839005363
6	DMU_5	6	8	5		3	6			-0.142857143		0.56143		0.751371429
7														
8	Multiplier	0	0	0.2		0	0.142857			1				
9														
10														
11	DMUj under evaluation		5			DMUd	5							
12														
13										Game Cross Efficiency				
14	Cross Efficiency (Max)	0.857143								0.641142857				
15										0.723428571				
16	cross efficiency	-0.29571								0.857142857				
17	constraint on α									0.678				
18										0.857142857				
19									average	0.751371429				

Fig. 4.5 Input-oriented game cross efficiency model

These DEA cross efficiency scores are then used to calculate model (4.9) in the initial step. Figure 4.5 shows the spreadsheet model for model (4.9). Column L stores cross efficiency scores that are used as $\alpha_d^1 = \overline{E}_d$. Column J contains formulas for the regular CRS multiplier model constraints. For example, Cell J2 has the formula

=SUMPRODUCT(F2:G2,$F$8:$G$8)-SUMPRODUCT(B2:D2,$B$8:$D$8)

Cell C11 is reserved to represent DMUj and cell G11 is reserved to represent DMUd. Cell B14 represents the objective function of model (4.9) with the following formula

=SUMPRODUCT(INDEX(F2:G6,C11,0),F8:G8)

Cell B16 represents the constraint of $\alpha_d \times \sum_{i=1}^{m} v_{ij}^d x_{id} - \sum_{r=1}^{s} u_{rj}^d y_{rd} \leq 0$ with the following formula

=INDEX(L2:L6,G11,1)*SUMPRODUCT(INDEX(B2:D6,G11,0),B8:D8)-SUMPRODUCT(INDEX(F2:G6,G11,0),F8:G8)

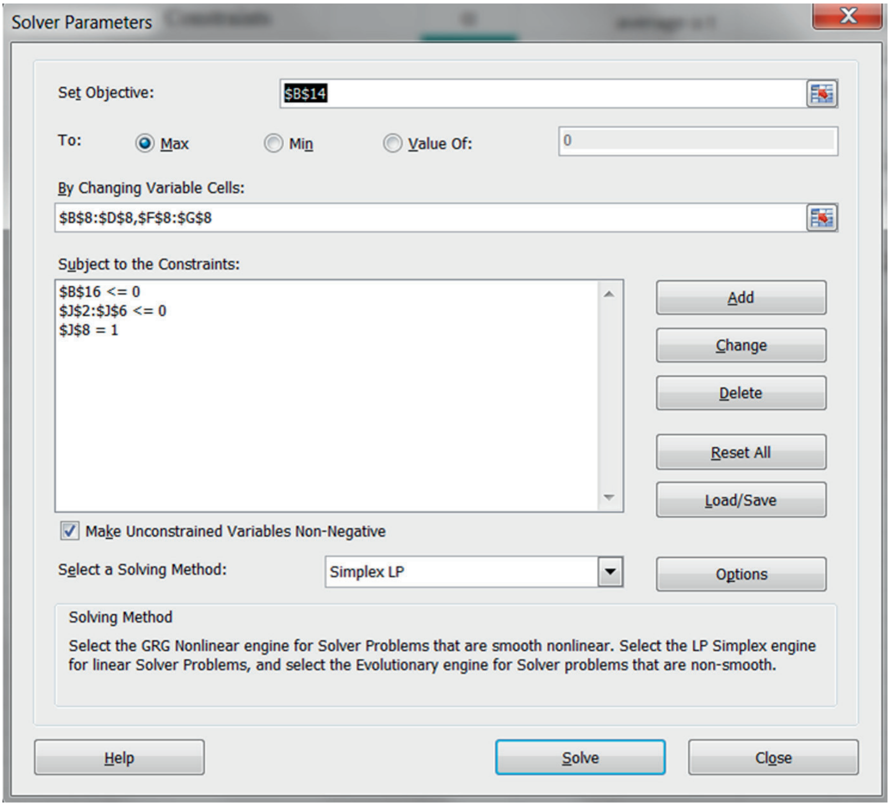


Fig. 4.6 Input-oriented game cross efficiency solver parameters

Cell J8 is used to represent $\sum_{i=1}^{m} v_{ij}^{d} x_{ij}$ which in Solver parameters will be set equal to one. This cell J8 contains the following formula

=SUMPRODUCT(INDEX(B2:D6,C11,0),B8:D8)

For each DMU_j, cells J14:J18 store the optimal values to model (4.9) when DMU_d goes from 1 to 5. The (average) game cross efficiency score for DMUj is then stored in cell J19 with the formula =AVERAGE(J14:J18). Each (average) game cross efficiency score defined in (4.10) is then stored in column N.

The Solver parameters are shown in Fig. 4.6.

In file “gamecross.xlsm”, the following VBA code is used to generate the results shown in Fig. 4.5.

```
Sub GameCrossEfficiency()
   Dim j As Integer, d As Integer
   For j = 1 To 5 'for each DMUj
   Range("C11") = j
   For d = 1 To 5 'for each DMUd, we cacluate the game cross efficiency

   Range("G11") = d
   SolverSolve UserFinish:=True

'record the cross efficiency scores
   Range("J13").Offset(d, 0) = Range("B14")

   Next d
   Range("N1").Offset(j, 0) = Range("J19") 'record the average game cross efficiency score for DMUj

   Next j
End Sub
```

However, the scores shown in column N are not the (final) game cross efficiency score. To carry out the iterations, we use the model shown in Fig. 4.7. Figure 4.7 only shows part of the model (spreadsheet). In column P, we add the absolute difference of $|\alpha_j^{t+1} - \alpha_j^t|$. For example, cell P2 has the formula "=ABS(L2-N2)". Cell P8 reports the maximum of all $|\alpha_j^{t+1} - \alpha_j^t|$, which enables us to easily test for $|\alpha_j^{t+1} - \alpha_j^t| \geq \varepsilon$.

After 10 iterations, the algorithm finds the game cross efficiency scores for the five DMUs, as shown in column N. The following VBA code is used.

```
Sub GameCrossEfficiency_Iteration()

   Dim j As Integer, d As Integer, t  As Integer

   For t = 1 To 1000 'set an upper limit of 1000, for example
   If Range("P8") < 0.001 Then
   Range("Q1") = t
   End
   Else
   Range("L2") = Range("N2")
   Range("L3") = Range("N3")
   Range("L4") = Range("N4")
   Range("L5") = Range("N5")
   Range("L6") = Range("N6")
   End If

   For j = 1 To 5 'for each DMUj
   Range("C11") = j
   For d = 1 To 5 'for each DMUd, we cacluate the game cross efficiency

   Range("G11") = d
   SolverSolve UserFinish:=True

'record the cross efficiency scores
   Range("J13").Offset(d, 0) = Range("B14")
   Next d
   Range("N1").Offset(j, 0) = Range("J19") 'record the average game cross efficiency score for DMUj

   Next j
   Next t
End Sub
```

	L	M	N	O	P	Q	R	S
1	α		average α t		ABS difference	10	report the value of t iteration	
2	0.63861		0.638133849		0.0004744			
3	0.97681		0.976379606		0.00043385			
4	1		1		0			
5	0.79908		0.798333958		0.00074414			
6	0.66747		0.666593983		0.00087223			
7								
8				maximum	0.00087223			

Fig. 4.7 Input-oriented game cross efficiency iteration

4.4.2 Output-oriented Game Cross Efficiency

In a similar manner, we can develop an output-oriented game cross efficiency approach. In this case, we rely on the output-oriented CRS model. First, α_{dj}, *game cross efficiency* for DMU_j relative to DMU_d, is defined as

$$\alpha_{dj} = \frac{\sum_{i=1}^{m} v_{ij}^{d} x_{ij}}{\sum_{r=1}^{s} u_{rj}^{d} y_{rj}}, \; d = 1, 2, \ldots, n \tag{4.11}$$

Similar to model (4.9), we have the following output-oriented model when DMU_j is under evaluation

$$\begin{aligned} & Min \sum_{i=1}^{m} v_{ij}^{d} x_{ij} \\ & \text{subject to} \\ & \sum_{i=1}^{m} v_{ij}^{d} x_{il} - \sum_{r=1}^{s} u_{rj}^{d} y_{rl} \geq 0, l = 1, 2, \ldots, n \\ & \sum_{r=1}^{s} u_{rj}^{d} y_{rj} = 1 \\ & \sum_{i=1}^{m} v_{ij}^{d} x_{id} - \alpha_d \times \sum_{r=1}^{s} u_{rj}^{d} y_{rd} \leq 0 \\ & v_{ij}^{d} \geq 0, i = 1, \ldots, m \\ & u_{rj}^{d} \geq 0, r = 1, \ldots, s \end{aligned} \tag{4.12}$$

where $\alpha_d \geq 1$ is a parameter. This model (4.6) is very similar to the output-oriented CRS multiplier model (4.7), except for the additional constraint of $\sum_{i=1}^{m} v_{ij}^{d} x_{id} - \alpha_d \times \sum_{r=1}^{s} u_{rj}^{d} y_{rd} \leq 0$ which ensures that the (cross) efficiency score of DMU_d cannot be greater than α_d. Note that under output-oriented model, a larger score indicates worse performance.

This α_d initially takes the value given by the (average) original output-oriented cross efficiency of DMU_d. When the algorithm converges, this α_d becomes the best (average) game-cross efficiency score. Model (4.12) is referred to as the output-oriented *DEA game d-cross efficiency model*.

For each DMU_j, model (4.12) is solved n times, once for each $d = 1, \ldots, n$. Note that for each d, at optimality, $\sum_{i=1}^{m} u_{rj}^{d} y_{rj} = 1$ holds for DMU_j ($j = 1, 2, \ldots, n$). Therefore, for each DMU_j, the optimal value to model (4.12) actually represents a game cross efficiency with respect to DMU_d (d-game cross efficiency), as defined in (4.11). We have

Definition 4.2: Let $v_{ij}^{d*}(\alpha_d)$ be an optimal solution to model (4.12). For each DMU_j, $\alpha_j = \frac{1}{n}\sum_{d=1}^{n}\sum_{i=1}^{m} v_{ij}^{d*}(\alpha_d) x_{ij}$ is called the output-oriented (average) *game cross efficiency* for DMU_j.

We now present the procedure for determining the *best* average output-oriented game-cross efficiency for DMU_j,

Algorithm Step 1: Solve model (4.7) and obtain a set of original DEA cross efficiency scores $\overline{E}_d$ defined in (4.3). Let $t = 1$ and $\alpha_d = \alpha_d^1 = \overline{E}_d$.

Step 2: Solve model (4.12). Let $\alpha_j^2 = \frac{1}{n}\sum_{d=1}^{n}\sum_{i=1}^{m} v_{ij}^{d*}(\alpha_d^1) x_{ij}$ or in a general format,

$$\alpha_j^{t+1} = \frac{1}{n}\sum_{d=1}^{n}\sum_{i=1}^{m} v_{ij}^{d*}(\alpha_d^t) x_{ij},$$

where $v_{ij}^{d*}(\alpha_d^t)$ represents optimal value of v_{ij}^{d} in model (4.12) when $\alpha_d = \alpha_d^t$.

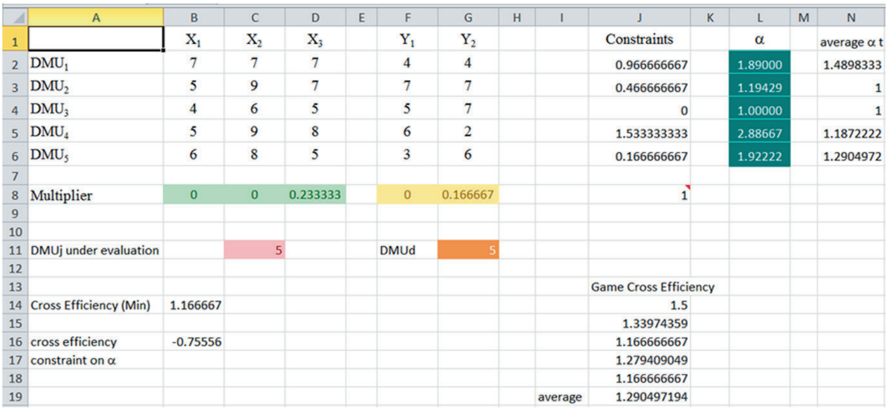
Step 3: If $|\alpha_j^{t+1} - \alpha_j^t| \geq \delta$ for some j, where ε is a specified small positive value, then let $\alpha_d = \alpha_d^{t+1}$ and go to Step 2. If $|\alpha_j^{t+1} - \alpha_j^t| < \delta$ for all j, then stop. α_j^{t+1} is the best output-oriented (average) game-cross efficiency given to DMU_j. (In calculation, we can set $\delta = 0.001$, for example.)

We use the regular output-oriented cross efficiency as the starting point for our game cross efficiency scores. Table 4.3 shows a cross efficiency matrix along with cross efficiency scores shown in column 2.

These DEA cross efficiency scores are then used to calculate model (4.12) in the initial step. Figure 4.8 shows the spread sheet model for model (4.12). This model is very similar to the input-oriented model shown in Fig. 4.5. Column L stores cross efficiency scores that are used as $\alpha_d^1 = \overline{E}_d$. Column J contains formulas for

Table 4.3 Output-oriented Cross Efficiency Matrix

	CRS	Cross efficiency matrix				
DMU	*Cross efficiency*	*DMU1*	*DMU2*	*DMU3*	*DMU4*	*DMU5*
DMU1	1.89000	1.45833	1.07143	1.00000	1.25000	2.22222
DMU2	1.19429	1.75000	1.00000	1.00000	1.16667	2.33333
DMU3	1.00000	2.04167	1.50000	1.00000	5.25000	1.55556
DMU4	2.88667	1.75000	1.00000	1.00000	1.16667	2.33333
DMU5	1.92222	2.45000	1.40000	1.00000	5.60000	1.16667



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		X_1	X_2	X_3		Y_1	Y_2			Constraints		α		average α t
2	DMU_1	7	7	7		4	4			0.966666667		1.89000		1.4898333
3	DMU_2	5	9	7		7	7			0.466666667		1.19429		1
4	DMU_3	4	6	5		5	7			0		1.00000		1
5	DMU_4	5	9	8		6	2			1.533333333		2.88667		1.1872222
6	DMU_5	6	8	5		3	6			0.166666667		1.92222		1.2904972
7														
8	Multiplier	0	0	0.233333		0	0.166667			1				
9														
10														
11	DMUj under evaluation		5			DMUd	5							
12														
13										Game Cross Efficiency				
14	Cross Efficiency (Min)	1.166667								1.5				
15										1.33974359				
16	cross efficiency	-0.75556								1.166666667				
17	constraint on α									1.279409049				
18										1.166666667				
19									average	1.290497194				

Fig. 4.8 Output-oriented CRS game cross efficiency model

the regular output-oriented CRS multiplier model constraints. For example, Cell J2 has the formula

=SUMPRODUCT(B2:D2,$B$8:$D$8)-SUMPRODUCT(F2:G2,$F$8:$G$8)

Cell C11 is reserved to represent DMUj and cell G11 is reserved to represent DMUd. Cell B14 represents the objective function of model (4.12) with the following formula

=SUMPRODUCT(INDEX(B2:D6,C11,0),B8:D8)

Cell B16 represents the constraint of $\sum_{i=1}^{m} v_{ij}^{d} x_{id} - \alpha_d \times \sum_{r=1}^{s} u_{rj}^{d} y_{rd} \leq 0$ with the following formula

=SUMPRODUCT(INDEX(B2:D6,G11,0),B8:D8)-INDEX(L2:L6,G11,1)*SUMPRODUCT(INDEX(F2:G6,G11,0),F8:G8)

Cell J8 is used to represent $\sum_{i=1}^{m} u_{rj}^{d} y_{rj}$ which in Solver parameters will be set equal to one. This cell J8 contains the following formula

=SUMPRODUCT(INDEX(F2:G6,C11,0),F8:G8)

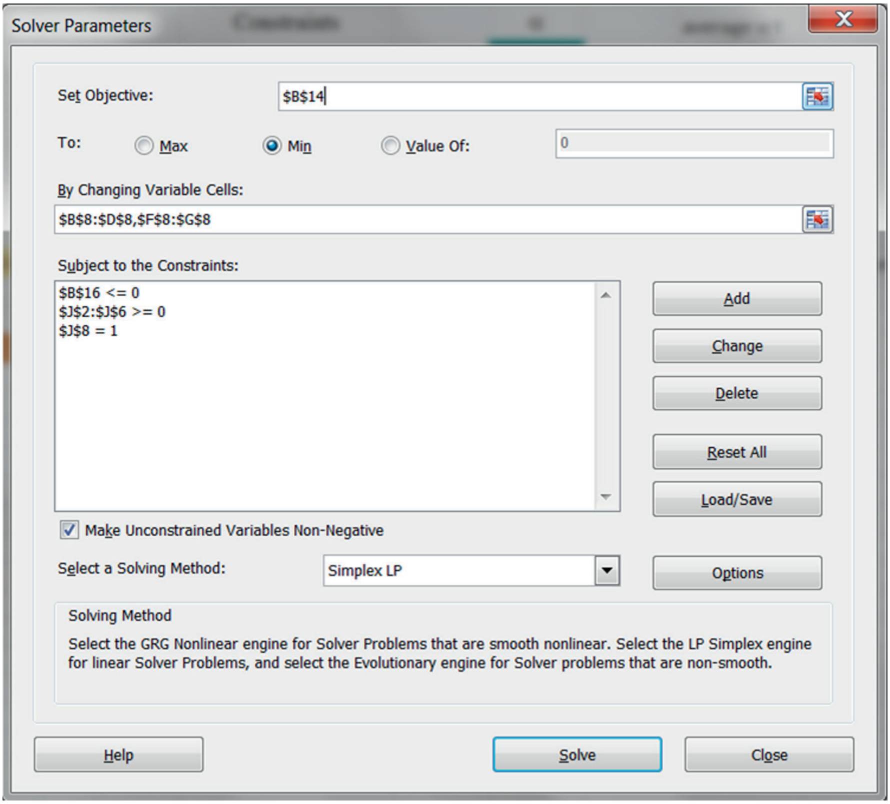


Fig. 4.9 Output-oriented game cross efficiency solver parameters

For each DMU_j, cells J14:J18 store the optimal values to model (4.12) when DMU_d goes from 1 to 5. The (average) game cross efficiency score for DMUj is then stored in cell J19 with the formula =AVERAGE(J14:J18). Each (average) game cross efficiency score is then stored in column N.

The Solver parameters are shown in Fig. 4.9. (see also file "gamecross.xlsm".)

The same VBA code/Macro "GameCrossEfficiency" can be used to generate the initial results shown in in Fig. 4.8 (column N).

However, the scores shown in column N are not the (final) game cross efficiency score. To carry out the iterations, we use a model similar to Fig. 4.7. Figure 4.8 only shows part of the model (spreadsheet). In column P, we add the absolute difference of $|\alpha_j^{t+1} - \alpha_j^t|$. For example, cell P2 has the formula "=ABS(L2-N2)". Cell P8 reports the maximum of all $|\alpha_j^{t+1} - \alpha_j^t|$, which enables us to easily test for $|\alpha_j^{t+1} - \alpha_j^t| \geq \delta$.

We are now ready to run the Macro "GameCrossEfficiency_Iteration" to obtain the output-oriented game cross efficiency scores. After 27 iterations, we obtain the final game cross efficiency scores as shown in column N in Fig. 4.10.

	A	L	M	N	O	P	Q
1		α		average α t		ABS difference	27
2	DMU_1	1.58882		1.5880901		0.00072773	
3	DMU_2	1.01319		1.0130415		0.00014725	
4	DMU_3	1.00000		1		7.7716E-16	
5	DMU_4	1.22878		1.2283306		0.00044958	
6	DMU_5	1.60622		1.605328		0.00089138	
7							
8	Multiplier				maximum	0.00089138	

Fig. 4.10 Output-oriented CRS game cross efficiency iteration

4.4.3 Output-oriented VRS Game Cross Efficiency

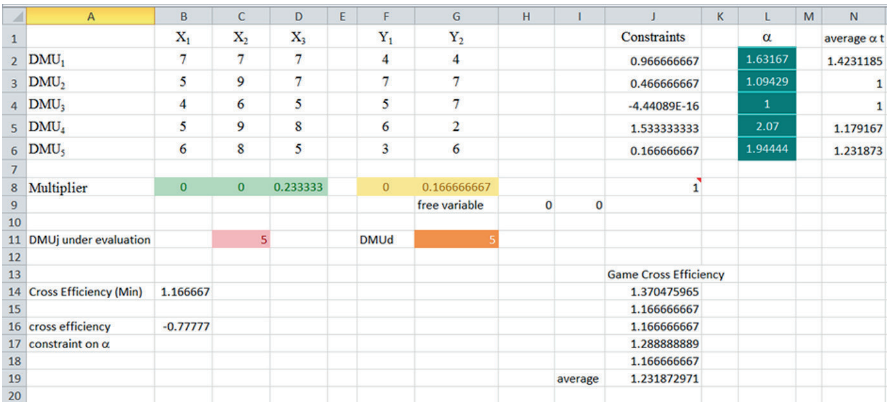
The above discussion is based upon CRS. We can also develop game cross efficiency under the condition of VRS. However, due to the fact that the input-oriented VRS model can yield negative cross efficiency, we here only present the game cross efficiency based upon the output-oriented VRS model where cross efficiency scores are always positive.

The output-oriented VRS game cross efficiency DMU_j relative to DMU_d is given by

$$\alpha_{dj} = \frac{\sum_{i=1}^{m} \omega_{ij}^{d} x_{ij} + v^{d}}{\sum_{r=1}^{s} \mu_{rj}^{d} y_{rj}}, \ d = 1, 2, \ldots, n \tag{4.13}$$

Based upon the output-oriented VRS multiplier model and model (4.12), we have the following VRS model for obtaining the output-oriented VRS game cross efficiency score.

$$\begin{aligned}
& Min \sum_{i=1}^{m} v_{ij}^{d} x_{ij} + v^{d} \\
& subject\ \ to \\
& \sum_{i=1}^{m} v_{ij}^{d} x_{il} - \sum_{r=1}^{s} u_{rj}^{d} y_{rl} + v^{d} \ge 0, l = 1, 2, \ldots, n \\
& \sum_{r=1}^{s} u_{rj}^{d} y_{rj} = 1 \\
& \sum_{i=1}^{m} v_{ij}^{d} x_{id} - \alpha_{d} \times \sum_{r=1}^{s} u_{rj}^{d} y_{rd} + v^{d} \le 0 \\
& v_{ij}^{d} \ge 0, i = 1, \ldots, m \\
& u_{rj}^{d} \ge 0, r = 1, \ldots, s \\
& v^{d}\ free
\end{aligned} \tag{4.14}$$



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		X_1	X_2	X_3		Y_1	Y_2			Constraints		α		average α t
2	DMU_1	7	7	7		4	4			0.966666667		1.63167		1.4231185
3	DMU_2	5	9	7		7	7			0.466666667		1.09429		1
4	DMU_3	4	6	5		5	7			-4.44089E-16		1		1
5	DMU_4	5	9	8		6	2			1.533333333		2.07		1.179167
6	DMU_5	6	8	5		3	6			0.166666667		1.94444		1.231873
7														
8	Multiplier	0	0	0.233333		0	0.166666667			1				
9							free variable	0	0					
10														
11	DMUj under evaluation		5			DMUd	5							
12														
13										Game Cross Efficiency				
14	Cross Efficiency (Min)	1.166667								1.370475965				
15										1.166666667				
16	cross efficiency	-0.77777								1.166666667				
17	constraint on α									1.288888889				
18										1.166666667				
19									average	1.231872971				
20														

Fig. 4.11 Output-oriented VRS Game Cross Efficiency model

where $\sum_{i=1}^{m} v_{ij}^{d} x_{id} - \alpha_d \times \sum_{r=1}^{s} u_{rj}^{d} y_{rd} + v^d \le 0$ ensures that the game cross efficiency score of DMU_d cannot be greater than α_d. Note that under output-oriented model, a larger score indicates worse performance.

The algorithm presented in the previous section can be used to obtain the VRS game cross efficiency. We demonstrate this with the spreadsheet model shown in Fig. 4.11. Figure 4.11 is very similar to Fig. 4.8. The differences are the additional free variable (v^d) and the formulas in cells J2:J6, B14 and B16. We note that the Excel Solver seems to have the ability to just treat the free variable as unconstrained in sign. We however still use two non-negative decision variables (two cells H9 and I9) to treat the free variable (v^d), namely, we express v^d as the difference of two non-negative decision variables.

Now, cell J2 contains the following formulas (with two added cells representing the free variable)

=SUMPRODUCT(B2:D2,$B$8:$D$8)-SUMPRODUCT(F2:G2,$F$8:$G$8) +**$H$9-$I$9**

The objective function cell B14 contains the following formula

=SUMPRODUCT(INDEX(B2:D6,C11,0),B8:D8)+**H9-I9**

Cell B16 representing $\sum_{i=1}^{m} v_{ij}^{d} x_{id} - \alpha_d \times \sum_{r=1}^{s} u_{rj}^{d} y_{rd} + v^d$ contains the following formula

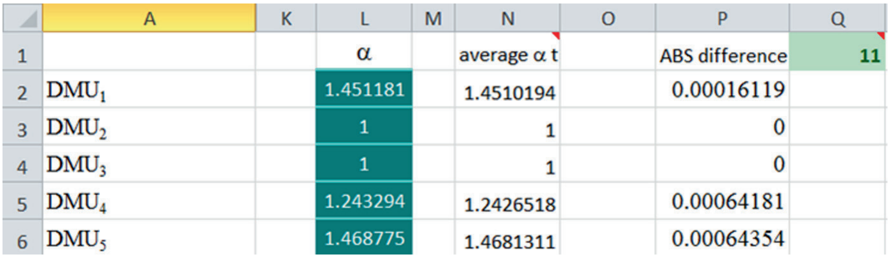
=SUMPRODUCT(INDEX(B2:D6,G11,0),B8:D8)-INDEX(L2:L6,G11,1) *SUMPRODUCT(INDEX(F2:G6,G11,0),F8:G8)+**H9-I9**

Cell J8 has the following formula

=SUMPRODUCT(INDEX(F2:G6,C11,0),F8:G8) which is set equal to one in the Solve parameters.

Table 4.4 Output-oriented VRS cross efficiency matrix

	VRS	Cross efficiency matrix				
DMU	*Cross efficiency*	*DMU1*	*DMU2*	*DMU3*	*DMU4*	*DMU5*
DMU1	1.63167	1.41667	1.00000	1.00000	1.16667	2.11111
DMU2	1.09429	1.41667	1.00000	1.00000	1.16667	2.11111
DMU3	1.00000	1.45833	1.07143	1.00000	1.25000	2.22222
DMU4	2.07000	1.41667	1.00000	1.00000	1.16667	2.11111
DMU5	1.94444	2.45000	1.40000	1.00000	5.60000	1.16667



	A	K	L	M	N	O	P	Q
1			α		average α t		ABS difference	11
2	DMU_1		1.451181		1.4510194		0.00016119	
3	DMU_2		1		1		0	
4	DMU_3		1		1		0	
5	DMU_4		1.243294		1.2426518		0.00064181	
6	DMU_5		1.468775		1.4681311		0.00064354	

Fig. 4.12 Output-oriented VRS Game Cross Efficiency Iteration

We use the regular output-oriented VRS cross efficiency as the starting point for our game cross efficiency scores. Table 4.4 shows a VRS cross efficiency matrix along with cross efficiency scores shown in column 2. These scores are stored in column L.

Once the model is set up, we need to add the two non-negative decision variables (cells H9 and I9) as changing cells in the Solver parameters shown in Fig. 4.9. We then use the Macro “GameCrossEfficiency” to store the initial results in column N.

At this stage, the scores shown in column N are not the (final) game cross efficiency scores. To carry out the iterations, we use the same approach associated with the output-oriented CRS game cross efficiency. The Macro “GameCrossEfficiency_Iteration” can be used obtain the output-oriented VRS game cross efficiency scores. After 11 iterations, we obtain the final output-oriented VRS game cross efficiency scores as shown in column N in Fig. 4.12.

To obtain the game cross efficiency scores using *DEAFrontier* software, the user selects the “Game Cross Efficiency” menu item. The user then selects a sheet storing the data, the frontier type (CRS vs VRS) and the model orientation (input vs output), as shown in Fig. 4.13. The Game Cross Efficiency function will generate (i) regular DEA efficiency scores with a set of optimal multipliers, as shown in Fig. 2.13, for example, (ii) cross efficiency scores in sheet “Cross Efficiency Report”, as shown in Table 4.2, (iii) Game Cross Efficiency Matrix in sheet “Game Cross Efficiency”, as shown in Fig. 4.14, and (iv) a set of optimal multiplier (in sheet “Game Multipliers”) when the last DMU is under evaluation when DMU_d varies from 1 to n, as shown Fig. 4.15. Note that during the last iteration, we have

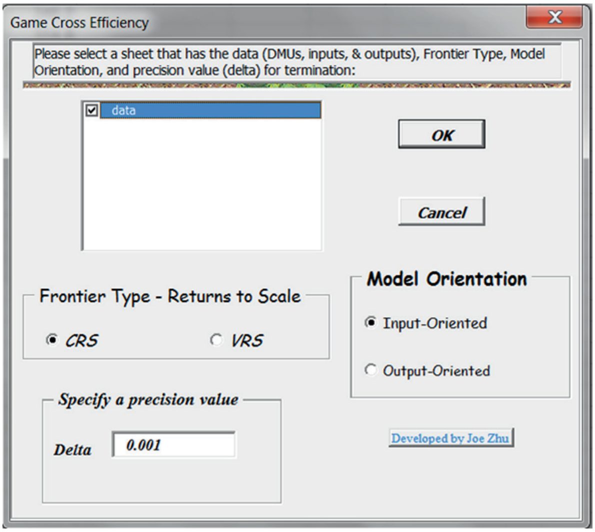


Fig. 4.13 Game cross efficiency

	A	B	C	D	E	F	G	H	I	J	K
1	**Inputs**		**Outputs**	Precision δ			The data sheet selected			No. of Iterations	
2	X1		Y1	**0.001**			**data**			**10**	
3	X2		Y2								
4	X3										
5											
6			*Input-Oriented*	*Game Cross*							
7			*CRS*	*Efficiency Matrix*							
8	*DMU No.*	*DMU Name*	*Game Cross Efficiency*	*DMU1*	*DMU2*	*DMU3*	*DMU4*	*DMU5*			
9	1	DMU1	0.63813	0.68571	0.95686	1.00000	0.82017	0.49637			
10	2	DMU2	0.97638	0.60493	1.00000	1.00000	0.85714	0.62087			
11	3	DMU3	1.00000	0.68571	1.00000	1.00000	0.85714	0.85714			
12	4	DMU4	0.79833	0.68571	1.00000	1.00000	0.85714	0.50145			
13	5	DMU5	0.66659	0.52859	0.92504	1.00000	0.60008	0.85714			

Fig. 4.14 Game cross efficiency matrix

	A	B	C	D	E	F	G	H
1	**Inputs**		**Outputs**	Precision δ			The data sheet select	
2	X1		Y1	**0.001**			**data**	
3	X2		Y2					
4	X3							
5								
6								
7			*CRS*	*Game Multipliers*				
8	*DMU No.*	*DMU Name*	*Input-Oriented*	*X1*	*X2*	*X3*	*Y1*	*Y2*
9	1	DMU1		0.00000	0.12500	0.00000	0.11393	0.02576
10	2	DMU2		0.00000	0.00000	0.20000	0.18377	0.01159
11	3	DMU3		0.00000	0.00000	0.20000	0.00000	0.14286
12	4	DMU4		0.04106	0.04106	0.08502	0.16715	0.00000
13	5	DMU5		0.00000	0.00000	0.20000	0.00000	0.14286

Fig. 4.15 Game cross efficiency matrix

$n \times n$ sets of optimal multipliers, namely one for each DMU_j under DMU_d. DEAFrontier only reports optimal multipliers associated with DMU_n in the last iteration.

4.5 Maximum Log Cross Efficiency

Let us re-visit Fig. 4.1. To address the non-uniqueness in cross efficiency, the idea of secondary goals was introduced, with the original proposal being to maximize or minimize the average appraisal of peers as indicated by A_k in Fig. 4.1. Specifically, A_k is the arithmetic average across the row *k*. However, due to the DEA model (CRS multiplier model) used, $E_{kj} = \frac{\sum_r \mu_{rk} y_{rj}}{\sum_i v_{ik} x_{ij}}$, where y_{rj}, $(r = 1, 2, \ldots, s)$ are outputs and x_{ij}, $(i = 1, 2, \ldots, m)$ are inputs for DMU_j, and μ_{rk}, v_{ik} are corresponding output and input weights chosen by DMU_k.

Thus, $A_k = \sum_j E_{kj}$ appears in the form of a non-linear fractional problem that cannot be converted into linear format. To remedy this, Sexton et al. (1986), and Doyle and Green (1994) suggested the use of linear surrogates for the secondary goal in form of the numerators in E_{kj} minus the sum of the denominators, and modified ratios that can be converted into linear relations. However, due to the fact that these surrogates are not equivalent to the optimal values of A_k, the resulting cross efficiency scores are, at best, approximations of these optimal values.

Cook and Zhu (2014), on the other hand, propose to use multiplicative DEA models developed in Charnes et al. (1982) and Charnes et al. (1983) to obtain maximum (*and unique*) cross efficiency scores under the condition that each DMU's DEA efficiency score remains unchanged. To introduce Cook and Zhu (2014) approach, we need first to present the multiplicative DEA models.

4.5.1 Multiplicative DEA Model

Charnes et al. (1982) introduce the following multiplicative DEA model when DMU_o is under evaluation

$$
\begin{aligned}
& \max \frac{\prod_{r=1}^{s} y_{ro}^{\mu_r}}{\prod_{i=1}^{m} x_{io}^{v_i}} \\
& s.t. \quad \frac{\prod_{r=1}^{s} y_{rj}^{\mu_r}}{\prod_{i=1}^{m} x_{ij}^{v_i}} \le 1, \qquad j = 1, \ldots, n \\
& \mu_r, v_i \ge 1
\end{aligned}
\tag{4.15}
$$

Taking logarithms (to any base), model (4.15) becomes

$$
\begin{aligned}
&\max \sum_{r=1}^{s} \mu_r \hat{y}_{ro} - \sum_{i=1}^{m} v_i \hat{x}_{io} \\
&\text{subject to} \\
&\sum_{r=1}^{s} \mu_r \hat{y}_{rj} - \sum_{i=1}^{m} v_i \hat{x}_{ij} \le 0 \\
&\mu_r, v_i \ge 1
\end{aligned} \tag{4.16}
$$

where (^) denotes logarithms.

The dual to model (4.16) can be written as

$$
\begin{aligned}
&\max \sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+ \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j \hat{x}_{ij} + s_i^- = \hat{x}_{io} \qquad i = 1, 2, \ldots, m; \\
&\sum_{j=1}^{n} \lambda_j \hat{y}_{rj} - s_r^+ = \hat{y}_{ro} \qquad r = 1, 2, \ldots, s; \\
&\lambda_j, s_i^-, s_r^+ \ge 0
\end{aligned} \tag{4.17}
$$

It can be seen that model (4.17) is actually the CRS slack-based model discussed in Chap. 5. We therefore call model (4.15) CRS multiplicative DEA model.

Further, to calculate model (4.15), we can follow the following steps:

1. Take logarithms (e.g., natural logarithms) of the inputs and outputs,
2. Calculate slack-based model (4.17)
3. The efficiency score in model (4.15) is then equal to $e^{\sum_{i=1}^{m} s_i^{-*} + \sum_{r=1}^{s} s_r^{+*}}$

Model (4.15) or (4.16) yields the best efficiency score for DMU_o with a set of "weights" chosen by DMU_o. Denote an optimal set of weights by μ_{ro}^*, v_{io}^*, and the efficiency score from (4.15) as θ_o^*. Then the cross efficiency of DMU_j using the weights that DMU_o has chosen is given by

$$
E_{oj} = \frac{\prod_{r=1}^{s} y_{rj}^{\mu_{ro}^*}}{\prod_{i=1}^{m} x_{ij}^{v_{io}^*}} \tag{4.18}
$$

The efficiency score for DMU_o obtained from model (4.15) is $E_{oo} = \theta_o^*$.

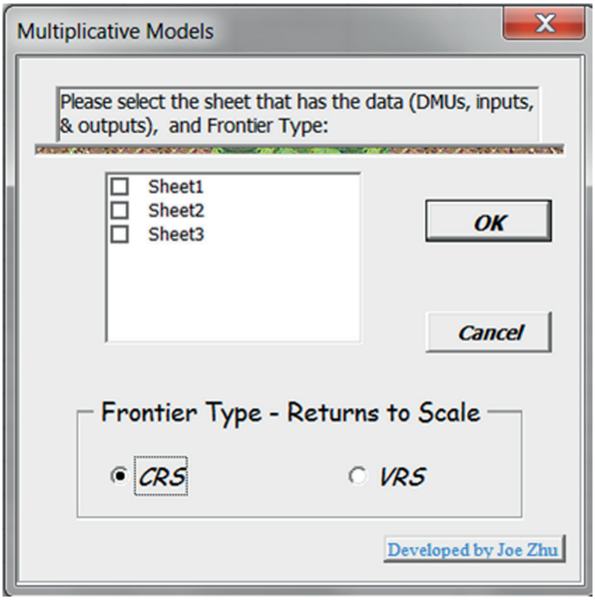


Fig. 4.16 Multiplicative DEA model

Then we define the following *geometric average* peer appraisal cross efficiency score as the CRS multiplicative cross efficiency score

$$\bar{E}_j = \Big(\prod_{k=1}^{n} E_{kj}\Big)^{1/n} = \left(\prod_{k=1}^{n} \frac{\prod_{r=1}^{s} y_{rj}^{\mu_{rk}^*}}{\prod_{i=1}^{m} x_{ij}^{v_{ik}^*}}\right)^{1/n} \tag{4.19}$$

The user can also use the DEAFrontier to perform the calculation for multiplicative DEA model. After the user selects the "Multiplicative Model", the user needs to specify the frontier type, as shown Fig. 4.16. The results are reported in sheet "Multiplicative Efficiency", as shown in Fig. 4.17. DEAFrontier also reports cross efficiency information in sheet "Multiplicative Cross Efficiency" based upon the multiplicative model selected (see Fig. 4.18).

Charnes et al. (1983) introduce the following multiplicative DEA model when DMU_o is under evaluation

$$\begin{aligned}
&\max \frac{e^{\eta}\prod_{r=1}^{s} y_{rk}^{\mu_r}}{e^{\xi}\prod_{i=1}^{m} x_{ik}^{v_i}} \\
&s.t.\quad \frac{e^{\eta}\prod_{r=1}^{s} y_{rj}^{\mu_r}}{e^{\xi}\prod_{i=1}^{m} x_{ij}^{v_i}} \le 1, \qquad j = 1,\dots,n \\
&\eta,\xi \ge 0,\ \mu_r, v_i \ge 1
\end{aligned} \tag{4.20}$$

	DMU No.	DMU Name	CRS Efficiency	Optimal Multipliers Assets	Equity	Employees	Revenue	Profit			Benchmarks
9	1	Mitsubishi	0.15262	1.00000	1.00000	1.00000	1.62271	1.65321	Sumitomo	Exxon	
10	2	Mitsui	1.00000	5.51573	8.95975	1.00000	10.24965	4.49693	Mitsui	Itochu	Sumitomo
11	3	Itochu	1.00000	1.00000	2.10469	1.00000	2.72201	1.00000	Itochu	Sumitomo	
12	4	General Motors	1.00000	9.29665	11.74513	1.00000	5.94305	19.72630	General Motors	Exxon	Wal-Mart
13	5	Sumitomo	1.00000	1.00000	1.00000	1.00000	1.91321	1.00000			Sumitomo
14	6	Marubeni	0.72483	1.00000	2.10469	1.00000	2.72201	1.00000	Itochu	Sumitomo	
15	7	Ford Motor	0.11063	1.00000	1.30017	1.00000	1.76457	1.82840	Sumitomo	Exxon	Nippon Life Insu
16	8	Totota Motor	0.09155	1.00000	1.00000	1.00000	1.62271	1.65321	Sumitomo	Exxon	
17	9	Exxon	1.00000	1.00000	1.00000	1.00000	1.62271	1.65321	Sumitomo	Exxon	
18	10	Royal Dutch/Shell Group	0.49042	1.00000	1.00000	1.00000	1.00000	2.47700			Exxon
19	11	Wal-Mart	1.00000	4.64318	1.00000	1.00000	4.46392	2.63673	Sumitomo	Exxon	Wal-Mart
20	12	Hitachi	0.01858	1.00000	1.00000	1.00000	1.62271	1.65321	Sumitomo	Exxon	
21	13	Nippon Life Insurance	1.00000	1.00000	1.30017	1.00000	1.76457	1.82840	Sumitomo	Exxon	Nippon Life Insu
22	14	Nippon Telegraph & Telephone	0.02557	1.00000	1.00000	1.00000	1.62271	1.65321	Sumitomo	Exxon	
23	15	AT&T	0.00072	1.00000	1.00000	1.00000	1.91321	1.00000	Sumitomo		

Multiplicative Cross Efficiency | Multiplicative Efficiency | DataSheet

Fig. 4.17 Multiplicative efficiency

	A	B	C	D	E	F
1	Inputs		Outputs			
2	Assets		Revenue			
3	Equity		Profit			
4	Employees					

DMU No.	DMU Name	CRS Multiplicative Cross Efficiency	Multiplicative Cross Efficiency Matrix DMU1	DMU2	DMU3
1	Mitsubishi	0.00643	0.15262	0.15081	0.42084
2	Mitsui	0.01992	0.00184	1.00000	1.00000
3	Itochu	0.02126	0.07098	0.15530	1.00000
4	General Motors	0.05487	0.00000	0.00000	0.00000
5	Sumitomo	0.07488	0.11338	0.11868	0.60527
6	Marubeni	0.02192	0.07098	0.15530	1.00000
7	Ford Motor	0.00935	0.14553	0.17301	0.43756
8	Totota Motor	0.00329	0.15262	0.15081	0.42084
9	Exxon	0.18874	0.15262	0.15081	0.42084
10	Royal Dutch/Shell Group	0.04197	0.00992	0.00915	0.01215
11	Wal-Mart	0.07987	0.03625	0.08983	0.09473
12	Hitachi	0.00051	0.15262	0.15081	0.42084
13	Nippon Life Insurance	0.25354	0.14553	0.17301	0.43756
14	Nippon Telegraph & Telephone	0.00051	0.15262	0.15081	0.42084
15	AT&T	0.00000	0.11338	0.11868	0.60527

Fig. 4.18 Multiplicative cross efficiency

Taking logarithms (to any base), model (4.20) becomes

$$
\begin{aligned}
&\max \ \eta - \xi + \sum_{r=1}^{s} \mu_r \hat{y}_{ro} - \sum_{i=1}^{m} \nu_i \hat{x}_{io} \\
&\text{subject to} \\
&\eta - \xi + \sum_{r=1}^{s} \mu_r \hat{y}_{rj} - \sum_{i=1}^{m} \nu_i \hat{x}_{ij} \le 0 \\
&\eta, \xi \ge 0 \\
&\mu_r, \nu_i \ge 1
\end{aligned}
\qquad (4.21)
$$

where (^) denotes logarithms.

The dual to model (4.19) is

$$\begin{aligned}
&\max \sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+ \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j \hat{x}_{ij} + s_i^- = \hat{x}_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j \hat{y}_{rj} - s_r^+ = \hat{y}_{ro} \qquad r = 1,2,\ldots,s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j, s_i^-, s_r^+ \geq 0
\end{aligned} \tag{4.22}$$

Obviously, model (4.22) is the VRS slack-based model discussed in Chap. 5. We therefore call model (4.20) VRS multiplicative DEA model. The VRS multiplicative cross efficiency score can be defined in a similar manner as in (4.18) and (4.19). Specifically, for a DMU_k under evaluation of model (4.20), we have

$$E_{kj} = \frac{e^{\eta_k^*} \prod_{r=1}^{s} y_{rj}^{\mu_{rk}^*}}{e^{\xi_k^*} \prod_{i=1}^{m} x_{ij}^{v_{ik}^*}} \tag{4.23}$$

as cross efficiency of DMU_j using the weights that DMU_k has chosen, where $\eta_k^*, \xi_k^*, \mu_{rk}^*, v_{ik}^*$ are optimal solutions from model (4.20).

Then we define the following *geometric average* peer appraisal VRS multiplicative cross efficiency score

$$\overline{E}_j = \left(\prod_{k=1}^{n} E_{kj}\right)^{1/n} = \left(\prod_{k=1}^{n} \frac{e^{\eta_k^*} \prod_{r=1}^{s} y_{rj}^{\mu_{rk}^*}}{e^{\xi_k^*} \prod_{i=1}^{m} x_{ij}^{v_{ik}^*}} \right)^{1/n} \tag{4.24}$$

where E_{kk} is the optimal value to model (4.20).

The above two multiplicative DEA models identify Cobb-Douglas production functions directly from observations (see Charnes et al. (1982, 1983) for more discussions.)

4.5.2 Maximum Log Cross Efficiency

We next present the approach developed in Cook and Zhu (2014). We use model (4.15) as an example.

Note that the multiplicative cross efficiency defined in (4.19) and (4.24) is not unique due to multiple optimal solutions. However, Cook and Zhu (2014) point out that one can maximize the average cross efficiency score $\bar{E}_j$ subject to the condition that $E_{kk} = \theta_k^*$ for all $k = 1,\ldots,n$. Specifically, for DMU_{j_o} associated with (4.19). Namely, we have

$$\begin{aligned}
&\max \left(\prod_{k=1}^{n} \frac{\prod_{r=1}^{s} y_{rj_o}^{\mu_{rk}}}{\prod_{i=1}^{m} x_{ij_o}^{v_{ik}}} \right)^{1/n} \\
&\text{s.t.} \quad \frac{\prod_{r=1}^{s} y_{rj}^{\mu_{rk}}}{\prod_{i=1}^{m} x_{ij}^{v_{ik}}} \le 1, \quad j = 1,\ldots,n, k = 1,\ldots n \qquad (4.25) \\
&E_{kk} = \frac{\prod_{r=1}^{s} y_{rk}^{\mu_{rk}}}{\prod_{i=1}^{m} x_{ik}^{v_{ik}}} = \theta_k^*, k = 1,\ldots,n \\
&\mu_{\text{rk}}, v_{ik} \ge 1, \quad k = 1,\ldots n; i - 1,\ldots,m; r = 1,\ldots,s
\end{aligned}$$

Making logarithmic transformations in (4.25), we arrive at the following linear program

$$\begin{aligned}
&\max \frac{1}{\text{n}} \left(\sum_{k} \sum_{r} \mu_{rk} \hat{y}_{rj_o} - \sum_{k} \sum_{i} v_{ik} \hat{x}_{ij_0} \right) \\
&s.t. \quad \sum_{r=1}^{s} \mu_{rk} \hat{y}_{rj} - \sum_{i=1}^{m} v_{ik} \hat{x}_{ij} \le 0, \quad j, k = 1,\ldots,n \qquad (4.26) \\
&\sum_{r=1}^{s} \mu_{rk} \hat{y}_{rk} - \sum_{i=1}^{m} v_{ik} \hat{x}_{ik} = \ln(\theta_k^*), \quad k = 1,\ldots n, \\
&\mu_{\text{rk}}, v_{ik} \ge 1, \quad k = 1,\ldots n; i - 1,\ldots,m; r = 1,\ldots,s
\end{aligned}$$

where "^" denotes data in logarithmic form. Since logarithms are used in the process, we call this type of cross efficiency (the optimal value to model (4.26)) "Maximum Log Cross Efficiency".

Cook and Zhu (2014) point out that the attractive feature of the proposed multiplicative approach is that the resulting cross efficiency score (the objective function in model (4.26)) is uniquely determined; this is not the case for any of the other approaches taken up to now. Note that while there may as well be alternate optimal solutions μ_{rk}^*, v_{ik}^* yielding this unique optimal value in (4.26), this fact in and of itself is immaterial. It is the uniqueness of the cross efficiency score, not the multipliers that lead to it, that matters.

The above development is based upon CRS. If we use model (4.20) to calculate the maximum log cross efficiency, we have the following VRS maximum log cross efficiency model

$$\begin{aligned} & \max \left(\prod_{k=1}^{n} \frac{e^{\eta_k} \prod_{r=1}^{s} y_{rj_o}^{\mu_{rk}}}{e^{\xi_k} \prod_{i=1}^{m} x_{ij_o}^{v_{ik}}} \right)^{1/n} \\ & \text{s.t.} \quad \frac{e^{\eta_k} \prod_{r=1}^{s} y_{rj}^{\mu_{rk}}}{e^{\xi_k} \prod_{i=1}^{m} x_{ij}^{v_{ik}}} \le 1, \quad j = 1,\dots,n, k = 1,\dots n \\ & E_{kk} = \frac{e^{\eta_k} \prod_{r=1}^{s} y_{rk}^{\mu_{rk}}}{e^{\xi_k} \prod_{i=1}^{m} x_{ik}^{v_{ik}}} = \theta_k^*, k = 1,\dots,n \\ & \eta_k, \xi_k \ge 0,\ \mu_{\mathrm{rk}}, v_{ik} \ge 1, \quad k = 1,\dots n; i = 1,\dots,m; r = 1,\dots s \end{aligned} \tag{4.27}$$

Making logarithmic transformations in (4.27), we arrive at the following linear program

$$\begin{aligned} & \max \frac{1}{\mathrm{n}} \left(\sum_k \eta_k + \sum_k \sum_r \mu_{rk} \hat{y}_{rj_o} - \sum_k \xi_k - \sum_k \sum_i v_{ik} \hat{x}_{ij_0} \right) \\ & s.t. \quad \eta_k + \sum_{r=1}^{s} \mu_{rk} \hat{y}_{rj} - \xi_k - \sum_{i=1}^{m} v_{ik} \hat{x}_{ij} \le 0, \quad j,k = 1,\dots,n \\ & \eta_k + \sum_{r=1}^{s} \mu_{rk} \hat{y}_{rk} - \xi_k - \sum_{i=1}^{m} v_{ik} \hat{x}_{ik} = \ln(\theta_k^*), \quad k = 1,\dots n, \\ & \eta_k, \xi_k \ge 0,\ \mu_{\mathrm{rk}}, v_{ik} \ge 1, \quad k = 1,\dots n; i = 1,\dots,m; r = 1,\dots s \end{aligned} \tag{4.28}$$

where "^" denotes data in logarithmic form.

To demonstrate the above approach, we apply it to the numerical example in Table 4.2. Table 4.5 reports the results from models (4.15) and (4.26) under CRS. Table 4.6 reports the maximum Log cross efficiency matrix under CRS.

Table 4.5 CRS multiplicative results[a]

	X_1	X_2	X_3	Y_1	Y_2	Efficiency model (4.15)	standard cross efficiency (4.19)	Maximum Log cross efficiency (4.26)
DMU_1	7	7	7	4	4	0.1348	0.0543	0.0563
DMU_2	5	9	7	7	7	1	0.6704	0.7603
DMU_3	4	6	5	5	7	1	1	1
DMU_4	5	9	8	6	2	0.1314	0.0841	0.1013
DMU_5	6	8	5	3	6	0.2332	0.0763	0.0763

[a] This is calculated using *DEAFrontier*

Table 4.6 Maximum log cross efficiency matrix under CRS

	Rated DMU				
Rating DMU	1	2	3	4	5
1	*0.1348*	0.6900	1	0.1314	0.1740
2	0.0021	*1*	1	0.0910	0.0016
3	0.1348	1	*1*	0.1314	0.23323
4	0.1348	0.6901	1	*0.1314*	0.1739
5	0.1122	0.5334	1	0.0517	*0.2332*
$\bar{E}_j$ (maximum)	0.0563	0.7603	1	0.1013	0.0763

We next demonstrate how to build Excel spreadsheets for the VRS case, namely model (4.28). The reader can easily obtain the spreadsheet models for the CRS as case, by setting $\eta_k = 0$ and $\xi_k = 0$ for all k (=1,…,n). Note that under VRS, we have the following variables need to be reserved as changing variable cells in Excel (see Fig. 4.19).

1. n (=5) η_k, one for each DMU_k . (cells P7:P11 – these cells are named as “Eta”),
2. n (=5) ξ_k, one for each DMU_k , (cells Q7:Q11 – these cells are named as “Ksi”),
3. m × n (3 × 5 = 15) input multipliers, v_{ik}, one for each input i and each DMU_k. (cells I7:K11 – these cells are named as “InputMultiplier”),
4. s × n (2 × 5 = 10) output multipliers, u_{rk}, one for each output r and each DMU_k. (cells M7:N11 – these cells are named as “OutputMultiplier”).

We take the natural logarithms of the inputs and outputs, and store the data in cells B7:D11 and cells F7:G11, as shown in Fig. 4.20. There are n × n (5 × 5 = 25) constraints of

$$\eta_k + \sum_{r=1}^{s} \mu_{rk} \hat{y}_{rj} - \xi_k - \sum_{i=1}^{m} v_{ik} \hat{x}_{ij}$$

and are formulated in cells B15:F19 which are named as “MaxLogCE_ConSet”. For example, cell B16 has the following formula

	A	I	J	K	L	M	N	O	P	Q
1	Max Log Cross Efficiency									
2										
3	DMU under evaluation									
4										
5		**Input multipliers**				**Output Multipliers**			**ηκ**	**ξκ**
6	DMU Name									
7	DMU1	1	1	1		1	1		1.232144	0
8	DMU2	1	1	1		2.868233	1		0	1.774662
9	DMU3	1	1	1		1	1		1.232144	0
10	DMU4	1	1	1		2.868233	1		0	1.774662
11	DMU5	1	1	1		1	1		1.232144	0

Fig. 4.19 Maximum log cross efficiency variables

	A	B	C	D	E	F	G	H	I
1	Max Log Cross Efficiency			-1.99975	exp	0.135369			
2									
3	DMU under evaluation	1							
4									
5		Inputs				Outputs			Input n
6	DMU Name	X1	X2	X3		Y1	Y2		
7	DMU1	1.94591	1.94591	1.94591		1.386294	1.386294		
8	DMU2	1.609438	2.197225	1.94591		1.94591	1.94591		
9	DMU3	1.386294	1.791759	1.609438		1.609438	1.94591		
10	DMU4	1.609438	2.197225	2.079442		1.791759	0.693147		
11	DMU5	1.791759	2.079442	1.609438		1.098612	1.791759		
12									
13	Log Constraint								
14		DMU 1	DMU 2	DMU 3	DMU 4	DMU 5		Average of a row	
15		-1.833	-2.24988	-1.833	-2.24988	-1.833		**-1.99975**	
16		-0.62861	0	-0.62861	0	-0.62861		**-0.37717**	
17		-2.2E-16	0	0	0	6.66E-16		**8.88E-17**	
18		-2.16905	-1.82843	-2.16905	-1.82843	-2.16905		**-2.03281**	
19		-1.35812	-2.31246	-1.35812	-2.31246	-1.35812		**-1.73986**	
20									
21									
22	Constraints on Efficiency=ln(theta)								
23		-1.833	=	-1.833					
24		0	=	0					
25		0	=	0					
26		-1.82843	=	-1.82843					
27		-1.35812	=	-1.35812					

Fig. 4.20 Maximum log cross efficiency model

=SUMPRODUCT($M$7:$N$7,$F$8:$G$8)−SUMPRODUCT($I$7:$K$7,$B$8:$D$8)+$P$7−$Q$7

Note that cells B15, C16, D17, E 18, and F19 are the left-hand-side of constraints

$$\eta_k + \sum_{r=1}^{s} \mu_{rk} \hat{y}_{rk} - \xi_k - \sum_{i=1}^{m} \nu_{ik} \hat{x}_{ik} = \ln(\theta_k^*), \quad k = 1, \ldots n$$

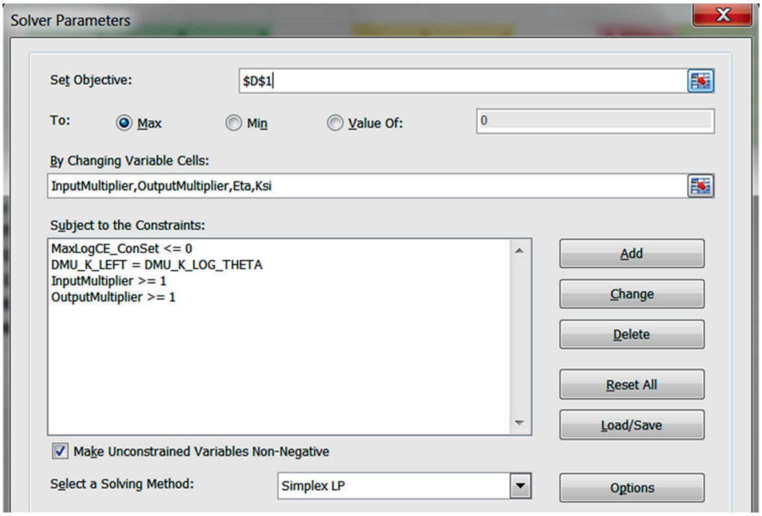


Fig. 4.21 Solver parameters maximum log cross efficiency

The right-hand-side of the above constraints are stored in cellsD23:D27 which are anmed as “DMU_K_LOG_THETA”.

Cells B23:B27 are set equal to cells B15, C16, D17, E 18, and F19, respectively, and are named as “DMU_K_LEFT”.

Now, the average of each row in cells B15:F19 is exactly the objective function, representing the log cross efficiency. These averages are recorded in cells H15:H19.

The objective function is in cell D1 (=INDEX(H15:H19,B3,1)) and the actual maximum Log cross efficiency score is calculated in cell F1 (=EXP(D1)). The Solver parameters are shown in Fig. 4.21.

The reader is referred to file “MaxLog.xlsm” for the above model along with results calculated by using DEAFrontier. To use DEAFrontier, the user selects the menu item “”Maximum Log Cross Efficiency”.

The use can also use the DEAFrontier software to obtain the maximum log cross efficiency. The user select the “Maximum Log-Cross Efficiency” (Fig. 4.22). The results are reported in several sheets as followings (Fig. 4.23):

1. “MaxLog CE Matrix” reports the maximum log cross efficiency;
2. “MaxLog Report” reports the multipliers;
3. “Multiplicative Cross Efficiency” reports the regular multiplicative cross efficiency;
4. “Multiplicative Efficiency” reports the multiplicative efficiency.

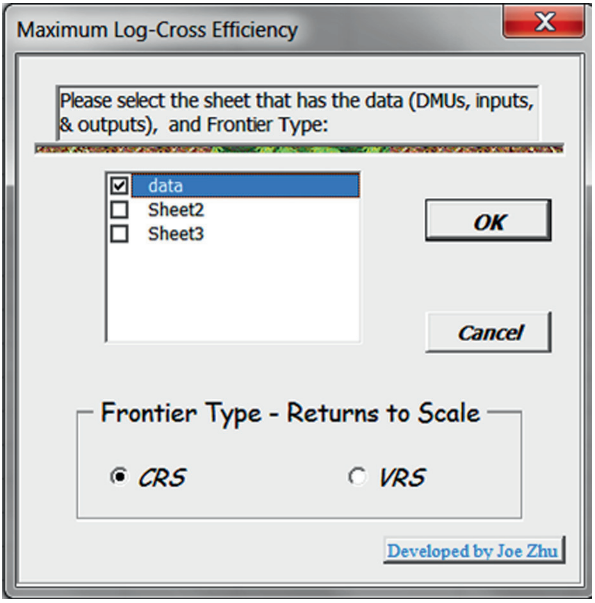
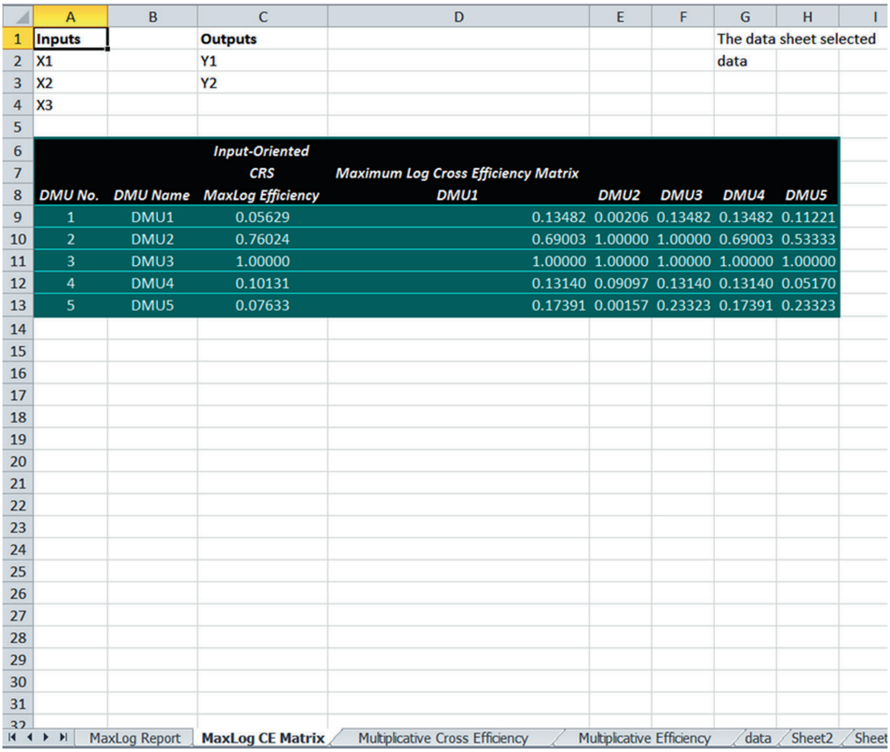


Fig. 4.22 Maximum log cross efficiency in DEAFrontier



	A	B	C	D	E	F	G	H	I
1	Inputs		Outputs				The data sheet selected		
2	X1		Y1				data		
3	X2		Y2						
4	X3								
5									
6			Input-Oriented						
7			CRS	Maximum Log Cross Efficiency Matrix					
8	DMU No.	DMU Name	MaxLog Efficiency	DMU1	DMU2	DMU3	DMU4	DMU5	
9	1	DMU1	0.05629	0.13482	0.00206	0.13482	0.13482	0.11221	
10	2	DMU2	0.76024	0.69003	1.00000	1.00000	0.69003	0.53333	
11	3	DMU3	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	
12	4	DMU4	0.10131	0.13140	0.09097	0.13140	0.13140	0.05170	
13	5	DMU5	0.07633	0.17391	0.00157	0.23323	0.17391	0.23323	

Fig. 4.23 Maximum log cross efficiency

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_4) contains supplementary material, which is available to authorized users.

References

Anderson, T.R., Hollingsworth, K.B., & Inman, L.B. (2002). The fixed weighting nature of a cross-evaluation model. *Journal of Productivity Analysis, 18*(1), 249–255.

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring efficiency of decision making units. *European Journal of Operational Research, 2,* 429–444.

Charnes, A., Cooper, W. W., Seiford, L., & Stutz J. (1982). A multiplicative model for efficiency analysis. *Socio-Economic Planning Sciences, 16*(5), 223–224.

Charnes, A., Cooper, W. W., Seiford, L. M., & Stutz, J. (1983). Invariant multiplicative efficiency and piecewise Cobb-Douglas envelopments. *Operations Research Letters, 2,* 101–103.

Cook, W. D., & Zhu, J. (2014). DEA Cobb-Douglas frontier and cross efficiency. *Journal of Operational Research Society, 65*(2), 265–268.

Doyle, J., & Green, R. (1994). Efficiency and cross efficiency in DEA: Derivations, meanings and the uses. *Journal of the Operational Research Society, 45*(5), 567–578.

Green, R., Doyle, J., & Cook, W. (1996). Preference voting and project ranking using DEA and cross-evaluation. *European Journal of Operational Research, 90,* 461–472.

Liang, L., Wu, J., Cook, W. D., & Zhu, J. (2008). The DEA game cross efficiency model and its Nash equilibrium. *Operations Research, 56,* 1278–1288.

Oral, M., Kettani, O., & Lang, P. (1991). A methodology for collective evaluation and selection of industrial RD projects. *Management Science, 37*(7), 871–883.

Sexton, T. R., Silkman, R. H., & Hogan, A. J. (1986). Data envelopment analysis: Critique and extensions. In: R. H. Silkman (Ed.), *Measuring efficiency: An assessment of data envelopment analysis* (Vol. 32, pp. 73–105). San Francisco: Jossey-Bass.

Chapter 5
Slack-Based DEA Models

5.1 Slack-Based Models

The input-oriented DEA models consider the possible (proportional) input reductions while maintaining the current levels of outputs. The output-oriented DEA models consider the possible (proportional) output augmentations while keeping the current levels of inputs. Charnes et al. (1985) develop an additive DEA model which considers possible input decreases as well as output increases simultaneously. The additive model is based upon input and output slacks. For example,

$$\begin{aligned}
&\max \sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+ \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \qquad i = 1, 2, \ldots, m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \qquad r = 1, 2, \ldots, s; \\
&\lambda_j, s_i^-, s_r^+ \geq 0
\end{aligned} \tag{5.1}$$

Note that model (5.1) assumes equal marginal worth for the nonzero input and output slacks. Therefore, caution should be excised in selecting the units for different input and output measures. Some *a priori* information may be required to prevent an inappropriate summation of non-commensurable measures. Previous management experience and expert opinion, which prove important in productivity analysis, may be used (see, e.g., Seiford and Zhu (1998)).

Model (5.1) therefore is modified to a weighted CRS slack-based model as follows (Ali et al. 1995).

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking,*
International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_5, © Springer International Publishing Switzerland 2014

Table 5.1 Slack-based models

Frontier type	Slack-based DEA Model
CRS	$\max \sum_{i=1}^{m} w_i^- s_i^- + \sum_{r=1}^{s} w_r^+ s_r^+$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \quad i = 1,2,\ldots,m;$ $\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \quad r = 1,2,\ldots,s;$ $\lambda_j, s_i^-, s_r^+ \geq 0$
VRS	Add $\sum_{j=1}^{n} \lambda_j = 1$
NIRS	Add $\sum_{j=1}^{n} \lambda_j \leq 1$
NDRS	Add $\sum_{j=1}^{n} \lambda_j \geq 1$

$$
\begin{aligned}
&\max \sum_{i=1}^{m} w_i^- s_i^- + \sum_{r=1}^{s} w_r^+ s_r^+ \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \qquad r = 1,2,\ldots,s; \\
&\lambda_j, s_i^-, s_r^+ \geq 0
\end{aligned}
\tag{5.2}
$$

where w_i^- and w_r^+ are user-specified weights obtained through value judgment. The DMU_o under evaluation will be termed efficient *if and only if* the optimal value to (5.2) is equal to zero. Otherwise, the nonzero optimal s_i^{-*} identifies an excess utilization of the *i*th input, and the non-zero optimal s_r^{+*} identifies a deficit in the *r*th output. Thus, the solution of (5.2) yields the information on possible adjustments to individual outputs and inputs of each DMU. Obviously, model (5.2) is useful for setting targets for inefficient DMUs with *a priori* information on the adjustments of outputs and inputs.

One should note that model (5.2) does not necessarily yield results that are different from those obtained from the model (5.1). In particular, it will not change the classification from efficient to inefficient (or vice versa) for any DMU.

Model (5.1) identifies a CRS frontier, and therefore is called CRS slack-based model. Table 5.1 summarizes the slack-based models in terms of the frontier types.

	A	B	C	D	E	F	G	H	I
1	Company	Assets	Equity	Employees		Revenue	Profit		λ
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		0
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0.2901
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		1.4476
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0
8	Ford Motor	243283	24547	346990		137137	4139		0
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0
10	Exxon	91296	40436	82000		110009	6470		0.001
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0
12	Wal-Mart	37871	14762	675000		93627	2740		0
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0
16	AT&T	88884	17274	299300		79609	139		0
17									
18		Reference		DMU under	1				
19	**Constraints**	set		Evaluation		**Slack**	**Weights**		
20	Assets	91920.6	=	91920.6		0	1		
21	Equity	10950	=	10950		0	1		
22	Employees	36000	=	36000		24871.42	1		
23	Revenue	184365.2	=	184365.2		107334.9	1		
24	Profit	346.2	=	346.2		0	1		
25						132206.4			

Fig. 5.1 CRS Slack-based DEA spreadsheet model

We should point out that the slack-based models here are usually called "additive DEA models". Tone (2011) develops several different versions of slacks-based measure of efficiency.

5.2 Slack-Based Models in Spreadsheets

Figure 5.1 shows a spreadsheet model for the CRS slack-based model when DMU1 is under evaluation. Cells I2:I16 are reserved for λ_j. Cells F20:F24 are reserved for input and output slacks. The weights on slacks are entered into Cells G20:G24. Currently, the weights are all equal to one.

Cells B20:B24 contain the following formulas

Cell B20 = SUMPRODUCT(B2:B16,$I$2:$I$16)+F20
Cell B21 = SUMPRODUCT(C2:C16,$I$2:$I$16)+F21
Cell B22 = SUMPRODUCT(D2:D16,$I$2:$I$16)+F22
Cell B23 = SUMPRODUCT(F2:F16,$I$2:$I$16)-F23
Cell B24 = SUMPRODUCT(G2:G16,$I$2:$I$16)-F24

The input and output values of the DMU under evaluation are placed into cells D20:D24 via the following formulas

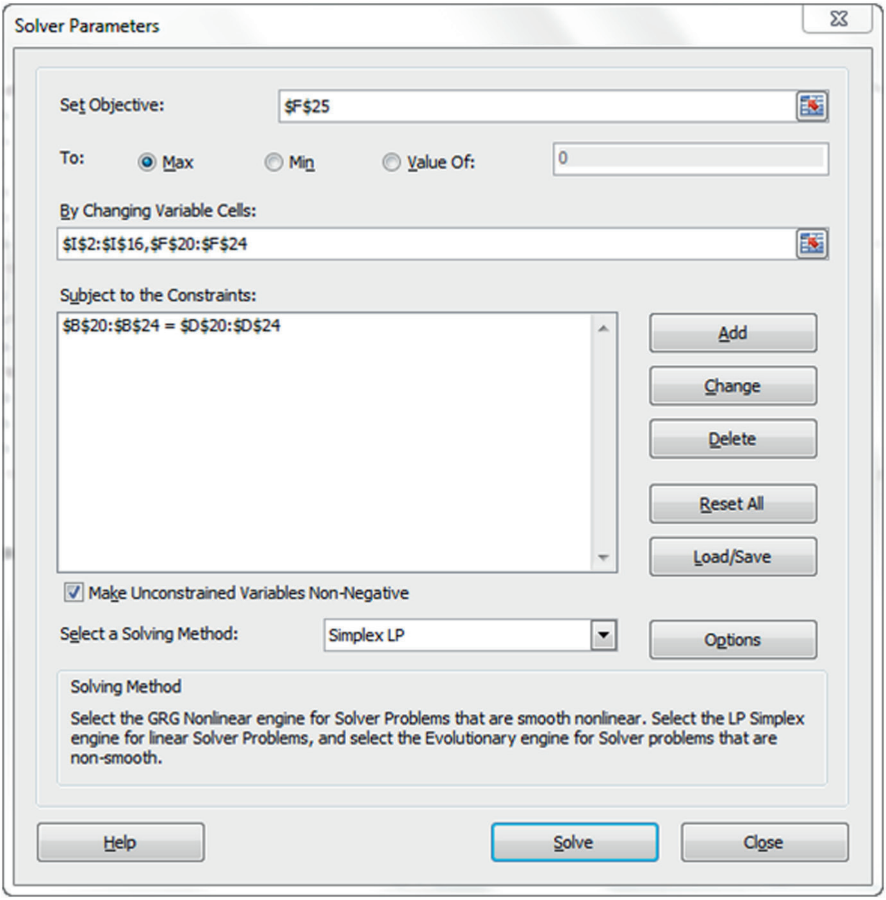


Fig. 5.2 Solver parameters for CRS Slack-based model

Cell D20 = INDEX(B2:B16,E18,1)
Cell D21 = INDEX(C2:C16,E18,1)
Cell D22 = INDEX(D2:D16,E18,1)
Cell D23 = INDEX(F2:F16,E18,1)
Cell D24 = INDEX(G2:G16,E18,1)

Cell F25 is the objective cell which represents the weighted slack. The formula for cell F25 is

Cell F25 = SUMPRODUCT(F20:F24,G20:G24)

Figure 5.2 shows the Solver parameters. Figure 5.3 shows the optimal slack values when DMU1 is under evaluation. Next, we insert a VBA procedure "CRSSlack" to calculate the optimal slacks for the remaining DMUs.

	J	K	L	M	N	O
1	Company	Assets	Equity	Employees	Revenue	Profit
2	Mitsubishi	0	0	24871.423	107334.9	0
3	Mitsui	0	0	-3.988E-10	0	0
4	Itochu	3.69E-11	0	0	-8.6E-14	1.75E-13
5	General Motors	0	0	-1.649E-10	0	0
6	Sumitomo	0	2.27E-12	1.99E-13	-2.1E-13	-7.1E-14
7	Marubeni	12794.2	0	0	5857.514	0.812974
8	Ford Motor	0	0	267229.26	112833	0
9	Toyota Motor	0	25289.03	108011.55	171436.8	0
10	Exxon	0	8.49E-12	2.118E-11	5.85E-11	0
11	Royal Dutch/Shell Group	0	13499.36	14957.88	78912.18	0
12	Wal-Mart	0	0	0	0	0
13	Hitachi	0	12685.39	307952.81	186551.3	0
14	Nippon Life Insurance	-1E-10	0	0	-1.9E-10	0
15	Nippon Telegraph & Telephone	0	17554.49	196254.33	288061.8	0
16	AT&T	0	5460.851	288349.72	216613.9	233.1999

Fig. 5.3 CRS slacks

```
Sub CRSSlack()
Dim i As Integer
    For i = 1 To 15
'set the value of cell E18 equal to i (=1, 2,..., 15)
    Range("E18") = i
'Run the Slack Solver model
    SolverSolve UserFinish:=True
'Select the cells containing the slacks
    Range("F20:F24").Select
'record optimal slacks in cells K2:O16
Selection.Copy
Range("K" & i + 1).Select
Selection.PasteSpecial Paste:=xlPasteValues, Transpose:=True
Next
End Sub
```

By adding an additional constraint on $\sum_{j=1}^{n} \lambda_j$, we can obtain spreadsheet models for other slack-based models (see Excel file slack-based.xlsm). For example, Fig. 5.4 shows a spreadsheet model for the VRS slack-based DEA model.

Range names are used in Fig. 2.13. Cells B2:D16 are named "InputUsed" and cells F2:G16 are named "OutputProduced". We also name cells I2:I16 "Lambdas", cells F20:F24 "Slacks", G20:G24 "Weights", and cell E18 "DMU". Accordingly, we have formulas

	A	B	C	D	E	F	G	H	I
1	Company	Assets	Equity	Employees		Revenue	Profit		λ
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		1
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0
8	Ford Motor	243283	24547	346990		137137	4139		0
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0
10	Exxon	91296	40436	82000		110009	6470		0
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0
12	Wal-Mart	37871	14762	675000		93627	2740		0
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0
16	AT&T	88884	17274	299300		79609	139		0
17									
18		Reference		DMU under	1				
19	**Constraints**	set		Evaluation		**Slack**	**Weights**		
20	Assets	91920.6	=	91920.6		0	1		
21	Equity	10950	=	10950		0	1		
22	Employees	36000	=	36000		0	1		
23	Revenue	184365.2	=	184365.2		0	1		
24	Profit	346.2	=	346.2		0	1		
25	Σλ	1	=	1		0			

Fig. 5.4 VRS Slack-based spreadsheet model

Cell B20 = SUMPRODUCT(INDEX(InputUsed,0,1),Lambdas)+Slacks
Cell B21 = SUMPRODUCT(INDEX(InputUsed,0,2),Lambdas)+Slacks
Cell B22 = SUMPRODUCT(INDEX(InputUsed,0,3),Lambdas)+Slacks
Cell B23 = SUMPRODUCT(INDEX(OutputProduced,0,1),Lambdas)-Slacks
Cell B24 = SUMPRODUCT(INDEX(OutputProduced,0,2),Lambdas)-Slacks
Cell B25 = SUM(Lambdas)
Cell F25 = SUMPRODUCT(Slacks, Weights)

We then name cells B20:B24 "ReferenceSet", cells D20:D24 "DMUEvaluation", B25 "SumLambda", and cell F25 "SumSlack". Figure 5.5 shows the Solver parameters for the VRS slack-based model.

Since range names are used in the Solver model, we can modify "CRSSlack" into a VBA procedure that can be applied to other data sets. The modified VBA procedure is called "Slack".

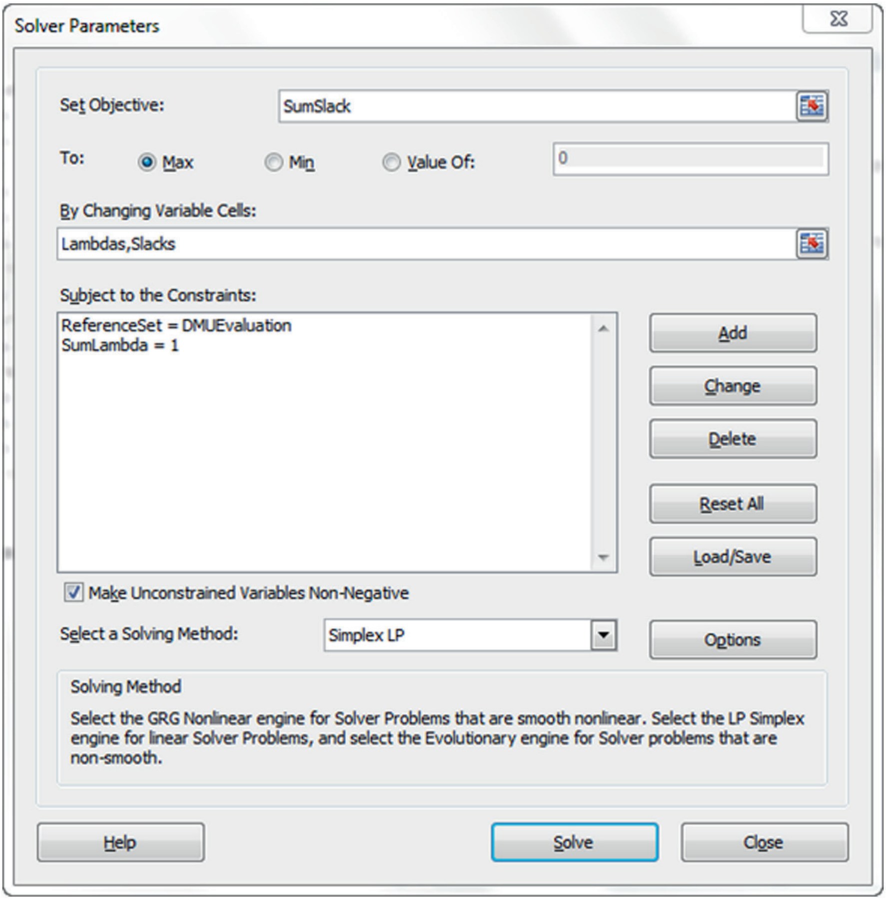


Fig. 5.5 Solver parameters for VRS Slack-based model

```
Sub Slack()
Dim NDMUs As Integer, NInputs As Integer, NOutputs As Integer
    NDMUs = 15
    NInputs = 3
    NOutputs = 2
Dim i As Integer
For i = 1 To NDMUs
Range("DMU") = i
SolverSolve UserFinish:=True
Range("Slacks").Copy
Range("A1").Offset(i, NInputs + NOutputs + 5).Select
Selection.PasteSpecial Paste:=xlPasteValues, Transpose:=True
Next
End Sub
```

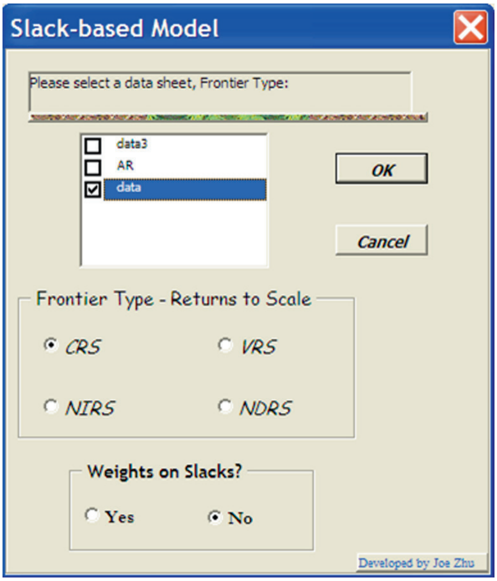


Fig. 5.6 Undesirable measure models

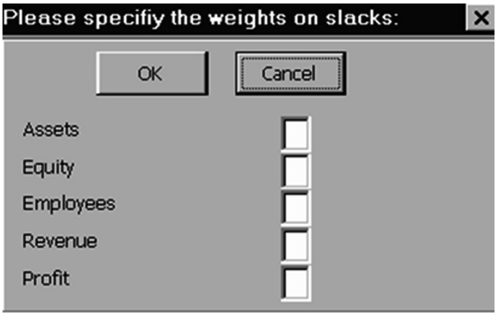


Fig. 5.7 Weights on slacks

5.3 Solving Slack-Based DEA Model Using DEAFrontier Software

To run the slack-based models, select the "Slack-based Model" menu item. You will be prompted with a form for selecting the models presented in Table 5.1, as shown in Fig. 5.6.

If you select "Yes" under the "Weights on Slacks", you will be asked to provide the weights, as shown in Fig. 5.7. If you select "No", then all the weights are set equal to one.

The results are reported in a sheet named "Slack Report" along with a sheet named "Efficient Target".

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_5) contains supplementary material, which is available to authorized users.

References

1. Ali, A. I., Lerme, C. S., & Seiford, L. M. (1995). Components of efficiency evaluation in data envelopment analysis. *European Journal of Operational Research, 80,* 462–473.
2. Charnes, A., Cooper, W. W., Golany, B., Seiford, L. M., & Stutz, J. (1985). Foundations of data envelopment analysis for Pareto-Koopman's efficient empirical production functions. *Journal of Econometrics, 30,* 1–17.
3. Seiford, L. M., & Zhu, J. (1998). Identifying excesses and deficits in Chinese industrial productivity (1953–1990): A weighted data envelopment analysis approach. *OMEGA, 26*(2), 279–269.
4. Tone, K. (2011). Slacks-based measure of efficiency. In W. W. Cooper, L. M. Seiford, & J. Zhu (Eds.), *Handbook on data envelopment analysis*. New York: Springer.

Chapter 6
Measure-Specific DEA Models

6.1 Measure-Specific Models

Although DEA does not need *a priori* information on the underlying functional forms and weights among various input and output measures, it assumes proportional improvements of inputs or outputs. This assumption becomes invalid when a preference structure over the improvement of different inputs (outputs) is present in evaluating (inefficient) DMUs (see also Chap. 7). We need models where a particular set of performance measures is given pre-emptive priority to improve.

Let $I \subseteq \{1,2, \ldots, m\}$ and $O \subseteq \{1,2, \ldots, s\}$ represent the sets of specific inputs and outputs of interest, respectively. Based upon the envelopment models, we can obtain a set of measure-specific models where only the inputs associated with I or the outputs associated with O are optimized (see Table 6.1).

The measure-specific models can be used to model uncontrollable inputs and outputs (see Banker and Morey 1986). The controllable measures are related to set I or set O.

A DMU is efficient under envelopment models if and only if it is efficient under measure-specific models. i.e., both the measure-specific models and the envelopment models yield the same frontier. However, for inefficient DMUs, envelopment and measure-specific models yield different efficient targets.

Consider Fig. 6.1. If the total supply chain cost input is of interest, then the measure-specific model yields the efficient target of S1 for inefficient S. If the response time input is of interest, S3 will be the target for S. The envelopment model projects S to S2 by reducing the two inputs proportionally.

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_6,

Table 6.1 Measure-specific models

Frontier type	Input-oriented	Output-oriented
CRS	$\min \theta - \varepsilon\left(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+\right)$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = \theta x_{io} \quad i \in I;$ $\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \quad i \notin I;$ $\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \quad r = 1,2,\ldots,s;$ $\lambda_j \geq 0 \quad j = 1,2,\ldots,n.$	$\max \phi + \varepsilon\left(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+\right)$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \quad i = 1,2,\ldots,m;$ $\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = \phi y_{ro} \quad r \in O;$ $\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \quad r \notin O;$ $\lambda_j \geq 0 \quad j = 1,2,\ldots,n.$
VRS	Add $\sum_{j=1}^{n} \lambda_j = 1$	
NIRS	Add $\sum_{j=1}^{n} \lambda_j \leq 1$	
NDRS	Add $\sum_{j=1}^{n} \lambda_j \geq 1$	
Efficient target	$\begin{cases} \hat{x}_{io} = \theta^* x_{io} - s_i^{-*} & i \in I \\ \hat{x}_{io} = x_{io} - s_i^{-*} & i \notin I \\ \hat{y}_{ro} = y_{ro} + s_r^{+*} & r = 1,2,\ldots,s \end{cases}$	$\begin{cases} \hat{x}_{io} = x_{io} - s_i^{-*} & i = 1,2,\ldots,m \\ \hat{y}_{ro} = \phi^* y_{ro} + s_r^{+*} & r \in O \\ \hat{y}_{ro} = y_{ro} + s_r^{+*} & r \notin O \end{cases}$

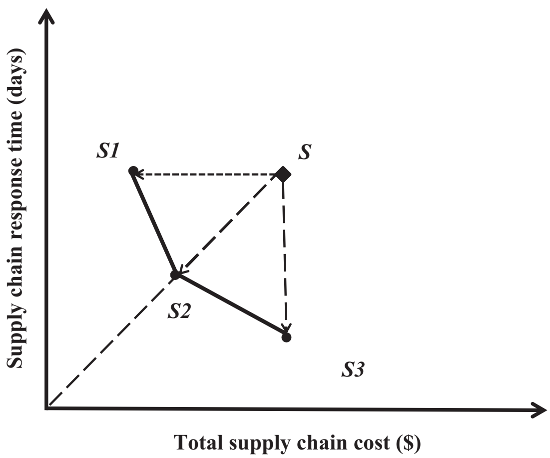


Fig. 6.1 Projections to frontier

	A	B	C	D	E	F	G	H	I	J
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	Efficiency
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		0	1
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0	1
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	1
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0	1
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0.56	1
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	1
8	Ford Motor	243283	24547	346990		137137	4139		0	0.377606
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	0.578288
10	Exxon	91296	40436	82000		110009	6470		0	1
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	1
12	Wal-Mart	37871	14762	675000		93627	2740		0.44	1
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	0.484837
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0	1
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0	0.42684
16	AT&T	88884	17274	299300		79609	139		0	0.504427
17										
18		Reference		DMU under	15	Efficiency				
19	**Constraints**	set		Evaluation		0.504427				
20	Assets	44835.477	≤	44835.477						
21	Equity	10222.526	≤	17274						
22	Employees	299300	≤	299300				Measure-Specific		
23	Revenue	135142.15	≥	79609						
24	Profit	1319.0622	≥	139						
25	Σλ	1	=	1						

Fig. 6.2 Input-oriented VRS measure-specific spreadsheet model

6.2 Measure-Specific Models in Spreadsheets

Since the measure-specific models are closely related to the envelopment models, the spreadsheet models can be modified from the envelopment spreadsheet models.

Figure 6.2 shows an input-oriented VRS measure-specific spreadsheet model where the Assets input is of interest. We only need to change the formulas in cells D21:D22 (representing Equity and Employee for the DMU under evaluation) in the input-oriented VRS envelopment spreadsheet model, as shown in Fig. 2.9 to

Cell D21 =INDEX(C2:C16,E18,1)
Cell D22 =INDEX(D2:D16,E18,1)

The Solver parameters remain the same, as shown in Fig. 2.15. All the VBA procedures developed for the envelopment models can be used. In Fig. 6.2, the VBA procedure "DEA_1" is assigned to the button "Measure-Specific".

If we apply the same formula changes in the Second-stage Slack Spreadsheet Model shown in Fig. 2.23, with the same Solver parameters shown in Fig. 2.24 and with the macro "DEASlack", we can optimize the slacks for the spreadsheet model shown in Fig. 6.2 after we obtain the efficiency scores. Figure 6.3 shows the results (see Excel file measure-specific.xlsm).

	E	F	G	H	I	J	K	L	M	N	O	P
1		Revenue	Profit		λ	Efficiency		Assets	Equity	Employees	Revenue	Profit
2		184365.2	346.2		0	1		0	0	0	0	0
3		181518.7	314.8		0	1		0	0	0	0	0
4		169164.6	121.2		0	1		0	0	0	0	0
5		168828.6	6880.7		0	1		0	0	0	0	0
6		167530.7	210.5		0.7705	1		0	0	0	0	0
7		161057.4	156.6		0	1		0	0	0	0	0
8		137137	4139		0	0.737556		75728.26	0	220277.5	0	0
9		111052	2662.4		0	0.603245		0	30247.6	58265.4	29763.39	0
10		110009	6470		0	1		0	0	0	0	0
11		109833.7	6904.6		0	1		0	0	0	0	0
12		93627	2740		0.2295	1		0	0	0	0	0
13		84167.1	1468.8		0	0.557596		0	17865.99	146812.7	58813.12	0
14		83206.7	2426.6		0	1		0	0	0	0	0
15		81937.2	2209.1		0	0.470611		0	25465.33	122500.6	60995.79	0
16		79609	139		0	0.533544		0	8738.346	139610.4	70960.21	652.0406
17												
18	15											
19		**Slack**										
20		0										
21		8738.346										
22		139610.4		Slack								
23		70960.21										
24		652.0406										
25		219961										

Fig. 6.3 Second-stage slacks for input-oriented VRS measure-specific model

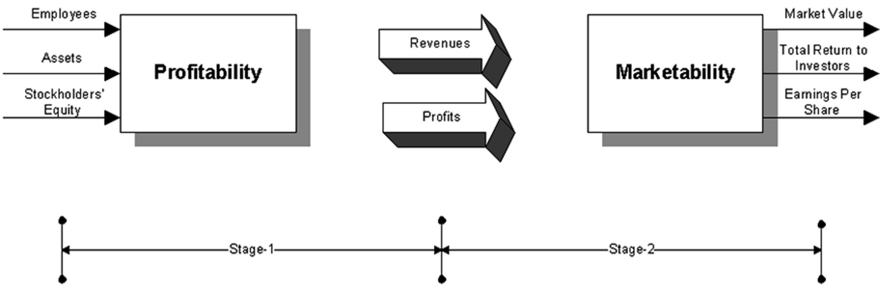


Fig. 6.4 Input-output system for Fortune 500 companies

6.3 Performance Evaluation of Fortune 500 Companies

Fortune magazine analyzes the financial performance of companies by eight measures: revenue, profit, assets, number of employees (employees), stockholders' equity (equity), market value (MV), earnings per share (EPS) and total return to investors (TRI).

In order to obtain an overall performance index, Zhu (2000) employs DEA to reconcile these eight measures via a two-stage transformation process described in Fig. 6.4. Each stage is defined by a group of "inputs (x)" and "outputs (y)".

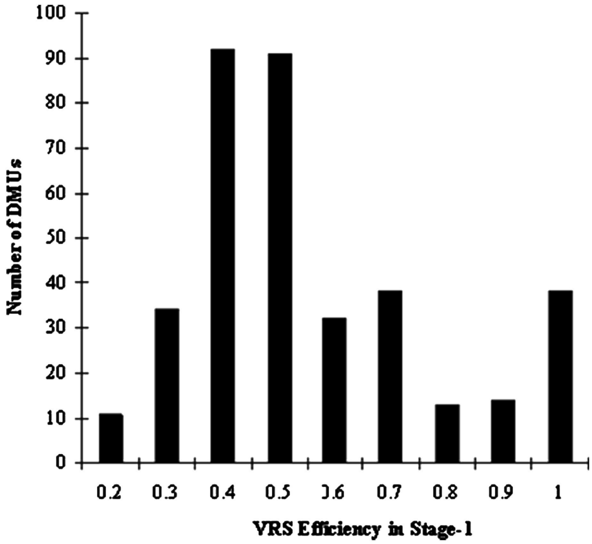


Fig. 6.5 Profitability VRS efficiency distribution

The performance in the first stage (stage-1) may be viewed as profitability, i.e., a company's ability to generate the revenue and profit in terms of its current labor, assets and capital stock. The performance in the second stage (stage-2) may be viewed as (stock) marketability, i.e., a company's performance in stock market by its revenue and profit generated.

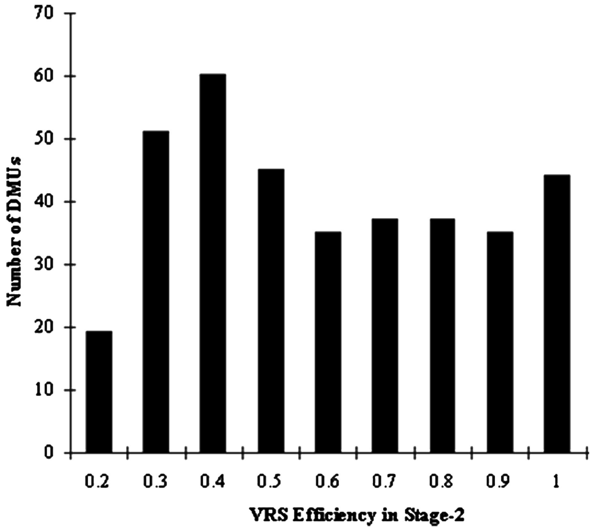
The data of 1995 is used. The DMU numbers correspond to the ranks by the magnitude of revenues. Because some data on MV, profit and equity are not available for some companies, we exclude these companies, and analyze the performance of the 364 companies.

6.3.1 Identification of Best Practice Frontier

Because the Fortune 500 list consists of a variety of companies representing different industries, we assume that the best-practice frontier exhibits VRS. We use the input-oriented VRS envelopment model to identify the best-practice.

Figures 6.5 and 6.6 show the distributions of VRS efficiency scores. 30 and 16 DMUs are VRS-efficient in profitability (stage-1) and marketability (stage-2), respectively. In stage-1, most VRS scores are distributed over [0.27, 0.51]. In stage-2, the VRS scores are almost evenly distributed over [0.16, 1]. Only four companies, namely, General Electric (DMU7), Coca-Cola (DMU48), Nash Finch (DMU437), and CompUSA (DMU451) are on the best-practice frontiers of stage-1 and stage-2.

Fig. 6.6 Marketability VRS efficiency distribution



6.3.2 *Measure-Specific Performance*[1]

Proportional reductions of all inputs are used to determine the best practice frontier for the Fortune 500 companies. However, in an evaluation of inefficient DMUs, non-proportional input (output) improvement may be more appropriate. Therefore, we seek an alternative way to further characterize the performance of inefficient companies by measure-specific models.

Because we have already obtained the VRS best-practice frontier and the measure-specific models yield the same frontier, we modify the VRS measure-specific models for a particular inefficient DMU_d,

$$\begin{aligned}
&\theta_d^{k*} = \min \theta_d^k \quad d \in \mathbf{N} \\
&\text{subject to} \\
&\sum_{j \in \mathbf{E}} \lambda_j^d x_{kj} = \theta_d^k x_{kd} \qquad k \in \{1,\ldots,m\} \\
&\sum_{j \in \mathbf{E}} \lambda_j^d x_{ij} \le x_{id} \qquad i \ne k \\
&\sum_{j \in \mathbf{E}} \lambda_j^d y_{rj} \ge y_{rd} \qquad r = 1,\ldots,s \\
&\sum_{j \in \mathbf{E}} \lambda_j^d = 1 \\
&\lambda_j^d \ge 0, \ j \in \mathbf{E}.
\end{aligned} \tag{6.1}$$

[1] The material in this section is adapted from European Journal of Operational Research, Vol 123, Zhu, J., Multi-factor Performance Measure Model with An Application to Fortune 500 Companies, 105–124, 2000, with permission from Elsevier Science.

$$
\begin{aligned}
&\phi_d^{q*} = \max \varphi_d^q \quad d \in \mathbf{N} \\
&\text{subject to} \\
&\sum_{j \in \mathbf{E}} \lambda_j^d y_{qj} = \phi_d^q y_{qd} \qquad q \in \{1, \ldots, s\} \\
&\sum_{j \in \mathbf{E}} \lambda_j^d y_{rj} \geq y_{rd} \qquad r \neq q \\
&\sum_{j \in \mathbf{E}} \lambda_j^d x_{ij} \leq x_{id} \qquad i = 1, \ldots, m \\
&\sum_{j \in \mathbf{E}} \lambda_j^d = 1 \\
&\lambda_j^d \geq 0, \;\; j \in \mathbf{E}.
\end{aligned} \tag{6.2}
$$

where **E** and **N** represent the index sets for the efficient and inefficient companies, respectively, identified by the VRS envelopment DEA model.

Models (6.1) and (6.2) determine the maximum potential decrease of an input and increase of an output while keeping other inputs and outputs at current levels.

Tables 6.2 and 6.3 report the results for the top-20 companies. Recall that revenue and profit are two factors served as the two outputs in stage-1 and the two inputs in stage-2. Therefore, we have two measure-specific efficiency scores for each revenue and each profit.

We may use the average measure-specific efficiency scores (optimal values to (6.1) or (6.2)) within each industry to characterize the measure-specific industry efficiency. However, different companies with different sizes may exist in each industry. Therefore arithmetic averages may not be a good way to characterize the industry efficiency. Usually, one expects large input and output levels, e.g., assets and revenue, form relatively big companies.

Thus, we define weighted measure-specific scores within each industry by considering the sizes of the companies.

(size-adjusted) kth input-specific industry efficiency measure for industry **F**

$$
I_k^{\mathbf{F}} = \sum_{d \in \mathbf{F}} \theta_d^{k*} \cdot \frac{x_{kd}}{\sum_{d \in \mathbf{F}} x_{kd}} = \frac{\sum_{d \in \mathbf{F}} \hat{x}_{kd}}{\sum_{d \in \mathbf{F}} x_{kd}} \tag{6.3}
$$

(size-adjusted) qth output-specific industry efficiency measure for industry **F**

$$
O_q^{\mathbf{F}} = \sum_{d \in \mathbf{F}} \phi_d^{q*} \cdot \frac{y_{qd}}{\sum_{d \in \mathbf{F}} y_{qd}} = \frac{\sum_{d \in \mathbf{F}} \hat{y}_{qd}}{\sum_{d \in \mathbf{F}} y_{qd}} \tag{6.4}
$$

Table 6.2 Profitability measure-specific efficiency

DMU	Company	Profitability				
No.	Name	employees	assets	equity	revenue	profit
1	General Motors	1.000	1.000	1.000	1.000	1.000
2	Ford Motors	1.000	1.000	1.000	1.000	1.000
3	Exxon	1.000	1.000	1.000	1.000	1.000
4	Wal-Mart Stores	1.000	1.000	1.000	1.000	1.000
5	AT&T	0.479	0.503	0.653	1.172	41.670
6	IBM	0.304	0.598	0.573	1.307	1.397
7	General Electric	1.000	1.000	1.000	1.000	1.000
8	Mobil	1.000	1.000	1.000	1.000	1.000
9	Chrysler	0.805	0.735	0.906	1.060	1.381
10	Philip Morris	1.000	1.000	1.000	1.000	1.000
13	Du Pont De Nemours	0.933	0.950	0.976	1.015	1.039
14	Texaco	0.933	0.862	0.936	1.046	2.475
15	Sears Roebuck	1.000	1.000	1.000	1.000	1.000
17	Procter & Gamble	0.325	0.743	0.654	1.291	1.413
18	Chevron	0.493	0.469	0.444	1.716	4.100
19	Citicorp	0.285	0.096	0.385	2.237	1.415
20	Hewlett-Packard	0.286	0.772	0.535	1.287	1.443
Average		0.755	0.808	0.827	1.184	3.784

Table 6.3 Marketability measure-specific efficiency

DMU	Company	Marketability				
No.	Name	Revenue	Profit	MV	TRI	EPS
1	General Motors	0.025	0.010	3.207	9.258	84.743
2	Ford Motors	0.028	0.013	3.314	33.754	170.670
3	Exxon	0.155	0.088	1.284	3.022	41.071
4	Wal-Mart Stores	0.058	0.029	2.011	43.768	447.912
5	AT&T	1.000	1.000	1.000	1.000	1.000
6	IBM	0.114	0.032	1.690	7.235	59.740
7	General Electric	1.000	1.000	1.000	1.000	1.000
8	Mobil	0.068	0.052	2.401	6.494	103.697
9	Chrysler	0.057	0.020	4.388	17.559	115.283
10	Philip Morris	0.239	0.095	1.408	2.465	47.335
13	Du Pont De Nemours	0.122	0.023	2.342	8.748	107.288
14	Texaco	0.080	0.059	3.202	7.863	243.004
15	Sears Roebuck	0.079	0.020	4.737	9.926	135.778
17	Procter & Gamble	0.168	0.127	1.789	5.644	138.595
18	Chevron	0.122	0.059	2.035	11.386	427.273
19	Citicorp	0.118	0.017	3.000	4.154	82.197
20	Hewlett-Packard	0.172	0.111	1.839	3.268	111.866
Average		0.212	0.162	2.391	10.385	136.379

where $\hat{x}_{kd}\left(=\theta_d^{k*}x_{kd}\right)$ and $\hat{y}_{qd}\left(=\phi_d^{q*}y_{qd}\right)$ are, respectively, the projected (potentially efficient) levels for kth input and qth output of DMU_d, $d\in$ **F**.

The weights in (6.3) $\left(\dfrac{x_{kd}}{\sum_{d\in\mathbf{F}} x_{kd}}, d\in\mathbf{F}\right)$ and (6.4) $\left(\dfrac{y_{qd}}{\sum_{d\in\mathbf{F}} y_{qd}}, d\in\mathbf{F}\right)$ are normalized, therefore a specific industry F achieves 100% efficiency, i.e., $I_k^{\mathbf{F}}=1$, and $O_q^{\mathbf{F}}=1$ if and only if, all of its companies are located on the best-practice frontier.

Tables 6.4 and 6.5 report the industry efficiency scores for the 30 selected industries where the number in parenthesis represents the corresponding arithmetic mean of measure-specific efficiency scores.

A relatively large discrepancy between weighted and arithmetic average scores is detected for six industries—General Merchandiser, Health Care, Motor Vehicles & Parts, Petroleum Refining, Pipelines, and Telecommunications. Since (6.3) and (6.4) determine the industry efficiency by considering the size of each company, this may imply that efficiency may highly correlate with size in these industries.

6.3.3 Benchmark Share

Non-zero λ_j^* indicates that DMU$_j$ is used as a benchmark. As an efficient company, the role it plays in evaluating inefficiency companies is to be of interest. One wants to know the importance of each efficient DMU in measuring the inefficiencies of inefficient DMUs. Based upon the non-zero λ_j^*, we develop benchmark-share measures for each efficient company via (6.1) and (6.2).

We define the kth input-specific benchmark-share for each efficient DMU_j, $j\in\mathbf{E}$,

$$\Delta_j^k=\frac{\sum_{d\in\mathbf{N}}\lambda_j^{d*}(1-\theta_d^{k*})x_{kd}}{\sum_{d\in\mathbf{N}}(1-\theta_d^{k*})x_{kd}} \tag{6.5}$$

where λ_j^{d*} and θ_d^{k*} are optimal values in (6.1).

We define the qth output-specific benchmark-share for each efficient DMU_j, $j\in\mathbf{E}$,

$$\Pi_j^q=\frac{\sum_{d\in\mathbf{N}}\lambda_j^{d*}(\phi_d^{q*}-1)y_{qd}}{\sum_{d\in\mathbf{N}}(\phi_d^{q*}-1)y_{qd}} \tag{6.6}$$

where λ_j^{d*} and ϕ_d^{q*} are optimal values in (6.2).

The benchmark-share Δ_j^k (or Π_j^q) depends on the values of λ_j^{d*} and θ_d^{k*} (or λ_j^{d*} and ϕ_d^{q*}). Note that $\left(1-\theta_d^{k*}\right)x_{kd}$ and $\left(\phi_d^{q*}-1\right)y_{qd}$ characterize the potential decrease on kth input and increase on qth output, respectively.

Table 6.4 Profitability measure-specific industry efficiency

Industries	Employees	Assets	Equity	Revenue	Profit
Aerospace	0.12 (0.11)	0.30 (0.35)	0.24 (0.23)	1.98 (2.26)	3.85 (4.16)
Airlines	0.12 (0.12)	0.24 (0.29)	0.24 (0.27)	1.95 (2.37)	4.54 (5.36)
Beverages	0.34 (0.33)	0.56 (0.46)	0.56 (0.39)	1.71 (2.17)	1.72 (4.96)
Chemicals	0.46 (0.32)	0.56 (0.46)	0.46 (0.31)	1.91 (2.54)	1.83 (2.46)
Commercial banks	0.13 (0.13)	0.06 (0.05)	0.31 (0.24)	3.66 (4.26)	2.62 (3.24)
Computer and data services	0.20 (0.30)	0.57 (0.54)	0.36 (0.31)	2.93 (3.12)	1.69 (2.39)
Computers, office equipment	0.25 (0.26)	0.60 (0.59)	0.48 (0.38)	1.51 (2.00)	1.77 (3.33)
Diversified financials	0.15 (0.38)	0.62 (0.39)	0.39 (0.43)	2.46 (3.09)	2.08 (2.40)
Electric & gas utilities	0.30 (0.32)	0.18 (0.18)	0.16 (0.15)	3.73 (3.85)	2.69 (4.03)
Electronics, electrical equipment	0.41 (0.30)	0.86 (0.53)	0.69 (0.38)	1.68 (2.35)	1.57 (3.12)
Entertainment	0.12 (0.16)	0.20 (0.24)	0.15 (0.25)	3.24 (3.06)	4.33 (7.86)
Food	0.29 (0.35)	0.45 (0.55)	0.32 (0.43)	1.84 (2.01)	2.49 (6.26)
Food & drug stores	0.35 (0.23)	0.70 (0.64)	0.44 (0.41)	1.49 (1.83)	2.39 (3.34)
Forest & paper products	0.16 (0.17)	0.30 (0.36)	0.21 (0.18)	2.47 (2.83)	3.44 (7.88)
General merchandisers	0.65 (0.32)	0.85 (0.65)	0.65 (0.43)	1.35 (2.16)	1.95 (3.52)
Health care	0.07 (0.30)	0.34 (0.47)	0.23 (0.32)	2.69 (2.75)	3.61 (4.94)
Industrial & farm equipment	0.14 (0.13)	0.33 (0.36)	0.24 (0.18)	2.43 (2.89)	2.78 (3.59)
Insurance: life & health (stock)	0.15 (0.25)	0.06 (0.07)	0.19 (0.22)	2.57 (2.97)	4.72 (5.23)
Insurance: property & causality (stock)	0.29 (0.37)	0.29 (0.27)	0.47 (0.35)	2.26 (2.76)	1.84 (2.64)
Metal products	0.12 (0.11)	0.43 (0.42)	0.19 (0.17)	3.04 (3.28)	2.82 (4.83)
Motor vehicles & parts	0.77 (0.32)	0.92 (0.51)	0.84 (0.39)	1.19 (2.11)	1.36 (2.67)

Table 6.4 (continued)

Industries	Employees	Assets	Equity	Revenue	Profit
Petroleum refining	0.64 (0.51)	0.71 (0.50)	0.73 (0.44)	1.29 (1.72)	1.85 (6.28)
Pharmaceuticals	0.38 (0.41)	0.63 (0.64)	0.52 (0.54)	2.02 (2.14)	1.44 (1.59)
Pipelines	0.63 (0.57)	0.51 (0.44)	0.59 (0.50)	1.76 (1.91)	1.40 (2.64)
Publishing, printing	0.13 (0.20)	0.39 (0.43)	0.16 (0.21)	3.35 (3.44)	2.77 (2.84)
Soaps, cosmetics	0.40 (0.47)	0.64 (0.64)	0.58 (0.58)	1.40 (1.45)	1.67 (3.00)
Special retailers	0.24 (0.27)	0.69 (0.66)	0.35 (0.40)	1.72 (2.11)	2.39 (4.28)
Telecommunications	0.36 (0.25)	0.41 (0.34)	0.42 (0.33)	1.70 (2.45)	3.72 (10.19)
Temporary help	0.50 (0.56)	0.84 (0.87)	0.68 (0.68)	1.30 (1.31)	1.39 (1.31)
Wholesalers	0.54 (0.58)	0.69 (0.74)	0.38 (0.55)	1.34 (1.46)	2.37 (2.08)

*The number in parenthesis represents the arithmetic average

Table 6.5 Marketability measure-specific industry efficiency

Industries	Revenue	Profit	MV	TRI	EPS
Aerospace	0.21 (0.29)	0.06 (0.06)	3.44 (4.51)	5.88 (6.39)	102.60 (131.24)
Airlines	0.35 (0.55)	0.34 (0.38)	4.73 (4.10)	2.66 (5.80)	35.65 (54.23)
Beverages	0.64 (0.61)	0.80 (0.37)	1.37 (2.67)	4.57 (4.94)	102.55 (95.84)
Chemicals	0.32 (0.52)	0.06 (0.11)	3.55 (4.16)	9.63 (13.37)	84.31 (100.13)
Commercial banks	0.34 (0.51)	0.07 (0.13)	3.94 (4.05)	6.40 (5.98)	85.39 (104.16)
Computer and data services	0.82 (0.85)	0.83 (0.66)	1.20 (1.52)	4.01 (5.68)	38.83 (38.77)
Computers, office equipment	0.24 (0.50)	0.09 (0.02)	2.37 (3.90)	4.28 (8.59)	84.83 (86.03)
Diversified financials	0.33 (0.47)	0.13 (0.17)	2.82 (3.79)	5.24 (7.31)	6.88 (117.12)
Electric & gas utilities	0.56 (0.65)	0.08 (0.14)	4.99 (5.00)	10.34 (11.68)	128.75 (126.03)
Electronics, electrical equipment	0.63 (0.54)	0.47 (0.19)	2.14 (3.54)	6.34 (14.22)	86.56 (87.05)
Entertainment	0.40 (0.50)	0.41 (0.39)	1.49 (1.56)	6.96 (10.07)	248.91 (285.42)
Food	0.30 (0.42)	0.09 (0.20)	3.42 (3.28)	8.83 (16.35)	145.47 (143.31)
Food & drug stores	0.28 (0.44)	0.10 (0.21)	5.52 (5.31)	6.24 (25.35)	83.66 (86.30)
Forest & paper products	0.49 (0.59)	0.06 (0.16)	4.24 (5.14)	11.63 (26.57)	67.17 (66.03)
General merchandisers	0.12 (0.32)	0.04 (0.08)	3.48 (4.33)	17.41 (24.60)	129.62 (152.9)
Health care	0.50 (0.66)	0.16 (0.28)	2.68 (2.91)	8.95 (13.56)	98.43 (95.67)
Industrial & farm equipment	0.39 (0.52)	0.08 (0.15)	4.19 (4.20)	7.56 (13.39)	98.35 (103.26)
Insurance: life & health (stock)	0.37 (0.53)	0.10 (0.14)	4.86 (5.16)	6.25 (6.58)	57.71 (65.72)
Insurance: property & causality (stock)	0.28 (0.47)	0.05 (0.08)	4.09 (5.55)	6.42 (8.49)	71.85 (80.33)
Metal products	0.63 (0.70)	0.30 (0.36)	2.22 (2.12)	7.47 (8.89)	103.54 (96.76)
Motor vehicles & parts	0.07 (0.32)	0.02 (0.08)	41.19 (5.72)	13.26 (30.38)	77.53 (77.95)
Petroleum refining	0.17 (0.36)	0.07 (0.16)	2.41 (4.21)	9.94 (18.53)	102.81 (122.06)
Pharmaceuticals	0.44 (0.44)	0.33 (0.29)	1.75 (2.00)	4.77 (5.65)	163.58 (193.30)
Pipelines	0.51 (0.65)	0.07 (0.13)	4.38 (4.22)	5.22 (6.47)	46.10 (103.94)
Publishing, printing	0.67 (0.73)	0.21 (0.22)	3.23 (3.19)	14.35 (19.16)	69.09 (73.99)
Soaps, cosmetics	0.25 (0.37)	0.13 (0.12)	1.89 (2.23)	10.04 (11.98)	101.42 (109.16)
Special retailers	0.40 (0.60)	0.15 (0.30)	3.19 (3.90)	6.79 (20.26)	103.34 (98.36)
Telecommunications	0.68 (0.44)	0.12 (0.25)	1.85 (2.53)	7.19 (61.16)	188.16 (214.68)
Temporary help	0.69 (0.78)	0.32 (0.36)	3.68 (3.52)	18.87 (93.62)	30.83 (30.48)
Wholesalers	0.37 (0.47)	0.18 (0.30)	3.95 (4.61)	6.64 (8.83)	49.55 (49.33)

*The number in parenthesis represents the arithmetic average

Δ_j^k and Π_j^q are weighted λ_j^* across all inefficient DMUs. The weights, $\dfrac{(1-\theta_d^{k*})x_{kd}}{\sum_{d\in\mathbf{N}}(1-\theta_d^{k*})x_{kd}}$ in (6.5) and $\dfrac{(\phi_d^{q*}-1)y_{qd}}{\sum_{d\in\mathbf{N}}(\phi_d^{q*}-1)y_{qd}}$ in (6.6) are normalized. Therefore, we have $\sum_{j\in\mathbf{E}}\Delta_j^k=1$ and $\sum_{j\in\mathbf{E}}\Pi_j^q=1$. (Note that $\sum_{j\in\mathbf{E}}\lambda_j^{d*}=1$ in (6.1) and (6.2).)

It is very clear form (6.5) and (6.6) that an efficient company which does not act as a referent DMU for any inefficient DMU will have zero benchmark-share. The bigger the benchmark-share, the more important an efficient company is in benchmarking.

Table 6.6 reports the benchmark-shares for 12 selected VRS-efficient companies. The benchmark-shares for the remaining VRS-efficient companies are less than 0.01 %. Of the total 60 benchmark-shares, 12 are greater than 10 %. Particularly, DMU48 (Coca-Cola), DMU156 (General Mills) and DMU281 (Bindley Western) have the biggest benchmark-share with respect to employees, equity and profit, respectively. This means that, e.g., General Mills plays a leading role in setting a benchmark with respect to equity input given the current levels of employees and assets. Note that General Mills had the highest returns on equity in 1995.

In Table 6.7, DMU226 (Continental Airlines) and DMU292 (Berkshire Hathaway) are two important companies in TRI and EPS benchmarking, respectively. (Note that Continental Airlines and Berkshire Hathaway had the highest TRI and EPS in 1995.) Although Berkshire Hathaway was ranked 18 in terms of MV levels by the Fortune magazine, the benchmark-share of 39.99 % indicates that it had an outstanding performance in terms of MV given other measures at their current levels. This indicates that single financial performance alone is not sufficient to characterize a company's performance.

Finally, note that, e.g., DMU292 and DMU474 both acted as a referent DMU in 63 % of the inefficient DMUs when measuring the revenue-specific efficiency. However, the benchmark-share indicates that DMU474 is more important.

We here explore the multidimensional financial performance of the Fortune 500 companies. Revenue-top-ranked companies do not necessarily have top-ranked performance in terms of profitability and (stock) marketability. Most companies exhibited serious inefficiencies. The measure-specific models enable us to study the performance based upon a specific measure while keeping the current levels of other measures. See Zhu (2000) for more discussion on measuring the performance of Fortune 500 companies.

Table 6.6 Benchmark-share for profitability

DMU No.	Company name	Employees	Assets	Equity	Revenue	Profit
8	Mobil	3.07 %	1.51 %	0.76 %	16.00 %	0.15 %
32	Fed. Natl. Mortgage	2.78 %	0	2.76 %	0.89 %	0.10 %
44	Loews	7.17 %	0.14 %	0	0.95 %	1.41 %
48	Coca-Cola	2.58 %	12.54 %	10.65 %	2.88 %	40.65 %
94	IBP	0	22.51 %	0.07 %	13.16 %	0.80 %
153	Bergen Brunswig	0.60 %	0	0.16 %	5.91 %	0.17 %
156	General Mills	1.86 %	0.01 %	60.91 %	17.19 %	7.85 %
168	Cardinal health	3.12 %	2.82 %	0.01 %	10.89 %	0
281	Bindley Western	52.91 %	4.79 %	2.93 %	5.97 %	2.86 %
419	Micron Technology	0.17 %	28.37 %	0.24 %	0.29 %	11.04 %
437	Nash Finch	0	10.16 %	0.02 %	0.24 %	0.27 %
447	Williams	8.68 %	0	0	0.02 %	8.62 %
Total		82.94 %	82.85 %	78.51 %	74.39 %	73.92 %

Table 6.7 Benchmark-share for marketability

DMU No.	Company Name	Revenue	Profit	MV	TRI	EPS
5	AT&T	0	12.33 %	6.95 %	2.22 %	0
7	IBM	0	0.20 %	3.83 %	6.39 %	0.79 %
48	Coca-Cola	5.44 %	0.80 %	11.37 %	0.13 %	0.11 %
78	Kimberly-Clark	0.04 %	36.66 %	6.96 %	0	0.10 %
210	Burlington Northern Santa FE	0.05 %	4.29 %	6.39 %	0	0
219	Microsoft	8.46 %	0	9.97 %	0	0
226	Continental Airlines	0.44 %	0.69 %	1.30 %	81.91 %	0.87 %
292	Berkshire Hathaway	23.56 %	8.37 %	39.99 %	0.18 %	73.96 %
312	Chiquita Brands International	0.00 %	15.49 %	0.07 %	0.17 %	11.41 %
376	Consolidated Natural Gas	0.99 %	11.29 %	4.89 %	0.05 %	0.00 %
417	Oracle	0.00 %	0.00 %	1.37 %	0	0
437	Nash Finch	0.09 %	0.51 %	0	0	3.56 %
451	CompUSA	1.43 %	7.22 %	0.88 %	8.85 %	4.11 %
474	Computer Associates	29.90 %	0.04 %	1.69 %	0.00 %	0.00 %
494	Foundation Health	5.21 %	2.07 %	4.25 %	0.07 %	2.58 %
495	State Street Boston Corp.	24.39 %	0.04 %	0.09 %	0.03 %	2.51 %
Total		100 %	100 %	100 %	100 %	100 %

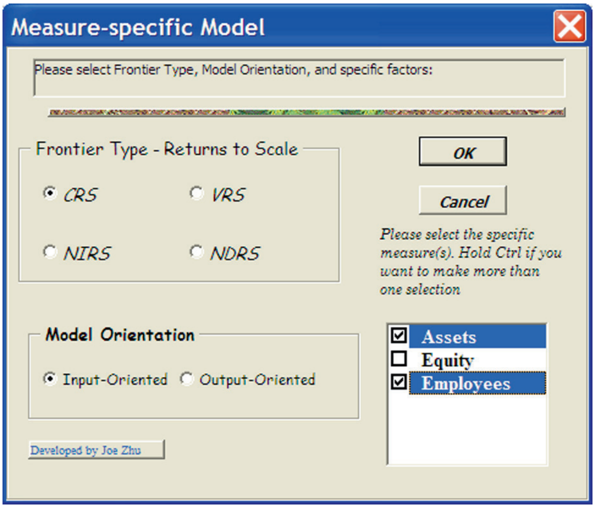


Fig. 6.7 Measure-specific models

6.4 Solving Measure-Specific Models Using DEAFrontier Software

To run the measure-specific models, select the "Measure Specific Model" menu item. You will be prompted with a form for selecting the data sheet and a form for selecting models presented in Table 6.1, as shown in Fig. 6.7.

Select the measures that are of interest. If you select all the input or all the output measures, then you have the envelopment models.

The results are reported in the "Efficiency", "Slack" and "Target" sheets.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_6) contains supplementary material, which is available to authorized users.

References

Banker, R. D., & Morey, R. C. (1986). Efficiency analysis for exogenously fixed inputs and outputs. *Operations Research, 34,* 513–521.

Zhu, J. (2000). Multi-factor performance measure model with an application to Fortune 500 companies. *European Journal of Operational Research, 123*(1), 105–124.

Chapter 7
Non-radial DEA Models and DEA with Preference

7.1 Non-radial DEA Models

We can call the envelopment DEA models as radial efficiency measures, because these models optimize all inputs or outputs of a DMU at a certain proportion. Färe and Lovell (1978) introduce a non-radial measure which allows non-proportional reductions in positive inputs or augmentations in positive outputs. Table 7.1 summarizes the non-radial DEA models with respect to the model orientation and frontier type.

The slacks in the non-radial DEA models are optimized in a second-stage model where θ_i^* or ϕ_r^* are fixed. For example, under CRS we have

Input Slacks for Output-oriented Non-radial DEA Model

$$
\begin{aligned}
&\max \sum_{r=1}^{s} s_r^+ \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} = \theta_i^* x_{io} && i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} && r = 1,2,\ldots,s; \\
&\lambda_j \geq 0 && j = 1,2,\ldots,n.
\end{aligned}
$$

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*,
International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_7,

Table 7.1 Non-radial DEA Models

Frontier type	Input-Oriented	Output-Oriented
CRS	$\begin{aligned} &\min\left(\frac{1}{m}\sum_{i=1}^{m}\theta_i - \varepsilon\sum_{r=1}^{s}s_r^+\right) \\ &\text{subject to} \\ &\sum_{j=1}^{n}\lambda_j x_{ij} = \theta_i x_{io} \quad i=1,2,\ldots,m; \\ &\sum_{j=1}^{n}\lambda_j y_{rj} - s_r^+ = y_{ro} \quad r=1,2,\ldots,s; \\ &\theta_i \le 1 \quad i=1,2,\ldots,m; \\ &\lambda_j \ge 0 \quad j=1,2,\ldots,n. \end{aligned}$	$\begin{aligned} &\max\left(\frac{1}{s}\sum_{r=1}^{s}\phi_r + \varepsilon\sum_{r=1}^{s}s_r^+\right) \\ &\text{subject to} \\ &\sum_{j=1}^{n}\lambda_j x_{ij} + s_i^- = x_{io} \quad i=1,2,\ldots,m; \\ &\sum_{j=1}^{n}\lambda_j y_{rj} = \phi_r y_{ro} \quad r=1,2,\ldots,s; \\ &\phi_i \ge 1 \quad r=1,2,\ldots,s; \\ &\lambda_j \ge 0 \quad j=1,2,\ldots,n. \end{aligned}$
VRS	Add $\sum_{j=1}^{n}\lambda_j = 1$	
NIRS	Add $\sum_{j=1}^{n}\lambda_j \le 1$	
NDRS	Add $\sum_{j=1}^{n}\lambda_j \ge 1$	
Efficient target	$\begin{cases} \hat{x}_{io} = \theta_i^* x_{io} & i=1,2,\ldots,m \\ \hat{y}_{ro} = y_{ro} + s_r^{+*} & r=1,2,\ldots,s \end{cases}$	$\begin{cases} \hat{x}_{io} = x_{io} - s_i^{-*} & i=1,2,\ldots,m \\ \hat{y}_{ro} = \phi_r^* y_{ro} & r=1,2,\ldots,s \end{cases}$

Output Slacks for Input-oriented Non-radial DEA Model

$$
\begin{aligned}
&\max \sum_{i=1}^{m} s_i^- \\
&\text{subject to} \\
&\sum_{j=1}^{n}\lambda_j x_{ij} + s_i^- = x_{io} \qquad i=1,2,\ldots,m; \\
&\sum_{j=1}^{n}\lambda_j y_{rj} = \phi_r^* y_{ro} \qquad r=1,2,\ldots,s; \\
&\lambda_j \ge 0 \qquad j=1,2,\ldots,n.
\end{aligned}
$$

Note that input slacks do not exist in the input-oriented non-radial DEA models, and output slacks do not exist in the output-oriented non-radial DEA models.

Because $\theta_i^* \le 1$ ($\phi_r^* \ge 1$), $\frac{1}{m}\sum_{i=1}^{m}\theta_i^* \le 1$ and $\frac{1}{m}\sum_{i=1}^{m}\theta_i^* = 1$ if and only if $\theta_i^* = 1$ for all i ($\frac{1}{s}\sum_{r=1}^{s}\phi_r^* \ge 1$ and $\frac{1}{s}\sum_{r=1}^{s}\phi_r^* = 1$ if and only if $\phi_r^* = 1$ for all r). Thus, $\frac{1}{m}\sum_{i=1}^{m}\theta_i^*$ ($\frac{1}{s}\sum_{r=1}^{s}\phi_r^*$) can be used as an efficiency index.

Both the envelopment models and the non-radial DEA models yield the same frontier, but may yield different efficient targets (even when the envelopment models do not have non-zero slacks). For example, if we change the second input from

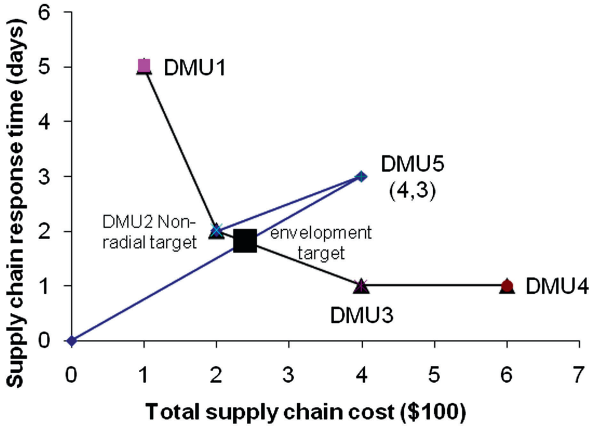


Fig. 7.1 Efficient targets

4 to 3 for DMU5 in Table 2.1 (Chap. 2), the input-oriented CRS envelopment model yields the efficient target of $x1 = 2.4$ and $x2 = 1.8$ (with $\lambda_2^* = 0.8$, $\lambda_3^* = 0.2$, and all zero slacks). Whereas the input-oriented CRS non-radial DEA model yields DMU2 as the efficient target for DMU5 (see Fig. 7.1). Note that both models yield the same target of DMU3 for DMU4.

7.2 DEA with Preference Structure and Cost/Revenue Efficiency

Both the envelopment models and the non-radial DEA models yield efficient targets for inefficient DMUs. However, these targets may not be preferred by the management or achievable under the current management and other external conditions. Therefore, some other targets along the efficient frontier should be considered as preferred ones. This can be done by constructing preference structures over the proportions by which the corresponding current input levels (output levels) can be changed. Zhu (1996) develops a set of weighted non-radial DEA models where various efficient targets along with the frontier can be obtained.

Let A_i $(i = 1, 2, \ldots, m)$ and B_r $(r = 1, 2, \ldots, s)$ be user-specified preference weights which reflect the relative degree of desirability of the adjustments of the current input and output levels, respectively. Then we can have a set of weighted non-radial DEA models based upon Table 7.1 by changing the objective functions $\frac{1}{m}\sum_{i=1}^{m}\theta_i$ and $\frac{1}{s}\sum_{r=1}^{s}\phi_r$ to $\sum_{i=1}^{m}A_i\theta_i / \sum_{i=1}^{m}A_i$ and $\sum_{r=1}^{s}B_r\phi_r / \sum_{r=1}^{s}B_r$, respectively.

Further, if we remove the constraint $\theta_i \le 1$ $(\phi_r \ge 1)$, we obtain the DEA/preference structure (DEA/PS) models shown in Table 7.2 (Zhu 1996).

If some $A_i = 0$ $(B_r = 0)$, then set the corresponding $\theta_i = 1$ $(\phi_r = 1)$. But at least one of such weights should be positive. Note that for example, the bigger the weight

Table 7.2 DEA/Preference structure models

Frontier type	Input-Oriented	Output-Oriented
CRS	$\min\left(\dfrac{\sum_{i=1}^{m} A_i\theta_i}{\sum_{i=1}^{m} A_i} - \varepsilon\sum_{r=1}^{s} s_r^+\right)$ subject to $\sum_{j=1}^{n}\lambda_j x_{ij} = \theta_i x_{io}\quad i=1,2,...,m;$ $\sum_{j=1}^{n}\lambda_j y_{rj} - s_r^+ = y_{ro}\quad r=1,2,...,s;$ $\lambda_j \ge 0\quad j=1,2,...,n.$	$\max\left(\dfrac{\sum_{r=1}^{s} B_r\phi_r}{\sum_{r=1}^{s} B_r} + \varepsilon\sum_{r=1}^{s} s_r^+\right)$ subject to $\sum_{j=1}^{n}\lambda_j x_{ij} + s_i^- = x_{io}\quad i=1,2,...,m;$ $\sum_{j=1}^{n}\lambda_j y_{rj} = \phi_r y_{ro}\quad r=1,2,...,s;$ $\lambda_j \ge 0\quad j=1,2,...,n.$
VRS	Add $\sum_{j=1}^{n}\lambda_j = 1$	
NIRS	Add $\sum_{j=1}^{n}\lambda_j \le 1$	
NDRS	Add $\sum_{j=1}^{n}\lambda_j \ge 1$	
Efficient target	$\begin{cases}\hat{x}_{io} = \theta_i^* x_{io} & i=1,2,...,m\\ \hat{y}_{ro} = y_{ro} + s_r^{+*} & r=1,2,...,s\end{cases}$	$\begin{cases}\hat{x}_{io} = x_{io} - s_i^{-*} & i=1,2,...,m\\ \hat{y}_{ro} = \phi_r^* y_{ro} & r=1,2,...,s\end{cases}$

A_i, the higher the priority DMU_o is allowed to adjust its ith input amount to a lower level. i.e., when inefficiency occurs, the more one wants to adjust an input or an output, the bigger the weight should be attached to θ_i or ϕ_r. If we can rank the inputs or outputs according to their relative importance, then we can obtain a set of ordinal weights. One may use Delphi-like techniques, or Analytic Hierarchy Process (AHP) to obtain the weights. However, caution should be paid when we convert the ordinal weights into preference weights. For example, if an input (output) is relatively more important and the DMU does not wish to adjust it with a higher rate, we should take the reciprocal of the corresponding ordinal weight as the preference weight. Otherwise, if the DMU does want to adjust the input (output) with a higher rate, we can take the ordinal weight as the preference weight. Also, one may use the principal component analysis to derive the information on weights (Zhu 1998).

Note that in the DEA/PS models, some θ_i^* (ϕ_r^*) may be greater (less) than one under certain weight combinations. i.e., the DEA/PS models are *not* restricted to the case where 100 % efficiency is maintained through the input decreases or output increases.

Now, in order to further investigate the property of DEA/PS models, we consider the dual program to the input-oriented CRS DEA/PS model.

$$\begin{aligned}
&\max \sum_{r=1}^{s} u_r y_{ro} \\
&\text{subject to} \\
&\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{m} \nu_i x_{ij} \le 0 \quad j = 1, \ldots, n; \\
&\nu_i x_{io} = A_i / \sum_{i=1}^{m} A_i \qquad i = 1, \ldots, m; \\
&\mu_r, \nu_i \ge 0
\end{aligned} \tag{7.1}$$

We see that the normalization condition $\sum_{i=1}^{m} \nu_i x_{io} = 1$ is also satisfied in (7.1). The DEA/PS model is actually a DEA model with fixed input multipliers.

Let p_i^o denote the *i*th input price for DMU_o and $\tilde{x}_{io}$ represents the *i*th input that minimizes the cost. Consider the following DEA model for calculating the "minimum cost".

$$\begin{aligned}
&\min \sum_{i=1}^{m} p_i^o \tilde{x}_{io} \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le \tilde{x}_{io} \qquad i = 1, \ldots, m \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} \qquad r = 1, \ldots, s \\
&\lambda_j, \tilde{x}_{io} \ge 0
\end{aligned} \tag{7.2}$$

The dual program to (7.2) is

$$\begin{aligned}
&\max \sum_{r=1}^{s} u_r y_{ro} \\
&\text{subject to} \\
&\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{m} \nu_i x_{ij} \le 0 \quad j = 1, \ldots, n; \\
&0 \le \nu_i \le p_i^o \qquad i = 1, \ldots, m; \\
&\mu_r, \nu_i \ge 0
\end{aligned} \tag{7.3}$$

By the complementary slackness condition of linear programming, we have that if $\tilde{x}_{io}^* > 0$ then $p_i^o = \nu_i^*$. Thus, ν_i^* can be interpreted as p_i^o. Consequently, the input prices can be used to develop the preference weights.

In the DEA literature, we have a concept called "cost efficiency" which is defined as

$$\frac{\sum_{i=1}^{m} p_i^o \tilde{x}_{io}^*}{\sum_{i=1}^{m} p_i^o x_{io}}$$

The following development shows that the related DEA/PS model can be used to obtain exact the cost efficiency scores. Because the actual cost —$\sum_{i=1}^{m} p_i^o x_{io}$ is a constant for a specific DMU_o, cost efficiency can be directly calculated by the following modified (7.2).

$$\begin{aligned}
&\min \frac{\sum_{i=1}^{m} p_i^o \tilde{x}_{io}}{\sum_{i=1}^{m} p_i^o x_{io}} \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le \tilde{x}_{io} \qquad i = 1, 2, \ldots, m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} \qquad r = 1, 2, \ldots, s; \\
&\lambda_j \ge 0 \qquad\qquad j = 1, 2, \ldots, n.
\end{aligned} \tag{7.4}$$

Let $\tilde{x}_{io} = \theta_i x_{io}$. Then (7.4) is equivalent to the input-oriented CRS DEA/PS model with $A_i = p_i^o x_{io}$. This indicates that if one imposes a proper set of preference weights for each DMU under consideration, then the DEA/PS model yields cost efficiency measure. (see Seiford and Zhu (2002) for an empirical investigation of DEA efficiency and cost efficiency.)

Similarly, the output-oriented DEA/PS model can be used to obtain the "revenue efficiency" which is defined as

$$\frac{\sum_{r=1}^{s} q_r^o \tilde{y}_{ro}^*}{\sum_{r=1}^{s} q_r^o y_{ro}}$$

where q_r^o indicates output price for DMU_o and $\tilde{y}_{ro}$ represents the rth output that maximizes the revenue in the following linear programming problem.

$$
\begin{aligned}
&\max \sum_{r=1}^{s} q_r^o \tilde{y}_{ro} \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io} \qquad i = 1, 2, \ldots, m \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge \tilde{y}_{ro} \qquad r = 1, 2, \ldots, s \\
&\lambda_j, \tilde{y}_{ro} \ge 0
\end{aligned} \tag{7.5}
$$

Let $\tilde{y}_{ro} = \phi_r y_{ro}$ and $B_r = q_r^o y_{ro}$ in the output-oriented DEA/PS model. We have

$$
\begin{aligned}
&\max \frac{\sum_{r=1}^{s} q_r^o \tilde{y}_{ro}}{\sum_{r=1}^{s} q_r^o y_{ro}} \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io} \qquad i = 1, 2, \ldots, m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge \tilde{y}_{ro} \qquad r = 1, 2, \ldots, s; \\
&\lambda_j \ge 0 \qquad j = 1, 2, \ldots, n.
\end{aligned}
$$

which calculates the revenue efficiency.

7.3 DEA/Preference Structure Models in Spreadsheets

Figure 7.2 shows an input-oriented VRS DEA/PS spreadsheet model. Cells I2:I16 are reserved for λ_j. Cells F20:F22 are reserved for θ_i. These are the changing cells in the Solver parameters shown in Fig. 7.3.

The target cell is cell F19 which contains the following formula
Cell F19 =SUMPRODUCT(F20:F22,G20:G22)/SUM(G20:G22)
where cells G20:G22 are reserved for the input weights.

The formulas for cells B20:B25 are
Cell B20 =SUMPRODUCT(B2:B16,$I$2:$I$16)
Cell B21 =SUMPRODUCT(C2:C16,$I$2:$I$16)
Cell B22 =SUMPRODUCT(D2:D16,$I$2:$I$16)
Cell B23 =SUMPRODUCT(F2:F16,$I$2:$I$16)

	A	B	C	D	E	F	G	H	I
1	Company	Assets	Equity	Employees		Revenue	Profit		λ
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		1
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0
8	Ford Motor	243283	24547	346990		137137	4139		0
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0
10	Exxon	91296	40436	82000		110009	6470		0
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0
12	Wal-Mart	37871	14762	675000		93627	2740		0
13	Hitachi	91620.9	[illegible]	[illegible]		[illegible]	[illegible]		0
14	Nippon Life Insurance	364762.5	[illegible]	[illegible]		[illegible]	[illegible]		0
15	Nippon Telegraph & Telephone	127077.3	[illegible]	[illegible]		[illegible]	[illegible]		0
16	AT&T	88884	17274	299300		79609	139		0
17									
18		Reference		DMU under	1	Efficiency			
19	**Constraints**	set		Evaluation		3	Weights		
20	Assets	91920.6	=	91920.6		1	1		
21	Equity	10950	=	10950		1	1		
22	Employees	36000	=	36000		1	1		
23	Revenue	184365.2	≥	184365.2					
24	Profit	346.2	≥	346.2					
25	Σλ	1							

Reserved to indicate the DMU under evaluation.
Efficiency; Weighted sum of individual θi
Changing cells; Individual θi

Fig. 7.2 Input-oriented VRS DEA/PS spreadsheet model

Cell B24 =SUMPRODUCT(G2:G16,$I$2:$I$16)
Cell B25 =SUM(I2:I16)

The formulas for cells D20:D24 are

Cell D20 =F20*INDEX(B2:B16,E18,1)
Cell D21 =F21*INDEX(C2:C16,E18,1)
Cell D22 =F22*INDEX(D2:D16,E18,1)
Cell D23 =INDEX(F2:F16,E18,1)
Cell D24 =INDEX(G2:G16,E18,1)

Figure 7.4 shows the results and the VBA procedure "DEAPS" which automates the calculation.

Note that the θ_i ($i=1, 2, 3$) are not restricted in Fig. 7.3. If we add $\theta_i \le 1$ ($F$20:FF22 <=1), then we obtain the results shown in Fig. 7.5.

7.4 DEA and Multiple Objective Linear Programming

Charnes et al. (1985) describe the relationship between DEA frontier and Pareto-Koopmans efficient empirical production frontier. This work points out the relation of efficiency in DEA and pareto optimality in multiple criteria decision making

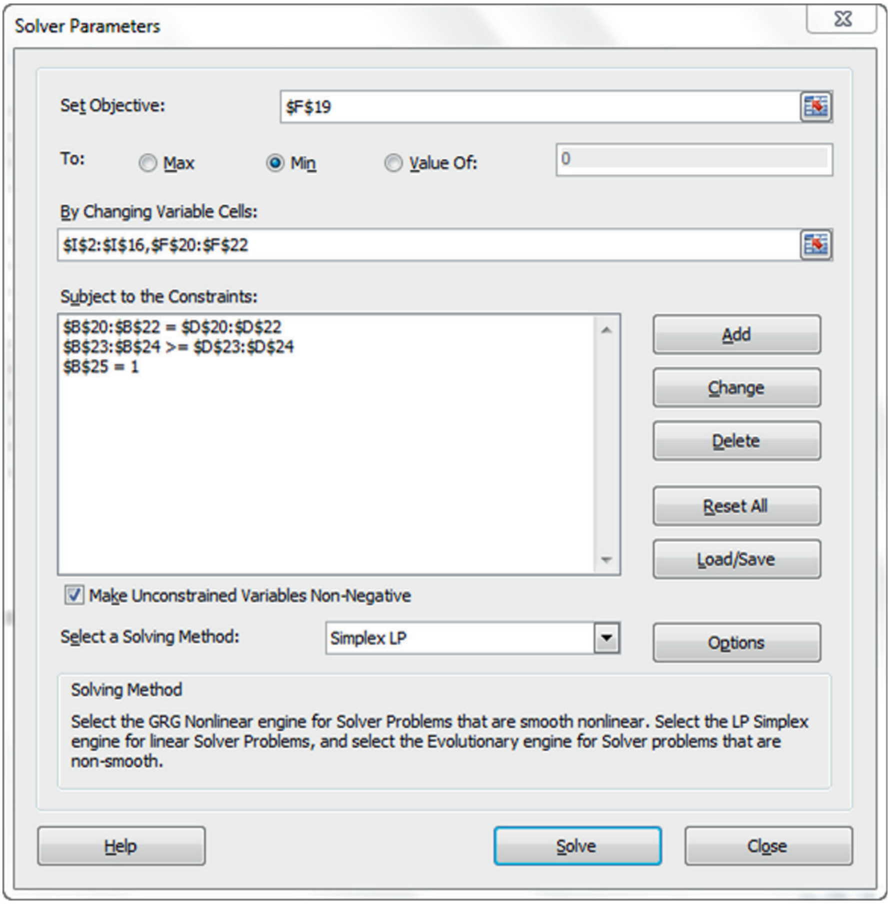


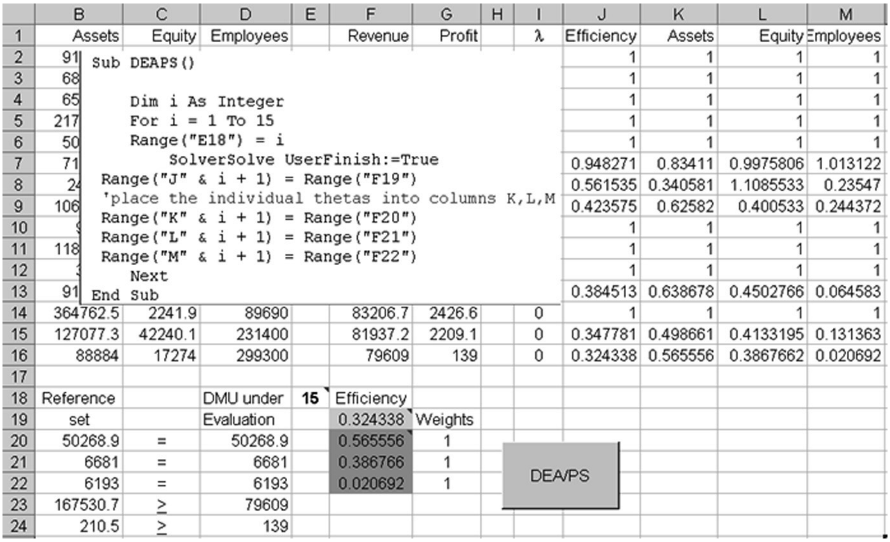
Fig. 7.3 Solver parameters for Input-oriented VRS DEA/PS model

(MCDM) or Multiple Objective Linear Programming (MOLP). The relationship between DEA and MOLP is again raised by Belton and Vickers (1993), Doyle and Green (1993) and Stewart (1994) in their discussion of DEA and MCDM. Joro et al. (1998) provide a structure comparison of DEA and MOLP.

In fact, as shown in Chen (2005), the DEA/PS models have a strong relationship with MOLP. To demonstrate this, we use vector presentation of $\mathbf{x}_j = (x_{1j}, \ldots, x_{mj})$ and $\mathbf{y}_j = (y_{1j}, \ldots, y_{sj})$.

7.4.1 Output-oriented DEA

Consider the following MOLP model

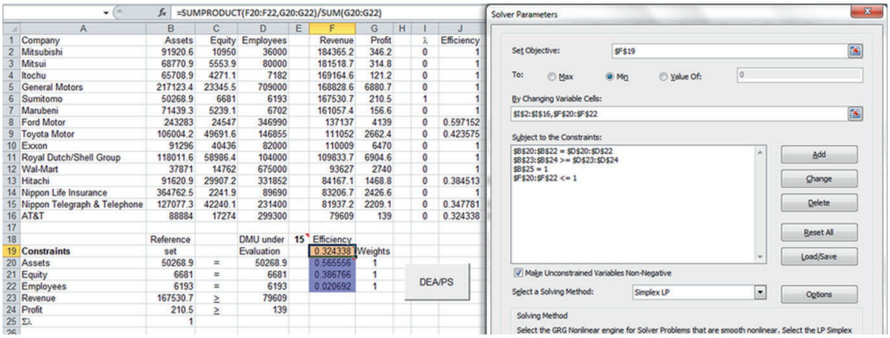


	B	C	D	E	F	G	H	I	J	K	L	M
1	Assets	Equity	Employees		Revenue	Profit		λ	Efficiency	Assets	Equity	Employees
2	91								1	1	1	1
3	68								1	1	1	1
4	65								1	1	1	1
5	217								1	1	1	1
6	50								1	1	1	1
7	71								0.948271	0.83411	0.9975806	1.013122
8	24								0.561535	0.340581	1.1085533	0.23547
9	106								0.423575	0.62582	0.400533	0.244372
10	9								1	1	1	1
11	118								1	1	1	1
12	3								1	1	1	1
13	91								0.384513	0.638678	0.4502766	0.064583
14	364762.5	2241.9	89690		83206.7	2426.6		0	1	1	1	1
15	127077.3	42240.1	231400		81937.2	2209.1		0	0.347781	0.498661	0.4133195	0.131363
16	88884	17274	299300		79609	139		0	0.324338	0.565556	0.3867662	0.020692
17												
18	Reference		DMU under	15	Efficiency							
19	set		Evaluation		0.324338	Weights						
20	50268.9	=	50268.9		0.565556	1						
21	6681	=	6681		0.386766	1						
22	6193	=	6193		0.020692	1						
23	167530.7	≥	79609									
24	210.5	≥	139									

```
Sub DEAPS()

    Dim i As Integer
    For i = 1 To 15
    Range("E18") = i
        SolverSolve UserFinish:=True
 Range("J" & i + 1) = Range("F19")
 'place the individual thetas into columns K,L,M
 Range("K" & i + 1) = Range("F20")
 Range("L" & i + 1) = Range("F21")
 Range("M" & i + 1) = Range("F22")
    Next
End Sub
```


Fig. 7.4 Efficiency result for Input-oriented VRS DEA/PS model



=SUMPRODUCT(F20:F22,G20:G22)/SUM(G20:G22)

	A	B	C	D	E	F	G	H	I	J
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	Efficiency
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		0	1
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0	1
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	1
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0	1
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		1	1
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	1
8	Ford Motor	243283	24547	346990		137137	4139		0	0.597152
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	0.423575
10	Exxon	91296	40436	82000		110009	6470		0	1
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	1
12	Wal-Mart	37871	14762	675000		93627	2740		0	1
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	0.384513
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0	1
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0	0.347781
16	AT&T	88884	17274	299300		79609	139		0	0.324338
17										
18		Reference		DMU under	15	Efficiency				
19	Constraints	set		Evaluation		0.324338	Weights			
20	Assets	50268.9	=	50268.9		0.565556	1			
21	Equity	6681	=	6681		0.386766	1			
22	Employees	6193	=	6193		0.020692	1			
23	Revenue	167530.7	≥	79609						
24	Profit	210.5	≥	139						
25	Σλ	1								

Fig. 7.5 Efficiency result for Input-oriented VRS non-radial DEA model

$$
\begin{aligned}
&\max_{\lambda_j}\left(\sum_{j=1}^{n}\lambda_j \mathbf{y}_j\right)=\left(\sum_{j=1}^{n}\lambda_j y_{1j},\ldots,\sum_{j=1}^{n}\lambda_j y_{sj}\right)\\
&\text{subject to}\\
&\sum_{j=1}^{n}\lambda_j x_{ij}\le x_{io} \qquad i=1,\ldots,m;\\
&\lambda_j\ge 0, \qquad\qquad j=1,\ldots,n.
\end{aligned}
\tag{7.6}
$$

where $\mathbf{x}_o=(x_{1o},\ldots,x_{mo})$ represents the input vector of DMU_o among others.

If all DMUs produce only one output, i.e., $\mathbf{y}_j$ is a scalar rather than a vector, then (7.6) is a single objective linear programming problem

$$\begin{aligned}&\max_{\lambda_j}\left(\sum_{j=1}^{n}\lambda_j y_j\right)\\&\text{subject to}\\&\sum_{j=1}^{n}\lambda_j x_{ij}\le x_{io}\qquad i=1,\ldots,m;\\&\lambda_j\ge 0,\qquad\qquad j=1,\ldots,n.\end{aligned}\tag{7.7}$$

Let $\lambda_j y_j = \lambda_j'$, then (7.7) turns into

$$\begin{aligned}&\max_{\lambda_j'}\sum_{j=1}^{n}\lambda_j'\\&\text{subject to}\\&\sum_{j=1}^{n}\lambda_j' x_{ij}'\le x_{io}'\qquad i=1,\ldots,m;\\&\lambda_j'\ge 0,\qquad\qquad j=1,\ldots,n.\end{aligned}\tag{7.8}$$

where $x_{ij}' = x_{ij} / y_j$ and $x_{io}' = x_{io} / y_o$.

As shown in Charnes et al. (1978), model (7.8) is equivalent to the output-oriented CRS envelopment model

$$\begin{aligned}&\max_{\lambda_j, z_o} z_o\\&\text{subject to}\\&\sum_{j=1}^{n}\lambda_j y_j\ge z_o y_o\\&\sum_{j=1}^{n}\lambda_j x_{ij}\le x_{io}\qquad i=1,\ldots,m;\\&\lambda_j\ge 0,\qquad\qquad j=1,\ldots,n.\end{aligned}$$

Next, if $\mathbf{y}_j$ is a vector with s components, then we define

$$\sum_{j=1}^{n}\lambda_j y_{rj} = \sigma_r y_{ro}\tag{7.9}$$

As a result, (7.6) becomes

$$\begin{aligned}
&\max_{\lambda_j \sigma_r} (\sigma_1 y_{1o}, \ldots, \sigma_s y_{so}) \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j y_{rj} = \sigma_r y_{ro} \qquad r = 1, \ldots, s; \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io}, \qquad i = 1, \ldots, m; \\
&\lambda_j \ge 0, \qquad j = 1, \ldots, n.
\end{aligned} \tag{7.10}$$

Let $\mathbf{W} = \{ w \mid w \in \mathbf{R}^s, w_r \ge 0 \text{ and } \sum_{r=1}^{s} w_r = 1 \}$ be the set of nonnegative weights. The weighting problem associated with (7.10) is defined for some $w \in \mathbf{W}$ as

$$\begin{aligned}
&\max_{\lambda_j \sigma_r} \sum_{r=1}^{s} w_r \sigma_r y_{ro} \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j y_{rj} = \sigma_r y_{ro} \qquad r = 1, \ldots, s; \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io} \qquad i = 1, \ldots, m; \\
&\lambda_j \ge 0 \qquad j = 1, \ldots, n.
\end{aligned} \tag{7.11}$$

Furthermore, let $\overline{w}_r = w_r y_{ro}$ for all $r = 1, \ldots, s$, then (7.11) is equivalent to the following linear programming problem

$$\begin{aligned}
&\max_{\lambda_j \sigma_r} \sum_{r=1}^{s} \overline{w}_r \sigma_r \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j y_{rj} = \sigma_r y_{ro} \qquad r = 1, \ldots, s; \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io} \qquad i = 1, \ldots, m; \\
&\lambda_j \ge 0 \qquad j = 1, \ldots, n.
\end{aligned} \tag{7.12}$$

Model (7.12) is exactly the output-oriented CRS DEA/preference model. However, if we wish output level cannot be decreased to reach the efficient frontier, we specify (7.13) instead of (7.9).

$$\sum_{j=1}^{n} \lambda_j y_{rj} = \sigma_r y_{ro} \quad \text{such that} \quad \sigma_r \geq 1 \quad \text{for all} \quad r = 1, \ldots, s. \tag{7.13}$$

We see that for a specific DMU_o, $\lambda_o^* = 1$ and $\lambda_j^* = 0$ $(j \neq o)$ is an optimal solution to (7.12), when $\sigma_r^* = 1$ for all $r = 1, \ldots, s$. Note that if some $\sigma_r^* \neq 1$, then $\lambda_o^* = 0$ is an optimal solution to (7.12). Therefore, (7.6) can be interpreted as follows: when $\mathbf{x}_o = (x_{1o}, \ldots, x_{mo})$ is regarded as resource, if the resource $\mathbf{x}_o$ can be used among other DMUs (associated with $\lambda_j^* \neq 0$), then more desirable or preferred output level $\mathbf{y}^*$ is produced and $\mathbf{y}_o$ is not a pareto solution to (7.6).

It can be seen that weighted non-radial DEA model (7.12) is equivalent to an MOLP problem. If we impose an additional on $\sum_{j=1}^{n} \lambda_j$ in (7.6), then we obtain other output-oriented DEA models.

7.4.2 *Input-oriented DEA*

Similar to (7.6), we write the following MOLP model.

$$\begin{aligned}
&\min_{\lambda_j} \left(\sum_{j=1}^{n} \lambda_j \mathbf{x}_j \right) = \left(\sum_{j=1}^{n} \lambda_j x_{1j}, \ldots, \sum_{j=1}^{n} \lambda_j x_{mj} \right) \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \leq y_{ro} \qquad r = 1, \ldots, s; \\
&\lambda_j \geq 0, \qquad j = 1, \ldots, n.
\end{aligned} \tag{7.14}$$

where $\mathbf{y}_o = (y_{1o}, \ldots, y_{so})$ represents the output vector of DMU_o. If all DMUs use only one input, i.e., $\mathbf{x}_j$ is a scalar, then (7.14) is a single objective linear programming problem and is equivalent to the input-oriented CRS envelopment model with single input.

Let $\mathbf{G} = \{ g \mid g \in \mathbf{R}^m, g_i \geq 0$ and $\sum_{i=1}^{m} g_i = 1 \}$ be the set of nonnegative weights. Then model (7.14) can be transformed into the following linear programming problem.

$$\begin{aligned}
&\min_{\lambda_j \tau_i} \sum_{i=1}^{m} \bar{g}_i \tau_i \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} = \tau_i x_{io} \qquad i = 1, \ldots, m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r = 1, \ldots, s; \\
&\lambda_j \geq 0 \qquad j = 1, \ldots, n.
\end{aligned} \tag{7.15}$$

where $\bar{g}_i = g_i x_{io}$ for all $i = 1, \ldots, m$, and τ_i is defined in (7.16) or (7.17).

$$\sum_{j=1}^{n} \lambda_j x_{ij} = \tau_i x_{io} \tag{7.16}$$

$$\sum_{j=1}^{n} \lambda_j x_{ij} = \tau_i x_{io} \text{ such that } \tau_i \leq 1 \text{ for all } i = 1, \ldots, m. \tag{7.17}$$

Model (7.15) is a weighted non-radial DEA model incorporated with preference over the adjustment of input levels. If we use (7.16), then there is no restrictions on τ_i and model (7.15) is the input-oriented CRS DEA/PS model.

Note that for a specific DMU_o, $\lambda_o^* = 1$ and $\lambda_j^* = 0$ $(j \neq o)$ is an optimal solution to (7.15), when $\tau_i^* = 1$ for all $i = 1, \ldots, m$. Note also that if some $\tau_i^* \neq 1$, then $\lambda_o^* = 0$ is an optimal solution to (7.15). If we impose an additional on $\sum_{j=1}^{n} \lambda_j$ in (7.15), then we obtain other input-oriented DEA models.

7.4.3 Non-Orientation DEA

Consider the following MOLP model.

$$\begin{aligned}
&\max_{\lambda_j} \left(\sum_{j=1}^{n} \lambda_j \mathbf{y}_j \right) = \left(\sum_{j=1}^{n} \lambda_j y_{1j}, \ldots, \sum_{j=1}^{n} \lambda_j y_{sj} \right) \\
&\min_{\lambda_j} \left(\sum_{j=1}^{n} \lambda_j \mathbf{x}_j \right) = \left(\sum_{j=1}^{n} \lambda_j x_{1j}, \ldots, \sum_{j=1}^{n} \lambda_j x_{mj} \right) \\
&\text{subject to} \\
&\lambda_j \geq 0 \qquad j = 1, \ldots, n.
\end{aligned} \tag{7.18}$$

We have the following equivalent linear programming model

$$\begin{aligned}
&\max_{\lambda_j, \sigma_r, \tau_i} \sum_{r=1}^{s} \bar{w}_r \sigma_r - \sum_{i=1}^{m} \bar{g}_i \tau_i \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} = \tau_i x_{io} \qquad i = 1, \ldots, m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} = \sigma_r y_{ro}, \qquad r = 1, \ldots, s; \\
&\tau_i \leq 1, \qquad i = 1, \ldots, m; \\
&\sigma_r \geq 1, \qquad r = 1, \ldots, s; \\
&\lambda_j \geq 0, \qquad j = 1, \ldots, n.
\end{aligned} \tag{7.19}$$

Note that $\sigma_r \geq 1$ and $\tau_i \leq 1$ in (7.19). Therefore, we have $\tau_i x_{io} = x_{io} - s_i^-$ and $\sigma_r y_{ro} = y_{ro} + s_r^+$, where $s_i^-, s_r^+ \geq 0$. Then, (7.19) becomes

$$
\begin{aligned}
&\max_{\lambda_j, s_i^-, s_r^+} \sum_{r=1}^{s} w_r s_r^+ + \sum_{i=1}^{m} g_i s_i^- - \sum_{i=1}^{m} \bar{g}_i + \sum_{r=1}^{s} \bar{w}_r \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \qquad i = 1,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \qquad r = 1,\ldots,s; \\
&s_i^-, s_r^+, \lambda_j \geq 0.
\end{aligned}
$$

which is a weighted slack-based DEA model (see Chap. 3 and Seiford and Zhu (1998)).

7.5 Using DEAFrontier Software

7.5.1 Non-radial Models

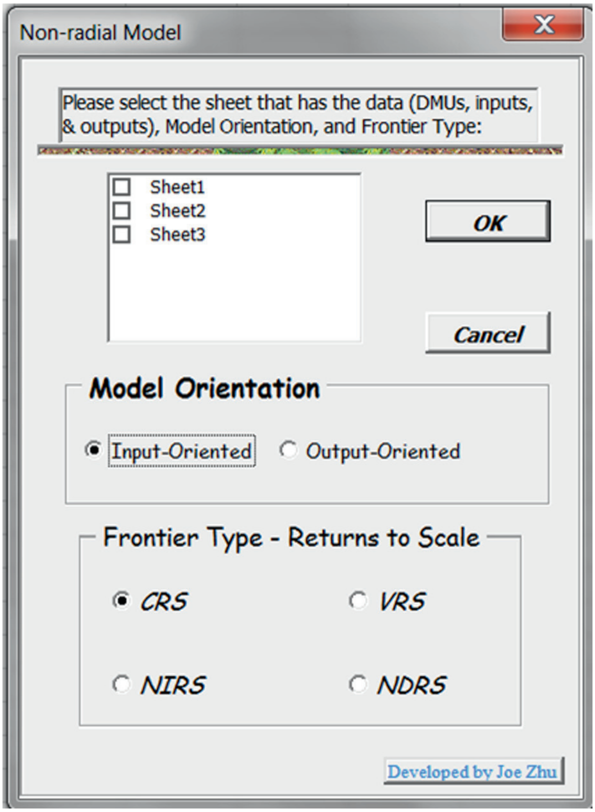
To run the non-radial models, select the "Non-radial Model" menu item. You will be prompted with a form as shown in Fig. 7.6 for selecting the models presented in Table 7.1. The Results are reported in "Efficiency", "Slack", and "Target" sheets.

7.5.2 Preference-Structure Models

To run the preference structure models, select the "Preference Structure Model" menu item. Figure 7.6 shows the form for specifying the models.

If "Yes" is selected under "Restrict Input/Output Change?", then we have weighted non-radial models. If "No" is selected, then we have the DEA/PS models presented in Table 7.2. The software will then ask you to specify the weights for the inputs or outputs, depending on the model orientation. The Results are reported in "Efficiency", "Slack", and "Target" sheets (Fig. 7.7).

Fig. 7.6 Non-radial models



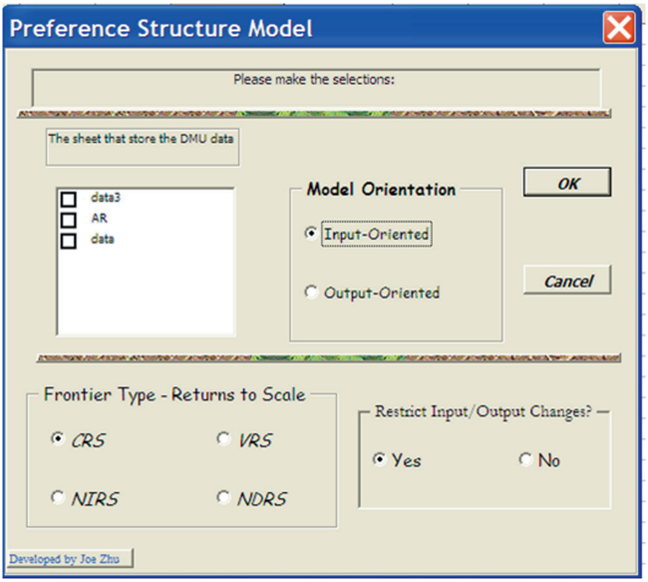


Fig. 7.7 Preference structure models

	A	B	C	D	E	F
1	Hospital	Doctor	Nurse		Outpat.	Inpat.
2	A	20	151		100	90
3	B	19	131		150	50
4	C	25	160		160	55
5	D	27	168		180	72
6	E	22	158		94	66
7	F	55	255		230	90
8	G	33	235		220	88
9	H	31	206		152	80
10	I	30	244		190	100
11	J	50	268		250	100
12	K	53	306		260	147
13	L	38	284		250	120

Fig. 7.8 Hospital data

Fig. 7.9 Input prices

	A	B	C	D	E
1	Hospital	Doctor	Nurse		
2	A	500	100		
3	B	350	80		
4	C	450	90		
5	D	600	120		
6	E	300	70		
7	F	450	80		
8	G	500	100		
9	H	450	85		
10	I	380	76		
11	J	410	75		
12	K	440	80		
13	L	400	70		

7.5.3 Cost Efficiency, Revenue Efficiency and Profit Efficiency

These models need information on the input and output prices. Consider the Hospital example in Cooper et al. (2000). The input and output data are reported in the "Data" sheet (Fig. 7.8), input price are reported in the "Input Price" sheet (Fig. 7.9) and the output price are reported in the "Output Price" sheet (Fig. 7.10).

The cost efficiency and revenue efficiency are discussed in Sect. 7.2. Table 7.3 summarizes the related models.

	A	B	C	D	E
1	Hospital	outpat.	Inpat.		
2	A	550	2010		
3	B	400	1800		
4	C	480	2200		
5	D	600	3500		
6	E	400	3050		
7	F	430	3900		
8	G	540	3300		
9	H	420	3500		
10	I	350	2900		
11	J	410	2600		
12	K	540	2450		
13	L	295	3000		

Fig. 7.10 Output price

Table 7.3 Cost efficiency and revenue efficiency models

Frontier type	Cost	Revenue
CRS	$\min \sum_{i=1}^{m} p_i^o \tilde{x}_{io}$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} \le \tilde{x}_{io} \quad i = 1,2,\ldots,m;$ $\sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} \quad r = 1,2,\ldots,s;$ $\lambda_j, \tilde{x}_{io} \ge 0$	$\max \sum_{r=1}^{s} q_r^o \tilde{y}_{ro}$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io} \quad i = 1,2,\ldots,m$ $\sum_{j=1}^{n} \lambda_j y_{rj} \ge \tilde{y}_{ro} \quad r = 1,2,\ldots,s$ $\lambda_j, \tilde{y}_{ro} \ge 0$
VRS	Add $\sum_{j=1}^{n} \lambda_j = 1$	
NIRS	Add $\sum_{j=1}^{n} \lambda_j \le 1$	
NDRS	Add $\sum_{j=1}^{n} \lambda_j \ge 1$	

In Table 7.3, p_i^o and q_r^o are unit price of the input i and unit price of the output r of DMU_o, respectively. These price data may vary from one DMU to another. The cost efficiency and revenue efficiency of DMU_o is defined as

Table 7.4 Profit efficiency models

Frontier type	
CRS	$\max \sum_{r=1}^{s} q_r^o \tilde{y}_{ro} - \sum_{i=1}^{m} p_i^o \tilde{x}_{io}$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} \le \tilde{x}_{io} \quad i = 1,2,\ldots,m$ $\sum_{j=1}^{n} \lambda_j y_{rj} \ge \tilde{y}_{ro} \quad r = 1,2,\ldots,s$ $\tilde{x}_{io} \le x_{io}, \tilde{y}_{ro} \ge y_{ro}$ $\lambda_j \ge 0$
VRS	Add $\sum_{j=1}^{n} \lambda_j = 1$
NIRS	Add $\sum_{j=1}^{n} \lambda_j \le 1$
NDRS	Add $\sum_{j=1}^{n} \lambda_j \ge 1$

$$\frac{\sum_{i=1}^{m} p_i^o \tilde{x}_{io}^*}{\sum_{i=1}^{m} p_i^o x_{io}} \quad \text{and} \quad \frac{\sum_{r=1}^{s} q_r^o y_{ro}}{\sum_{r=1}^{s} q_r^o \tilde{y}_{ro}^*}$$

Note that the revenue efficiency is defined as the reciprocal of the one defined in Sect. 7.2. As a result, the cost and revenue efficiency scores are within the range of 0 and 1.

The efficiency scores are reported in the "Cost Efficiency" ("Revenue Efficiency") sheet. The optimal inputs (outputs) are reported in the "OptimalData Cost Efficiency" ("OptimalData Revenue Efficiency") sheet.

Table 7.4 presents the models used to calculate the profit efficiency defined as

$$\frac{\sum_{r=1}^{s} q_r^o y_{ro} - \sum_{i=1}^{m} p_i^o x_{io}}{\sum_{r=1}^{s} q_r^o \tilde{y}_{ro}^* - \sum_{i=1}^{m} p_i^o \tilde{x}_{io}^*}$$

The results are reported in the "Profit Efficiency" and "OptimalData Profit Efficiency" sheets.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_7) contains supplementary material, which is available to authorized users.

References

Belton, V., & Vickers, S. P. (1993). Demystifying DEA—a visual interactive approach based on multiple criteria analysis. *Journal of Operational Research Society, 44,* 883–896.

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring efficiency of decision making units. *European Journal of Operational Research, 2,* 429–444.

Charnes, A., Cooper, W. W., Golany, B., Seiford, L. M., & Stutz, J. (1985). Foundations of data envelopment analysis for Pareto-Koopman's efficient empirical production functions. *Journal of Econometrics, 30,* 1–17.

Chen, Y. (2005). On preference structure in data envelopment analysis. *International Journal of Information Technology and Decision Making, 4*(3), 411–431.

Cooper, W. W., Seiford, L. M., & Tone, K. (2000). *Data envelopment analysis: A comprehensive text with models, applications, references and DEA-Solver software*. Boston: Kluwer Academic Publishers.

Doyle, J., & Green, R. (1993). Data envelopment analysis and multiple criteria decision making. *OMEGA, 21,* 713–715.

Färe, R., & Lovell, C. A. K. (1978). Measuring the technical efficiency. *Journal of Economic Theory, 19,* 150–162.

Joro, T., Korhonen, P., & Wallenius, J. (1998). Structural comparison of data envelopment analysis and multiple objective linear programming. *Management Science, 44,* 962–970.

Seiford, L. M., & Zhu, J. (1998). Identifying excesses and deficits in Chinese industrial productivity (1953–1990): A weighted data envelopment analysis approach. *OMEGA, 26*(2), 279–269.

Seiford, L. M., & Zhu, J. (2002). Value judgment versus allocative efficiency: A case of Tennessee county jails. *Studies in Regional Urban Planning,* (4), 89–98.

Stewart, T. J. (1994). Data envelopment analysis and multiple criteria decision making: A response. *Omega, 22*(2), 205–206.

Zhu, J. (1996). Data envelopment analysis with preference structure. *Journal of Operational Research Society, 47,* 136–150.

Zhu, J. (1998). Data envelopment analysis vs. principal component analysis: An illustrative study of economic performance of Chinese cities. *European Journal of Operational Research, 111*(1), 50–61.

Chapter 8
Modeling Undesirable Measures

8.1 Introduction

Both desirable (good) and undesirable (bad) outputs and inputs may be present. For example, the number of defective products is an undesirable output. One wants to reduce the number of defects to improve the performance. If inefficiency exists in production processes where final products are manufactured with a production of wastes and pollutants, the outputs of wastes and pollutants are undesirable and should be reduced to improve the performance.

Note that in the conventional DEA models, e.g., the VRS envelopment models, it is assumed that outputs should be increased and the inputs should be decreased to improve the performance or to reach the best-practice frontier. If one treats the undesirable outputs as inputs so that the bad outputs can be reduced, the resulting DEA model does not reflect the true production process.

Situations when some inputs need to be increased to improve the performance are also likely to occur. For example, in order to improve the performance of a waste treatment process, the amount of waste (undesirable input) to be treated should be increased rather than decreased as assumed in the conventional DEA models.

Seiford and Zhu (2002) develop an approach to treat undesirable input/outputs in the VRS envelopment models. The key to their approach is the use of DEA classification invariance under which classifications of efficiencies and inefficiencies are invariant to the data transformation.

8.2 Efficiency Invariance

Suppose that the inputs and outputs are transformed to $\bar{x}_{ij} = x_{ij} + u_i$ and $\bar{y}_{rj} = y_{rj} + v_r$, where u_i and v_r are nonnegative. Then the input-oriented and the output-oriented VRS envelopment models become

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213, DOI 10.1007/978-3-319-06647-9_8,

$$
\begin{aligned}
&\min \theta - \varepsilon\left(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+\right)\\
&\text{subject to}\\
&\sum_{j=1}^{n} \lambda_j \bar{x}_{ij} + s_i^- = \theta \bar{x}_{io} \qquad i = 1,2,\ldots,m;\\
&\sum_{j=1}^{n} \lambda_j \bar{y}_{rj} - s_r^+ = \bar{y}_{ro} \qquad r = 1,2,\ldots,s;\\
&\sum_{j=1}^{n} \lambda_j = 1\\
&\lambda_j \ge 0 \qquad j = 1,2,\ldots,n.
\end{aligned} \tag{8.1}
$$

$$
\begin{aligned}
&\max \phi + \varepsilon\left(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+\right)\\
&\text{subject to}\\
&\sum_{j=1}^{n} \lambda_j \bar{x}_{ij} + s_i^- = \bar{x}_{io} \qquad i = 1,2,\ldots,m;\\
&\sum_{j=1}^{n} \lambda_j \bar{y}_{rj} - s_r^+ = \phi \bar{y}_{ro} \qquad r = 1,2,\ldots,s;\\
&\sum_{j=1}^{n} \lambda_j = 1\\
&\lambda_j \ge 0 \qquad j = 1,2,\ldots,n.
\end{aligned} \tag{8.2}
$$

Ali and Seiford (1990) show that DMU_o is efficient under (2.4) or (2.5) (in Chap. 2) if and only if DMU_o is efficient under (8.1) or (8.2). This conclusion is due to the presence of the convexity constraint $\sum_{j=1}^{n} \lambda_j = 1$. This property also enables us to treat possible negative inputs and outputs before applying the VRS model (see Appendix of this chapter.)

In general, there are three cases of invariance under data transformation in DEA. The first case is restricted to the "classification invariance" where the classifications of efficiencies and inefficiencies are invariant to the data transformation. The second case is the "ordering invariance" of the inefficient DMUs. The last case is the "solution invariance" in which the new DEA model (after data translation) must be equivalent to the old one, i.e., both mathematical programming problems must have exactly the same solution. The method of Seiford and Zhu (2002) is concerned only with the first level of invariance—classification invariance. See Pastor (1996) and Lovell and Pastor (1995) for discussions in invariance property in DEA.

8.3 Undesirable Outputs

Let y_{rj}^g and y_{rj}^b denote the desirable (good) and undesirable (bad) outputs, respectively. Obviously, we wish to increase y_{rj}^g and to decrease y_{rj}^b to improve the performance. However, in the output-oriented VRS envelopment model, both y_{rj}^g and y_{rj}^b are supposed to increase to improve the performance. In order to increase the desirable outputs and to decrease the undesirable outputs, we proceed as follows.

First, we multiply each undesirable output by "–1" and then find a proper value v_r to let all negative undesirable outputs be positive. That is, $\bar{y}_{rj}^b = -y_{rj}^b + v_r > 0$. This can be achieved by $v_r = \max_j \left\{ y_{rj}^b \right\} + 1$, for example.

Based upon (8.2), we have

$$
\begin{aligned}
&\max h \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j y_{rj}^g \geq h y_{ro}^g \\
&\sum_{j=1}^{n} \lambda_j \bar{y}_{rj}^b \geq h \bar{y}_{ro}^b \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \leq x_{io} \\
&\sum_{j=1}^{n} \lambda_j \quad = 1 \\
&\lambda_j \geq 0, \ \ j = 1, \ldots, n
\end{aligned} \tag{8.3}
$$

Note that (8.3) increases desirable outputs and decreases undesirable outputs. The following theorem ensures that the optimized undesirable output of $y_{ro}^b (= v_r - h^* \bar{y}_{ro}^b)$ cannot be negative.

Theorem 8.1 Given a translation vector v, suppose h^* is the optimal value to (8.3), we have $h^* \bar{y}_{ro}^b \leq v_r$.

[Proof] Note that all outputs now are non-negative. Let λ_j^* be an optimal solution associated with h^*. Since $\sum_{j=1}^{n} \lambda_j^* = 1$, $h^* \bar{y}_{ro}^b \leq \bar{y}_r^*$, where $\bar{y}_r^*$ is composed from (translated) maximum values among all bad outputs. Note that $\bar{y}_r^* = -y_r^* + v_r$, where y_r^* is composed from (original) minimum values among all bad outputs. Thus, $h^* \bar{y}_o^b < v_r$.

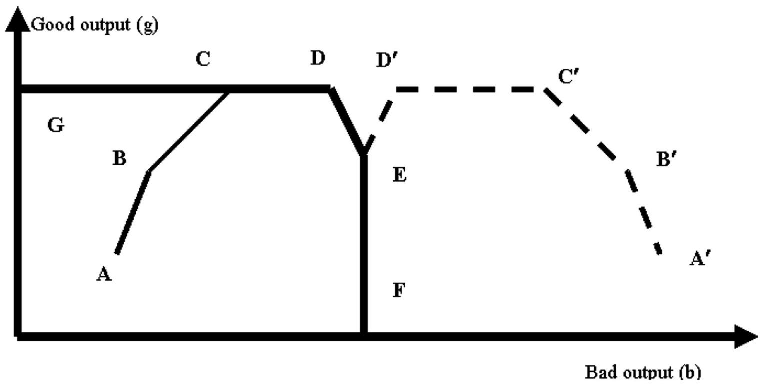


Fig. 8.1 Treatment of bad output

We may treat the undesirable outputs as inputs. However, this does not reflect the true production process. We may also apply a monotone decreasing transformation (e.g., $1/y_{rj}^b$) to the undesirable outputs and then to use the adapted variables as outputs. The current method, in fact, applies a linear monotone decreasing transformation. Since the use of linear transformation preserves the convexity, it is a good choice for a DEA model.

Figure 8.1 illustrates the method. The five DMUs A, B, C, D and E use an equal input to produce one desirable output (g) and one undesirable output (b). GCDEF is the (output) frontier. If we treat the undesirable output as an input, then ABCD becomes the VRS frontier. Model (8.2) rotates the output frontier at EF and obtains the symmetrical frontier. In this case, DMUs A′, B′ and C′, which are the adapted points of A, B and C, respectively, are efficient.

The efficient target for DMU_o is

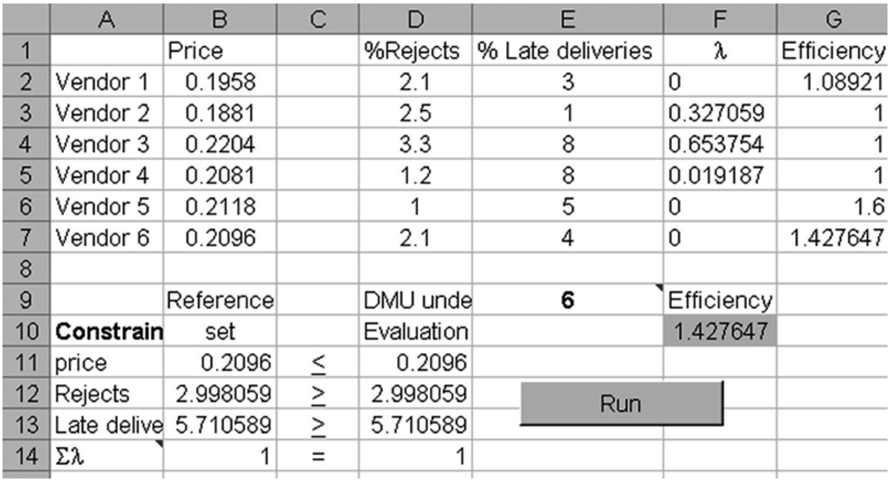
$$\begin{cases} \hat{x}_{io} = x_{io} - s_i^{-*} \\ \hat{y}_{ro}^g = h^* y_{ro}^g + s_r^{+*} \\ \hat{y}_{ro}^b = v_r - (h^* \bar{y}_{ro}^b + s_r^{+*}) \end{cases}$$

We conclude this section by applying the method to the six vendors studied in Weber and Desai (1996). Table 8.1 presents the data. The input is price per unit, and the outputs are percentage of late deliveries and percentage of rejected units. (See Weber and Desai (1996) for detailed discussion on the input and the two outputs.)

Obviously, the two outputs are bad outputs. We use an translation vector of (3.3 %, 8 %). (Or one could use (100 %, 100 %) as in Chap. 7.) Figure 8.2 shows the translated data and the spreadsheet model. This is actually a spreadsheet model for the output-oriented VRS envelopment model. Figure 8.3 shows the Solver parameters. Column G in Fig. 8.2 reports the efficiency scores. (see also file “badoutput.xlsm”.)

Table 8.1 Vendors. (Source: Weber and Desai 1996)

Vendors	Price ($/unit)	% Rejects	% Late deliveries
1	0.1958	1.2	5
2	0.1881	0.8	7
3	0.2204	0	0
4	0.2081	2.1	0
5	0.2118	2.3	3
6	0.2096	1.2	4



	A	B	C	D	E	F	G
1		Price		%Rejects	% Late deliveries	λ	Efficiency
2	Vendor 1	0.1958		2.1	3	0	1.08921
3	Vendor 2	0.1881		2.5	1	0.327059	1
4	Vendor 3	0.2204		3.3	8	0.653754	1
5	Vendor 4	0.2081		1.2	8	0.019187	1
6	Vendor 5	0.2118		1	5	0	1.6
7	Vendor 6	0.2096		2.1	4	0	1.427647
8							
9		Reference		DMU unde	6	Efficiency	
10	**Constrain**	set		Evaluation		1.427647	
11	price	0.2096	≤	0.2096			
12	Rejects	2.998059	≥	2.998059	Run		
13	Late delive	5.710589	≥	5.710589			
14	Σλ	1	=	1			

Fig. 8.2 Bad outputs spreadsheet model

If we do not translate the bad outputs and calculate the regular output-oriented VRS envelopment model, vendor 5 is classified as efficient, and vendor 3 is classified as inefficient. (see Fig. 8.4 where 0.0001 is used to replace 0.) The same Solver parameters shown in Fig. 8.3 are used.

If we treat the two bad outputs as inputs and use the input-oriented VRS envelopment model, we obtain the efficiency scores shown in Fig. 8.5 (Fig. 8.6 shows the Solver parameters). In this case, we do not have outputs.

8.4 Undesirable Inputs

The above discussion can also be applied to situations when some inputs need to be increased rather than decreased to improve the performance. In this case, we denote x_{ij}^I and x_{ij}^D the inputs that need to be increased and decreased, respectively.

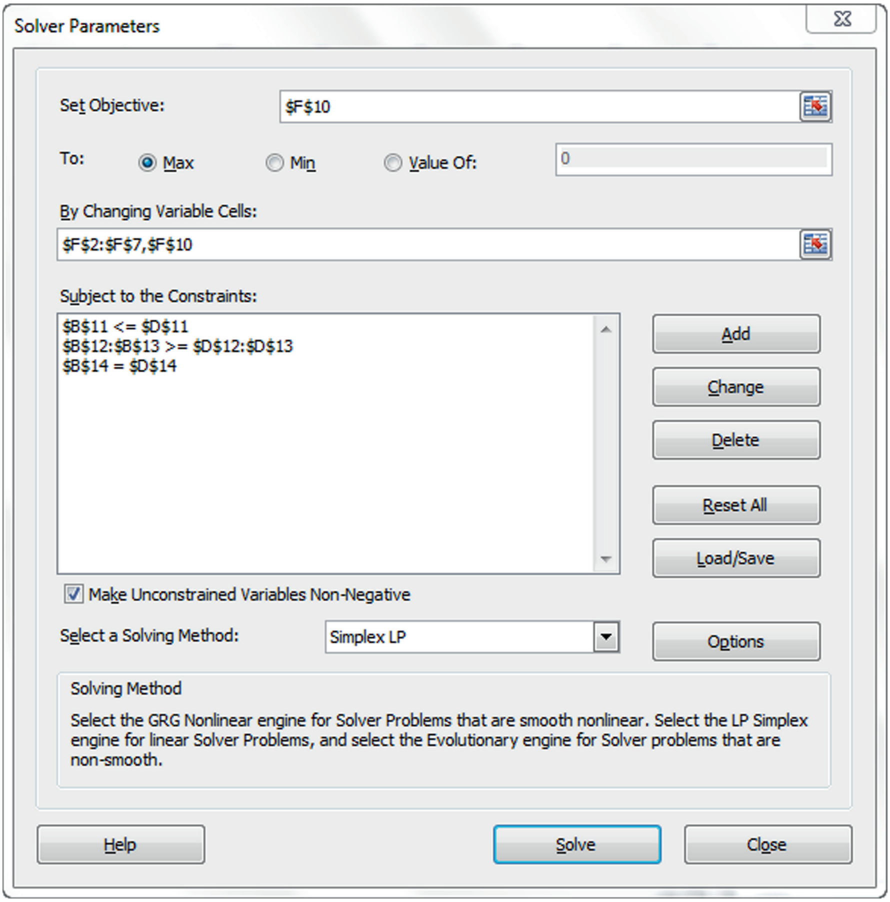


Fig. 8.3 Solver parameters for bad outputs spreadsheet model

	A	B	C	D	E	F	G
1		Price		%Rejects	% Late deliveries	λ	Efficiency
2	Vendor 1	0.1958		1.2	5	0	1.075329
3	Vendor 2	0.1881		0.8	7	0.518519	1
4	Vendor 3	0.2204		0.0001	0.0001	0	23000
5	Vendor 4	0.2081		2.1	0	0	1
6	Vendor 5	0.2118		2.3	3	0.481481	1
7	Vendor 6	0.2096		1.2	4	0	1.268519
8							
9		Reference		DMU unde	**6**	Efficiency	
10	**Constrain**	set		Evaluation		1.268519	
11	price	0.199511	≤	0.2096			
12	Rejects	1.522222	≥	1.522222	Run		
13	Late delive	5.074074	≥	5.074074			
14	Σλ	1	=	1			

Fig. 8.4 Efficiency scores when bad outputs are not translated

	A	B	C	D	E	F	G
1		Price		%Rejects	% Late deliveries	λ	Efficiency
2	Vendor 1	0.1958		1.2	5	0	0.990548
3	Vendor 2	0.1881		0.8	7	0.541904	1
4	Vendor 3	0.2204		0	0	0.122632	1
5	Vendor 4	0.2081		2.1	0	0.335464	1
6	Vendor 5	0.2118		2.3	3	0	0.944315
7	Vendor 6	0.2096		1.2	4	0	0.948332
8							
9		Reference		DMU unde	**6**	Efficiency	
10	**Constrain**	set		Evaluation		0.948332	
11	price	0.19877	≤	0.19877			
12	Rejects	1.137998	≤	1.137998	Run		
13	Late delive	3.793326	≤	3.793326			
14	Σλ	1	=	1			

Fig. 8.5 Efficiency scores when bad outputs are treated as inputs

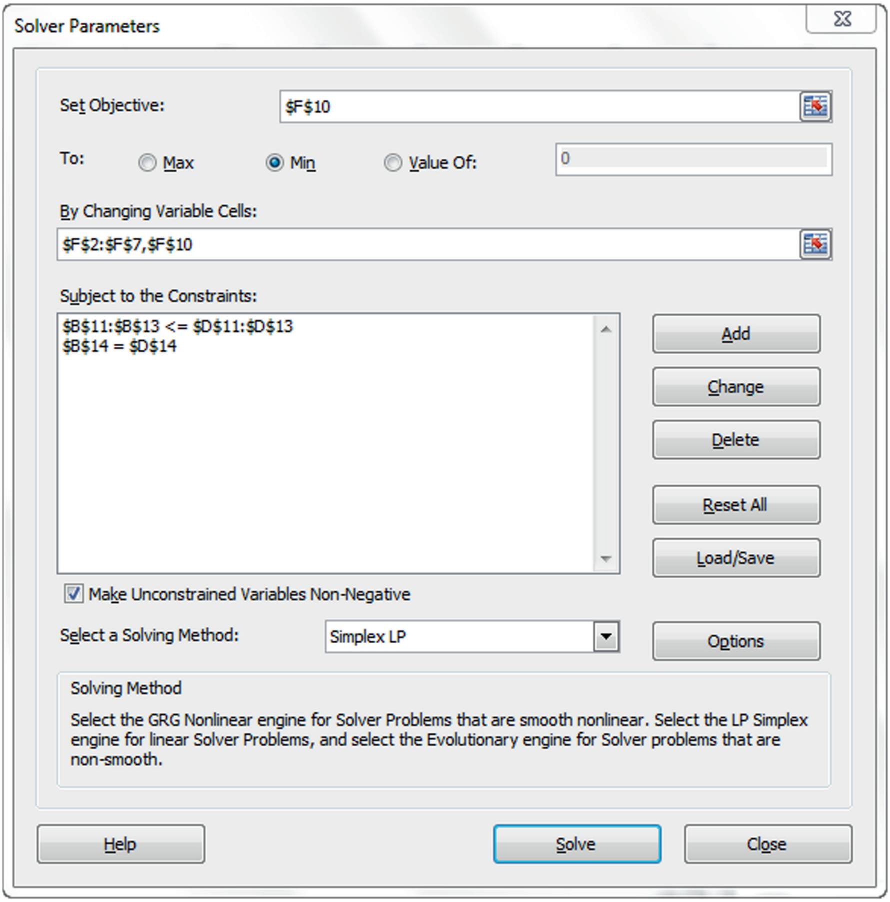


Fig. 8.6 Solver parameters when bad outputs are treated as inputs

We next multiply x_{ij}^I by "-1" and then find a proper u_i to let $\bar{x}_{ij}^I = -x_{ij}^I + u_i > 0$. Based upon model (8.1), we have

$$
\begin{aligned}
&\min \tau \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij}^D \le \tau x_{io}^D \\
&\sum_{j=1}^{n} \lambda_j \bar{x}_{ij}^I \le \tau \bar{x}_{oi}^I \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} \\
&\sum_{j=1}^{n} \lambda_j \quad = 1 \\
&\lambda_j \ge 0 \qquad j = 1,\ldots,n.
\end{aligned}
\tag{8.4}
$$

where x_{ij}^I is increased and x_{ij}^D is decreased for a DMU to improve the performance. The efficient target for DMU_o is

$$
\begin{cases}
\hat{x}_{io}^D = \tau^* x_{io}^I - s_i^{-*} \\
\hat{x}_{io}^I = u_i - (\tau^* x_{io}^I - s_i^{-*}) \\
\hat{y}_{ro} = y_{ro} + s_r^{+*}
\end{cases}
$$

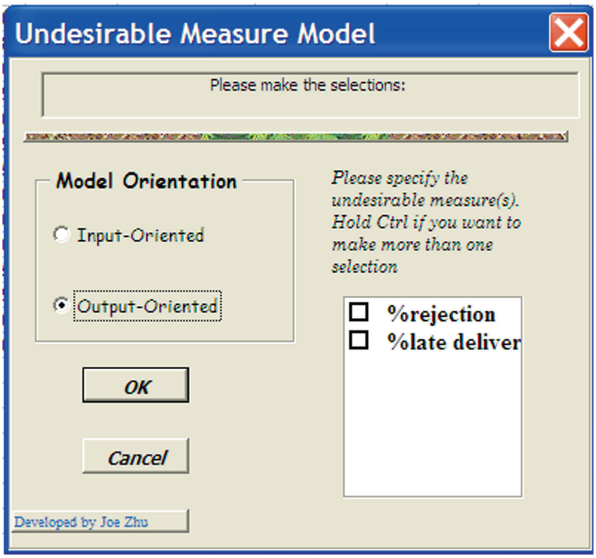
8.5 Solving DEA Using DEAFrontier Software

To run the models for treating undesirable measures, select the Undesirable-Measure Model menu item. You will be prompted with a form for selecting the DMU data sheet and then a form for specifying the models, as shown in Fig. 8.7. The results are reported in "Efficiency", "Slack", and "Target" sheets.

8.6 Negative Data

So far, we have assumed that all inputs and outputs are either positive or zero. However, we have cases where some inputs and (or) outputs are negative. For example, when a company experiences a loss, its profit is negative. Similarly, returns on some stocks can be negative. This can be easily solved by way of the *translation invari-*

Fig. 8.7 Undesirable measure models



ance property of the VRS models (Ali and Seiford 1990). Specifically, the VRS frontier remains the same if x_{ij} and y_{rj} is replaced to $\bar{x}_{ij}$ and $\bar{y}_{rj}$, respectively.

Consider the example given in the following Table where we have 10 DMUs[1]. We have two inputs x_1 = Standard Deviation and x_2 = PropNeg (proportion of negative monthly returns during the year), and three outputs y_1 = Return (average monthly return), y_2 = Skewness and y_3 = Min (minimum return).

Note that some values for return, skewness and Min are negative. In the table the average monthly return, skewness and minimum return are displaced by 3.7, 2, and 26 %, respectively so that all the output values are positive across all the DMUs. The translation values can be chosen randomly as long as the negative values become positive (Table 8.2).

Since negative data are present only in the outputs, we thus use the input-oriented VRS model. When DMU1 is under evaluation, we have $\theta_0^* = 0.75$, indicating this DMU is inefficient, and $\lambda_3^* = 0.51$ and $\lambda_4^* = 0.49$, indicating DMU3 and DMU4 are the benchmarks.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_8) contains supplementary material, which is available to authorized users.

[1] These DMUs are called commodity trading advisors (CTAs) in Wilkens and Zhu (2001).

Table 8.2 Negative data example

	Original data					Transformed data $\hat{y}_r = y_{rj} + \pi_r$		
DMU	x_1=Standard deviation (%)	x_2=Proportion negative (%)	y_1=Ave. monthly return (%)	y_2=Skewness	y_3=Minimum return (%)	$\hat{y}_1$; $\pi_1 = 3.7\%$	$\hat{y}_2$; $\pi_2 = 2$	$\hat{y}_3$; $\pi_3 = 26\%$
1	6.80	58.30	0.10	1.13	−8.10	3.80	3.13	17.90
2	4.00	41.70	0.70	0.61	−7.90	4.40	2.61	18.10
3	3.40	37.50	0.90	0.58	−4.00	4.60	2.58	22.00
4	5.00	50.00	0.60	1.7	−5.60	4.30	3.7	20.40
5	4.70	37.50	1.10	0.28	−8.20	4.80	2.28	17.80
6	3.80	50.00	−0.10	0.08	−6.30	3.60	2.08	19.70
7	11.20	45.80	3.20	0.39	−17.10	6.90	2.39	8.90
8	12.80	58.30	−1.00	0.46	−25.70	2.70	2.46	0.30
9	8.40	52.20	−1.20	−0.26	−17.10	2.50	1.74	8.90
9	5.00	54.50	0.40	1.1	−6.70	4.10	3.1	19.30
10	8.60	25.00	−3.60	−1.98	−16.50	0.10	0.02	9.50

References

Ali, A. I., & Seiford, L. M. (1990). Translation invariance in data envelopment analysis. *Operations Research Letters, 9,* 403–405.

Lovell, C. A. K., & Pastor, J. T. (1995). Units invariant and translation invariant DEA models. *Operations Research Letters, 18,* 147–151.

Pastor, J. T. (1996). Translation invariance in data envelopment analysis: A generalization. *Annals of Operations Research, 66,* 93–102.

Seiford, L. M., & Zhu, J. (2002). Modeling undesirable factors in efficiency evaluation. *European Journal of Operational Research, 142*(1), 16–20.

Weber, C. A., & Desai, A. (1996). Determination of paths to vendor market efficiency using parallel coordinates representation: A negotiation tool for buyers. *European Journal of Operational Research, 90,* 142–155.

Wilkens, K., & Zhu, J. (2001). Portfolio evaluation and benchmark selection: A mathematical programming approach. *Journal of Alternative Investments, 4*(1), 9–19.

Chapter 9
Context-dependent Data Envelopment Analysis

9.1 Introduction

Adding or deleting an inefficient DMU or a set of inefficient DMUs does not alter the efficiencies of the existing DMUs and the best-practice frontier. The inefficiency scores change only if the best-practice frontier is altered. i.e., the performance of DMUs depends only on the identified best-practice frontier. In contrast, researchers of the consumer choice theory point out that consumer choice is often influenced by the context. e.g., a circle appears large when surrounded by small circles and small when surrounded by larger ones. Similarly a product may appear attractive against a background of less attractive alternatives and unattractive when compared to more attractive alternatives (Simonson and Tversky 1992).

Considering this influence within the framework of DEA, one could ask "what is the relative attractiveness of a particular DMU when compared to others?" As in Tversky and Simonson (1993), one agrees that the relative attractiveness of DMU_x compared to DMU_y depends on the presence or absence of a third option, say DMU_z (or a group of DMUs). Relative attractiveness depends on the evaluation context constructed from alternative options (or DMUs). In the original DEA methodology, each DMU is evaluated against a set of frontier DMUs. That is, the original DEA methodology can rank the performance of inefficient DMUs with respect to the best-practice frontier. However, when DMU_x and DMU_y are members of best-practice frontier, DEA cannot identify which of DMU_x and DMU_y is a better option with respect to DMU_z (or a set of inefficient DMUs). Because both DMU_x and DMU_y have an efficiency score of one. Although one may use the super-efficiency DEA models (see Chap. 10) to rank the performance of efficient DMUs, the evaluation context or third option (the reference set) changes in each evaluation. i.e., DMU_x and DMU_y are not evaluated against the same third option by the super-efficiency concept.

From the above discussion, we see that DEA provides performance measures that are absolute in the sense that all DMUs are evaluated against the best-practice frontier. In order to obtain the relative attractiveness within DEA, Seiford and Zhu

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_9,

(2003) modify the original DEA methodology to a situation where the relative performance is defined with respect to a particular best-practice context (evaluation context).

In order to obtain the evaluation contexts, an algorithm is developed to remove the (original) best-practice frontier to allow the remaining (inefficient) DMUs to form a new second-level best-practice frontier. If we remove this new second-level best-practice frontier, a third-level best-practice frontier is formed, and so on, until no DMU is left. In this manner, we partition the set of DMUs into several levels of best-practice frontiers.

Note that each best-practice frontier provides an evaluation context for measuring the relative attractiveness. e.g., the second-level best-practice frontier serves as the evaluation context for measuring the relative attractiveness of the DMUs located on the first-level (original) best-practice frontier. It can be seen that the presence or absence (or the shape) of the second-level best-practice frontier affects the relative attractiveness of DMUs on the first-level best-practice frontier. A relative attractiveness measure is obtained when DMUs having worse performance are chosen as the evaluation context. When DMUs in a specific level are viewed as having equal performance, the attractiveness measure allows us to differentiate the "equal performance" based upon the same specific evaluation context (or third option).

On the other hand, we can measure the performance of DMUs on the third-level best-practice frontier with respect to the first or second level best-practice frontier. We define this type of measure as a progress measure where DMUs having better performance are chosen as the evaluation context. i.e., we measure the progress of DMUs with respect to best-practice frontiers at advanced levels. Note that the original DEA method provides a projection function to improve the performance of inefficient DMUs. However, it is likely that a particular inefficient DMU is unable to immediately improve its performance onto the first-level best-practice frontier because of such restrictions as management expertise, available resources, etc. Therefore intermediate (and more easily achievable) targets may be desirable for an inefficient DMU. By focusing on different levels of best-practice frontiers, the progress measure provides incremental improvements for a DMU's performance. i.e., we move the DMU step by step onto an attainable best-practice frontier. The resulting intermediate targets are *local targets*, whereas the targets on the first-level (original) best-practice frontier are *global targets*.

9.2 Stratification DEA Method

Define $\mathbf{J}^1 = \{DMU_j, j = 1,\ldots,n\}$ (the set of all n DMUs) and interactively define $\mathbf{J}^{l+1} = \mathbf{J}^l - \mathbf{E}^l$ where $\mathbf{E}^l = \{DMU_k \in \mathbf{J}^l \mid \theta^*(l,k) = 1\}$, and $\theta^*(l,k)$ is the optimal value to the following input-oriented CRS envelopment model when DMU_k is under evaluation

$$
\begin{aligned}
&\theta^*(l,k) = \min_{\lambda_j,\theta(l,k)} \theta(l,k) \\
&\text{subject to} \\
&\sum_{j\in F(\mathbf{J}^l)} \lambda_j x_{ij} \le \theta(l,k) x_{ik} \\
&\sum_{j\in F(\mathbf{J}^l)} \lambda_j y_{rj} \ge y_{rk} \\
&\lambda_j \ge 0 \quad j \in F(\mathbf{J}^l).
\end{aligned}
\tag{9.1}
$$

where $j \in F(\mathbf{J}^l)$ means $DMU_j \in \mathbf{J}^l$, i.e., $F(\bullet)$ represents the correspondence from a DMU set to the corresponding subscript index set.

When $l=1$, model (9.1) becomes the original input-oriented CRS envelopment model, and $\mathbf{E}^1$ consists of all the frontier DMUs. These DMUs in set $\mathbf{E}^1$ define the first-level best-practice frontier. When $l=2$, model (9.1) gives the second-level best-practice frontier after the exclusion of the first-level frontier DMUs. And so on. In this manner, we identify several levels of best-practice frontiers. We call $\mathbf{E}^l$ the lth-level best practice frontier. The following algorithm accomplishes the identification of these best-practice frontiers by model (9.1).

Step 1: Set $l=1$. Evaluate the entire set of DMUs, $\mathbf{J}^1$, by model (9.1) to obtain the first-level frontier DMUs, set $\mathbf{E}^1$ (the first-level best-practice frontier).

Step 2: Exclude the frontier DMUs from future DEA runs. $\mathbf{J}^{l+1} = \mathbf{J}^l - \mathbf{E}^l$. (If $\mathbf{J}^{l+1} = \varnothing$ then stop.)

Step 3: Evaluate the new subset of "inefficient" DMUs, $\mathbf{J}^{l+1}$, by model (9.1) to obtain a new set of efficient DMUs $\mathbf{E}^{l+1}$ (the new best-practice frontier).

Step 4: Let $l=l+1$. Go to step 2.

Stopping rule: $\mathbf{J}^{l+1} = \varnothing$, the algorithm stops.

Thus, model (9.1) yields a stratification of the whole set of DMUs. From the algorithm, we know that l goes from 1 to L, where L is determined by the stopping rule. Consider Fig. 1.2 in Chap. 1. DMUs 1, 2, 3 and 4 form the first-level CRS frontier and DMU5 forms the second-level CRS frontier (L=2).

It is easy to show that these sets of DMUs have the following properties

1. $\mathbf{J}^1 = \cup_{l=1}^{L} \mathbf{E}^l$ and $\mathbf{E}^l \cap \mathbf{E}^{l'} = \varnothing$ for $l \neq l'$;
2. The DMUs in $\mathbf{E}^{l'}$ are dominated by the DMUs in $\mathbf{E}^l$ if $l' > l$;
3. Each DMU in set $\mathbf{E}^l$ is efficient with respect to the DMUs in set $\mathbf{E}^{l+l'}$ for all $0 < l' \le l - L$.

We next use a data set from the DEA Dataset Repository at http://java.emp.pdx.edu/etm/dea/dataset/ to illustrate the algorithm. Table 9.1 presents the data. The data set contains 24 flexible manufacturing systems (FMS). Each FMS has five inputs (1) total cost (TC) ($millions), (2) work in process (WIP) (units), (3) throughput (TT) (hours/unit), (4) employees (EMP) (persons), and (5) space requirements (SR) (thousands of square feet), and three outputs (1) volume flexibility (VF) (average

Table 9.1 Data for the flexible manufacturing systems

FMS	Inputs					Outputs		
	TC	WIP	TT	EMP	SR	VF	PF	RF
1	1.19	98	12.33	5	5.30	619	88	2
2	4.91	297	34.84	14	1.10	841	14	4
3	4.60	418	16.68	12	6.30	555	39	1
4	3.69	147	40.83	10	3.80	778	31	2
5	1.31	377	20.82	3	9.80	628	51	6
6	3.04	173	38.87	4	1.60	266	13	5
7	1.83	202	49.67	13	4.30	46	60	4
8	2.07	533	30.07	14	8.80	226	21	4
9	3.06	898	27.67	2	3.90	354	86	5
10	1.44	423	6.02	10	5.40	694	20	3
11	2.47	470	4.00	13	5.30	513	40	5
12	2.85	87	43.09	8	2.40	884	17	7
13	4.85	915	54.79	5	2.40	439	58	4
14	1.31	852	86.87	3	0.50	401	18	4
15	4.18	924	54.46	4	6.00	491	27	4
16	1.99	273	91.08	3	2.50	937	6	3
17	1.60	983	37.93	13	8.80	709	39	2
18	4.04	106	23.39	11	2.90	615	91	3
19	3.76	955	54.98	1	9.40	499	46	3
20	4.76	416	1.55	9	1.50	58	2	6
21	3.60	660	3.98	6	3.90	592	29	4
22	3.24	771	52.26	8	1.60	535	61	1
23	3.05	318	35.09	4	9.20	124	25	2
24	1.60	849	62.83	15	7.30	923	60	3

range of production capacity per product type), (2) production mix flexibility (PF) (product types), and (3) routing flexibility (RF) (average number of operations per machining center).

Figure 9.1 shows the spreadsheet model for identifying the first level of CRS frontier. The target cell is F28, and the changing cells are K2:K25 and F28. The formulas for cells B29:B36 are

Cell B29=SUMPRODUCT(B2:B25,K2:K25)
Cell B30=SUMPRODUCT(C2:C25,K2:K25)
Cell B31=SUMPRODUCT(D2:D25,K2:K25)
Cell B32=SUMPRODUCT(E2:E25,K2:K25)
Cell B33=SUMPRODUCT(F2:F25,K2:K25)
Cell B34=SUMPRODUCT(H2:H25,K2:K25)
Cell B35=SUMPRODUCT(I2:I25,K2:K25)
Cell B36=SUMPRODUCT(J2:J25,K2:K25)

The formulas for cell D29:D36 are

Cell D29=F28*INDEX(B2:B25,E27,1)
Cell D30=F28*INDEX(C2:C25,E27,1)

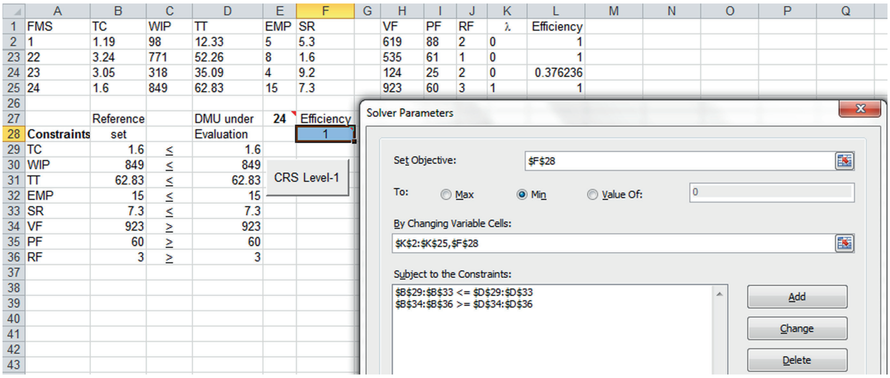


Fig. 9.1 First level CRS frontier

Cell D31=F28*INDEX(D2:D25,E27,1)
Cell D32=F28*INDEX(E2:E25,E27,1)
Cell D33=F28*INDEX(F2:F25,E27,1)
Cell D34=INDEX(H2:H25,E27,1)
Cell D35=INDEX(I2:I25,E27,1)
Cell D36=INDEX(J2:J25,E27,1)

The button "CRS Level-1" is linked to the VBA procedure "Level1" which records the CRS efficiency scores in column L.

```
Sub Level1()
    Dim i As Integer
    For i = 1 To 24
'set the value of cell E27 equal to i (1, 2,..., 24)
    Range("E27") = i
    SolverSolve UserFinish:=True
'Place the efficiency into column L
    Range("L" & i + 1) = Range("F28")
    Next i
End Sub
```

Sixteen FMSs are on the first level CRS frontier. They are DMUs 1, 2, 5, 6, 9, 10, 11, 12, 14, 16, 18, 19, 20, 21, 22, and 24.

Next, we remove those FMSs with efficiency score of one. Because no absolute references are used in the formulas in the spreadsheet shown in Fig. 9.1, the Solver automatically adjusts the parameters as we remove the rows related to the FMSs with efficiency score of one. Figure 9.2 shows the new spreadsheet.

Seven FMSs are on the second level CRS frontier. Because only one FMS, namely DMU 23, is left, this DMU forms the third level CRS frontier (L=3).

	A	B	C	D	E	F	G	H	I	J	K	L
1	FMS	TC	WIP	TT	EMP	SR		VF	PF	RF	λ	Efficiency
2	3	4.6	418	16.68	12	6.3		555	39	1	0	1
3	4	3.69	147	40.83	10	3.8		778	31	2	0	1
4	7	1.83	202	49.67	13	4.3		46	60	4	0.201	1
5	8	2.07	533	30.07	14	8.8		226	21	4	0	1
6	13	4.85	915	54.79	5	2.4		439	58	4	0.157	1
7	15	4.18	924	54.46	4	6		491	27	4	0.142	1
8	17	1.6	983	37.93	13	8.8		709	39	2	0	1
9	23	3.05	318	35.09	4	9.2		124	25	2	0	0.991748
10												
11		Reference		DMU under	**8**	Efficiency						
12	**Constraints**	set		Evaluation		0.991748						
13	TC	1.722457	≤	3.024832								
14	WIP	315.3759	≤	315.3759	CRS Level-2							
15	TT	26.31838	≤	34.80044								
16	EMP	3.966992	≤	3.966992								
17	SR	2.093367	≤	9.124083								
18	VF	147.8423	≥	124								
19	PF	25	≥	25								
20	RF	2	≥	2								

Fig. 9.2 Second level CRS frontier

9.3 Input-oriented Context-dependent DEA

The DEA stratification model (9.1) partitions the set of DMUs into different frontier levels characterized by $\mathbf{E}^l$ ($l = 1, \ldots, L$). We present the input-oriented context-dependent DEA based upon the evaluation context $\mathbf{E}^l$. The context-dependent DEA is characterized by an attractiveness measure and a progress measure.

9.3.1 *Attractiveness*

Consider a specific $DMU_q = (x_q, y_q)$ from a specific level $\mathbf{E}^{l_o}$, $l_o \in \{1, \ldots, L-1\}$. We have the following model to characterize the attractiveness.

$$
\begin{aligned}
& H_q^*(d) = \min \ H_q(d) \qquad d = 1, \ldots, L - l_o \\
& \text{subject to} \\
& \sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d})} \lambda_j x_j \le H_q(d) x_q \\
& \sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d})} \lambda_j y_j \ge y_q \\
& \lambda_j \ge 0 \quad j \in \mathrm{F}(\mathbf{E}^{l_o+d})
\end{aligned}
\tag{9.2}
$$

Lemma 9.1 For a specific $DMU_q \in \mathbf{E}^{l_o}$, $l_o \in \{1,\ldots, L-1\}$, model (9.2) is equivalent to the following linear programming problem

$$\begin{aligned}
&\tilde{H}_q^*(d) = \min_{\lambda_j, \tilde{H}_q(d)} \tilde{H}_q(d) \qquad d = 1,\ldots,L-l_o \\
&\text{subject to} \\
&\sum_{j \in \mathrm{F}(\mathbf{J}^{l_o+d})} \lambda_j x_j \le \tilde{H}_q(d) x_q \\
&\sum_{j \in \mathrm{F}(\mathbf{J}^{l_o+d})} \lambda_j y_j \ge y_q \\
&\lambda_j \ge 0 \quad j \in \mathrm{F}(\mathbf{J}^{l_o+d})
\end{aligned} \tag{9.3}$$

[Proof]: Note that $\mathbf{J}^{l_o+d} = \bigcup_{l=\alpha}^{L-l_o} \mathbf{E}^{l_o+l}$. Therefore, (9.3) can be rewritten as

$$\begin{aligned}
&\tilde{H}_q^*(d) = \min_{\mu_j, \lambda_j, \tilde{H}_q(d)} \tilde{H}_q(d) \qquad d = 1,\ldots,L-l_o \\
&\text{subject to} \\
&\sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d})} \mu_j x_j + \sum_{j \in \mathrm{F}(\mathbf{J}^{l_o+d} - \mathbf{E}^{l_o+d})} \lambda_j x_j \le \tilde{H}_q(d) x_q \\
&\sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d})} \mu_j y_j + \sum_{j \in \mathrm{F}(\mathbf{J}^{l_o+d} - \mathbf{E}^{l_o+d})} \lambda_j y_j \ge y_q \\
&\mu_j, \lambda_j \ge 0
\end{aligned} \tag{9.4}$$

Obviously, $\lambda_j = 0$ for all $j \in F(\mathbf{J}^{l_o+d} - \mathbf{E}^{l_o+d}) = F(\cup_{l=d+1}^{L-l_o} \mathbf{E}^{l_o+l})$ in any optimal solutions to (9.4). Otherwise, if some $\lambda_j \ne 0$, then $\mathbf{E}^{l_o+d} \cap (\cup_{l=d+1}^{L-l_o} \mathbf{E}^{l_o+l}) \ne \varnothing$ (A contradiction). Therefore, $\tilde{\Omega}_q^*(d) = \Omega_q^*(d)$ and (9.2) is equivalent to (9.3).■

Theorem 9.1 For a specific $DMU_q \in \mathbf{E}^{l_o}$, $l_o \in \{1,\ldots, L-1\}$, we have

1. $H_q^*(d) > 1$ for each $g = 1,\ldots,l_o - 1$.
2. $H_q^*(d+1) > H_q^*(d)$.

[Proof]:

1. Suppose $H_q^*(d) < 1$.

If $H_q^*(d) = 1$, then $DMU_q \in \mathbf{E}^{l_o+d}$. This means that $\mathbf{E}^{l_o+d} \cap \mathbf{E}^{l_o} \ne \varnothing$. A contradiction.

If $H_q^*(d) < 1$, then DMU_q is dominated by $\mathbf{E}^{l_o+d}$. However, $\mathbf{E}^{l_o+d}$ is dominated by $\mathbf{E}^{l_o}$. Thus, DMU_q is dominated by $\mathbf{E}^{l_o}$. This means that $DMU_q \notin \mathbf{E}^{l_o}$. A contradiction. Therefore, $H_q^*(d) > 1$.

2. $H_q^*(d+1)$ is obtained by solving the following problem

$$\begin{aligned} & H_q^*(d+1) = \min_{\lambda_j, H_q(d+1)} H_q(d+1) \\ & \text{subject to} \\ & \sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d+1})} \lambda_j x_j \le H_q(d+1) x_q \\ & \sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d+1})} \lambda_j y_j \ge y_q \\ & \lambda_j \ge 0 \quad j \in \mathrm{F}(\mathbf{E}^{l_o+d+1}) \end{aligned} \tag{9.5}$$

$H_q^*(d)$ is obtained by solving the following problem

$$\begin{aligned} & H_q^*(d) = \min_{\mu_j, H_q(d)} H_q(d) \\ & \text{subject to} \\ & \sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d})} \mu_j x_j \le H_q(d) x_q \\ & \sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d})} \mu_j y_j \ge y_q; \\ & \mu_j \ge 0 \quad j \in \mathrm{F}(\mathbf{E}^{l_o+d}) \end{aligned}$$

which is, by Lemma 9.1, equivalent to the following linear programming problem

$$\begin{aligned} & H_q^*(d) = \min_{\mu_j, \lambda_j, H_q(d)} H_q(d) \\ & \text{subject to} \\ & \sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d})} \mu_j x_j + \sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d+1})} \lambda_j x_j \le H_q(d) x_q \\ & \sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d})} \mu_j y_j + \sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d+1})} \lambda_j y_j \ge y_q; \\ & \mu_j, \lambda_j \ge 0 \end{aligned} \tag{9.6}$$

It can be seen that any optimal solution to (9.5) is a feasible solution to (9.6). Therefore $H_q^*(d+1) \ge H_q^*(d)$. However, if $H_q^*(d+1) = H_q^*(d)$, then $\mathbf{E}^{l_o+d} \cap \mathbf{E}^{l_o+d+1} \ne \varnothing$. Thus, $H_q^*(d+1) > H_q^*(d)$.■

Definition 9.1 $H_q^*(d)$ is called (input-oriented) *d-degree* attractiveness of DMU_q from a specific level $\mathbf{E}^{l_o}$.

	A	B	C	D	E	F	G	H	I	J	K
1	FMS (Level1)	TC	WIP	TT	EMP	SR		VF	PF	RF	Attractiveness
2	1	1.19	98	12.33	5	5.3	16	619	88	2	5.257398
3	2	4.91	297	34.84	14	1.1		841	14	4	3.883129
4	5	1.31	377	20.82	3	9.8	Score	628	51	6	3.942397
5	6	3.04	173	38.87	4	1.6	1.594	266	13	5	3.222047
6	9	3.06	898	27.67	2	3.9		354	86	5	3.706897
7	10	1.44	423	6.02	10	5.4		694	20	3	5.347655
8	11	2.47	470	4	13	5.3		513	40	5	10.45603
9	12	2.85	87	43.09	8	2.4		884	17	7	4.625919
10	14	1.31	852	86.87	3	0.5		401	18	4	4.8
11	16	1.99	273	91.08	3	2.5		937	6	3	3.334894
12	18	4.04	106	23.39	11	2.9		615	91	3	3.528959
13	19	3.76	955	54.98	1	9.4		499	46	3	4.845534
14	20	4.76	416	1.55	9	1.5		58	2	6	29.1
15	21	3.6	660	3.98	6	3.9		592	29	4	9.24551
16	22	3.24	771	52.26	8	1.6		535	61	1	1.789802
17	24	1.6	849	62.83	15	7.3		923	60	3	1.593583
18	DMU under evaluation	2.5497	1353	100.12	23.9	11.63		923	60	3	
19		≥	≥	≥	≥	≥		≤	≤	≤	
20	Evaluation background	2.5497	1227.1	57.697	18.73	11.63		923	60	3.27	
21	FMS(Level2)	TC	WIP	TT	EMP	SR		VF	PF	RF	λ
22	3	4.6	418	16.68	12	6.3		555	39	1	0
23	4	3.69	147	40.83	10	3.8		778	31	2	0.082849
24	7	1.83	202	49.67	13	4.3		46	60	4	0.177585
25	8	2.07	533	30.07	14	8.8		226	21	4	0
26	13	4.85	915	54.79	5	2.4		439	58	4	0
27	15	4.18	924	54.46	4	6		491	27	4	0
28	17	1.6	983	37.93	13	8.8		709	39	2	1.1994

Fig. 9.3 First degree attractiveness spreadsheet model

Suppose, e.g., each DMU in the first-level best practice frontier represents an option, or product. Customers may compare a specific DMU in $\mathbf{E}^{l_o}$ with other alternatives that are currently in the same level as well as with relevant alternatives that serve as evaluation contexts. The relevant alternatives are those DMUs, say, in the second or third level best-practice frontier, etc. Given the alternatives (evaluation contexts), model (9.2) enables us to select the best option—the most attractive one.

In model (9.2), each best-practice frontier of $\mathbf{E}^{l_o+d}$ represents an evaluation context for measuring the relative attractiveness of DMUs in $\mathbf{E}^{l_o}$. The larger the value of $H_q^*(d)$, the more attractive the DMU_q is. Because this DMU_q makes itself more distinctive from the evaluation context $\mathbf{E}^{l_o+d}$. We are able to rank the DMUs in $\mathbf{E}^{l_o}$ based upon their attractiveness scores and identify the best one.

Figure 9.3 shows a spreadsheet model for the attractiveness measure—model (9.2). Cells A1:J17 and A21:J28 store the DMUs in the first and second levels, respectively. This spreadsheet model measures the first-degree attractiveness. Cell G2 is reserved to indicate the DMU under evaluation. Cell G5 represents the attractiveness score, and is the target cell and a changing cell. Cells K22:K28 are reserved for changing cells of λ_j.

Cells B20:F20, and cells H20:J20 contain formulas for the reference set (DMUs in the second level CRS frontier). The formula for cell B20 is "=SUMPRODUCT (B22:B28,$K$22:$K$28)" and is copied into cells C20:F20 and cells H20:J20.

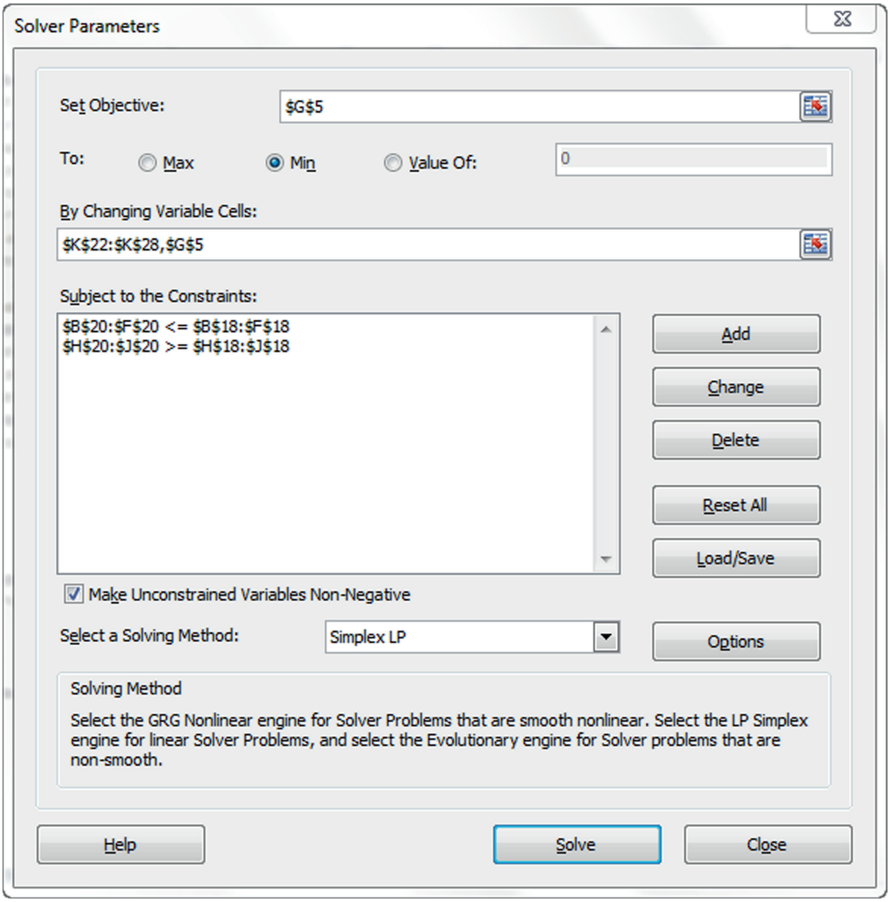


Fig. 9.4 Solver parameters for first degree attractiveness

Cells B18:F18, and cells H18:J18 contain formulas for the DMU under evaluation (DMUs in the first level CRS frontier). The formula for cell B18 is "=$G$5*INDEX(B2:B17,$G$2,1)" and is copied into cells C18:F18. The formula for cell H18 is "=INDEX(H2:H17,$G$2,1)" and is copied into cells I18:J18.

Figure 9.4 shows the Solver parameters. After solve the model for the first DMU, we use the VBA procedure "Attractiveness" to obtain the attractiveness scores for the remaining DMUs.

	A	B	C	D	E	F	G	H	I	J	K
1	FMS (Level1)	TC	WIP	TT	EMP	SR		VF	PF	RF	Attractiveness
2	1	1.19	98	12.33	5	5.3	16	619	88	2	16.19832
3	2	4.91	297	34.84	14	1.1		841	14	4	56.72434
4	5	1.31	377	20.82	3	9.8	Score	628	51	6	11.79143
5	6	3.04	173	38.87	4	1.6	14.19	266	13	5	14.375
6	9	3.06	898	27.67	2	3.9		354	86	5	8.114872
7	10	1.44	423	6.02	10	5.4		694	20	3	32.62306
8	11	2.47	470	4	13	5.3		513	40	5	36.29268
9	12	2.85	87	43.09	8	2.4		884	17	7	27.32796
10	14	1.31	852	86.87	3	0.5		401	18	4	59.50323
11	16	1.99	273	91.08	3	2.5		937	6	3	27.80774
12	18	4.04	106	23.39	11	2.9		615	91	3	15.73415
13	19	3.76	955	54.98	1	9.4		499	46	3	16.09677
14	20	4.76	416	1.55	9	1.5		58	2	6	67.91613
15	21	3.6	660	3.98	6	3.9		592	29	4	42.09207
16	22	3.24	771	52.26	8	1.6		535	61	1	24.80847
17	24	1.6	849	62.83	15	7.3		923	60	3	14.18926
18	DMU under evaluation	22.703	12047	891.51	212.8	103.6		923	60	3	
19		≤	≤	≤	≤	≤		≥	≥	≥	
20	Evaluation background	22.703	2367	261.19	29.77	68.48		923	186.1	14.9	
21	FMS(Level3)	TC	WIP	TT	EMP	SR		VF	PF	RF	λ
22	23	3.05	318	35.09	4	9.2		124	25	2	7.443548

Fig. 9.5 Second degree attractiveness spreadsheet model

```
Sub Attractiveness()
 Dim i As Integer
    For i = 1 To 16
'set the value of cell G2 equal to i (1, 2,..., 16)
    Range("G2") = i
    SolverSolve UserFinish:=True
'Place the attractiveness score in cell G5 into column K
    Range("K" & i + 1) = Range("G5")
    Next
End Sub
```

If we change the evaluation background to the third level CRS frontier (DMU23), we obtain the spreadsheet model for measuring the second degree attractiveness (see Fig. 9.5). This spreadsheet can be obtained via replacing the second level CRS frontier by the DMU23 in Fig. 9.3.

Based upon the attractiveness scores shown in Fig. 9.3, DMU20 and DMU11 are ranked as the top two most attractive systems. However, if we change the evaluation context to the third level CRS frontier, DMU11 is ranked fifth, and DMU14 becomes the second most attractiveness system. This example illustrates that under a different evaluation context, the attractiveness of DMUs on the same level may be different. Therefore, the context-dependent DEA can differentiate the performance of efficient DMUs, or DMUs on the same performance level.

9.3.2 Progress

Consider the following linear programming problem for determining the progress measure for $DMU_q \in \mathbf{E}^{l_o}$, $l_o \in \{2,\ldots, L\}$.

$$
\begin{aligned}
& G_q^*(g) = \min_{\lambda_j, G_q(g)} G_q(g) \qquad g = 1,\ldots,l_o - 1 \\
& \text{subject to} \\
& \sum_{j \in \mathrm{F}(\mathbf{E}^{l_o - g})} \lambda_j x_j \le G_q(g) x_q \\
& \sum_{j \in \mathrm{F}(\mathbf{E}^{l_o - g})} \lambda_j y_j \ge y_q \\
& \lambda_j \ge 0 \quad j \in \mathrm{F}(\mathbf{E}^{l_o - g})
\end{aligned}
\tag{9.7}
$$

Lemma 9.2 For a specific $DMU_q \in \mathbf{E}^{l_o}$, $l_o \in \{2, \ldots, L\}$, model (9.7) is equivalent to the following linear programming problem

$$
\begin{aligned}
& \tilde{G}_q^*(g) = \min_{\lambda_j, \tilde{G}_q(\beta)} \tilde{G}_q(g) \\
& \text{subject to} \\
& \sum_{j \in \mathrm{F}(\mathbf{J}^{l_o - g})} \lambda_j x_j \le \tilde{G}_q(g) x_q \\
& \sum_{j \in \mathrm{F}(\mathbf{J}^{l_o - g})} \lambda_j y_j \ge y_q \\
& \lambda_j \ge 0 \quad j \in \mathrm{F}(\mathbf{J}^{l_o - g})
\end{aligned}
\tag{9.8}
$$

[Proof]: Note that $\mathbf{J}^{l_o - g} = \mathbf{E}^{l_o - g} \cup \mathbf{J}^{l_o - g + 1}$, since all DMUs in $\mathbf{J}^{l_o - g}$ are dominated by the frontiers constructed by the DMUs in $\mathbf{E}^{l_o - g}$. Therefore, by the nature of DEA method, we know that $\lambda_j = 0$ ($j \notin F(\mathbf{E}^{l_o - g})$) in any optimal solutions to (9.8). Thus, $\tilde{G}_q^*(g) = G_q^*(g)$ and (9.7) is equivalent to (9.8).■

Theorem 9.2 For a specific $DMU_q \in \mathbf{E}^{l_o}$, $l_o \in \{2,\ldots, L\}$, we have

3. $G_q^*(g) < 1$ for each $g = 1,\ldots,l_o - 1$.
4. $G_q^*(g+1) < G_q^*(g)$.

[Proof]: The proof is similar to that of Theorem 9.1 by using Lemma 9.2 and therefore is omitted.■

Definition 9.2 $M_q^*(g) \equiv 1 / G_q^*(g)$ is called (input-oriented) *g-degree* progress of DMU_q from a specific level $\mathbf{E}^{l_o}$.

Obviously $M_q^*(g) > 1$. For a larger $M_q^*(g)$, more progress is expected. Each best-practice frontier, $\mathbf{E}^{l_o-g}$, contains a possible target for a specific DMU in $\mathbf{E}^{l_o}$ to improve its performance. The progress here is a level-by-level improvement.

Now consider the following linear programming problem

$$
\begin{aligned}
&\max \ \|S^+(g)\|_1 + \|S^-(g)\|_1 \qquad g = 1,\ldots,l_o - 1\\
&\text{subject to}\\
&S^-(g) = G_q^*(g)x_q - \sum_{j\in \mathrm{F}(E^{l_o-g})} \lambda_j x_j\\
&S^+(g) = \sum_{j\in \mathrm{F}(E^{l_o-g})} \lambda_j y_j - y_q\\
&S^+(g), S^-(g) \ge 0\\
&\lambda_j \ge 0 \quad j \in \mathrm{F}(E^{l_o-g})
\end{aligned}
\tag{9.9}
$$

where $\|S^+(g)\|_1$ and $\|S^-(g)\|_1$ represent L_1-*norm*s for $S^+(g) = (s_1^+(g),\ldots,s_s^+(g))$ and $S^-(g) = (s_1^-(g),\ldots,s_m^-(g))$, respectively, i.e., $\|S^+(g)\|_1 + \|S^-(g)\|_1 = \sum_{r=1}^{s} s_r^+(g) + \sum_{i=1}^{m} s_i^-(g)$.

Definition 9.3 (Global Efficient Target and Local Efficient Target). The following point

$$
\begin{cases}
\hat{y}_q = G_q^*(g)y_q + S^{+*}(g)\\
\hat{x}_q = x_q - S^{-*}(g)
\end{cases}
$$

is the *global efficient target* for $DMU_q \in \mathbf{E}^{l_o}$, $l_o \in \{2,\ldots, L\}$ if $g = l_o - 1$; Otherwise, if $g < l_o - 1$, it represents a *local efficient target*, where $G_q^*(g)$ is the optimal value to (9.7), and $S^{+*}(g)$ and $S^{-*}(g)$ represent the optimal values in (9.9).

It can be seen that if the first-level best-practice frontier is chosen as the evaluation context $(g = l_o - 1)$, then we obtain the global efficient target, i.e., the original DEA efficient target. The local efficient targets are obtained when other best-practice frontiers are selected. Although the local efficient targets are not non-dominated points compared to the DMUs in $\mathbf{E}^1$, they may represent a better alternative and an attainable target for a specific inefficient DMU. That is, in the presence of possible external or internal restrictions, a DMU may be unable to move itself onto the first-level best-practice frontier (global efficient target). Thus, our progress measure extends the original DEA projection function and enables an inefficient DMU to improve its performance at a reasonable and desirable scale.

Figure 9.6 shows a spreadsheet for the progress measure where the evaluation background is the second level DMUs and the DMU under evaluation is the DMU23.

	A	B	C	D	E	F	G	H	I	J	K
1	FMS(Level3)	TC	WIP	TT	EMP	SR		VF	PF	RF	Progress
2	23	3.05	318	35.09	4	9.2	1	124	25	2	0.991748
3											
4	DMU under evaluation	3.0248	315.38	34.8	3.97	9.124		124	25	2	
5		≥	≥	≥	≥	≥		≤	≤	≤	
6	Evaluation background	1.7225	315.38	26.318	3.97	2.093		147.8	25	2	
7	FMS(Level2)	TC	WIP	TT	EMP	SR		VF	PF	RF	λ
8	3	4.6	418	16.68	12	6.3		555	39	1	0
9	4	3.69	147	40.83	10	3.8		778	31	2	0
10	7	1.83	202	49.67	13	4.3		46	60	4	0.201125
11	8	2.07	533	30.07	14	8.8		226	21	4	0
12	13	4.85	915	54.79	5	2.4		439	58	4	0.156867
13	15	4.18	924	54.46	4	6		491	27	4	0.142008
14	17	1.6	983	37.93	13	8.8		709	39	2	0

Fig. 9.6 First degree progress spreadsheet model

The formula for cell B4 is "=$K$2*INDEX(B2,$G$2,1)", where cell K2 is the target cell (its reciprocal represents the first-degree progress). This formula is copied into cells C4:F4.

The formula for H4 is "=INDEX(H2,$G$2,1)" and is copied into cells I4:J4

The formula for cell B6 is "=SUMPRODUCT(B8:B14,$K$8:$K$14)" and is copied into cells C6:F6, and cells H6:J6.

Figure 9.7 shows the Solver parameters for the spreadsheet model shown in Fig. 9.6. In Fig. 9.6, the first degree progress score for DMU23 is 1/0.99175 = 1.0083. The optimal values in cells B6:F6 and cells H6:J6 represent the local target for DMU23.

If we replace the second-level DMUs with the first-level DMUs, we obtain the second-degree progress measure for DMU23. Figure 9.8 shows the spreadsheet.

This spreadsheet model is actually the input-oriented CRS envelopment model when DMU23 is under evaluation. Figure 9.9 shows the Solver parameters for the model shown in Fig. 9.8. The second degree progress score for DMU23 is 1/0.37624 = 2.6579, and the optimal values in cells B6:F6 and H6:J6 represent the global target for DMU23

9.4 Output-oriented Context-dependent DEA

Similar to the discussion on the input-oriented context-dependent DEA, for a specific $DMU_q = (x_q, y_q)$ from a specific level $\mathbf{E}^{l_o}$, $l_o \in \{1, ..., L-1\}$, we have the following model to characterize the output-oriented attractiveness

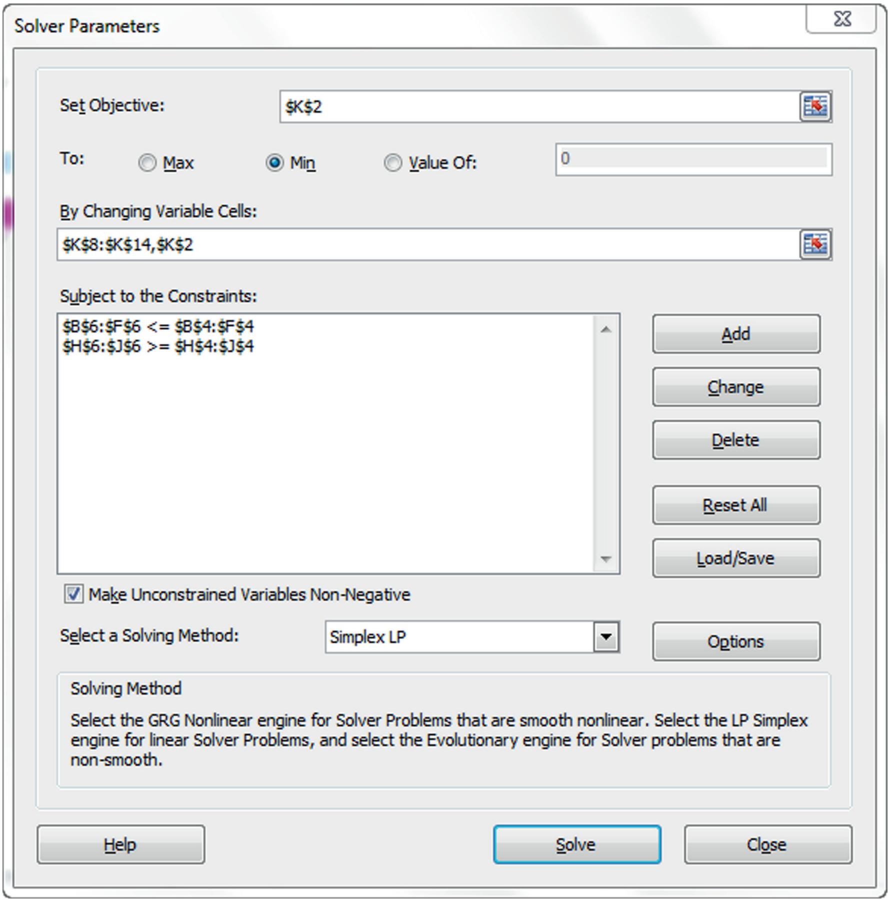


Fig. 9.7 Solver parameters for first degree progress

$$
\begin{aligned}
&\Omega_q^*(d) = \max_{\lambda_j, \Omega_q(d)} \Omega_q(d) \qquad d = 1, \ldots, L - l_o \\
&\text{subject to} \\
&\sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d})} \lambda_j y_j \geq \Omega_q(d) y_q \\
&\sum_{j \in \mathrm{F}(\mathbf{E}^{l_o+d})} \lambda_j x_j \leq x_q \\
&\lambda_j \geq 0 \quad j \in \mathrm{F}(\mathbf{E}^{l_o+d})
\end{aligned}
\tag{9.10}
$$

Similar to Theorem 9.1, we have

	A	B	C	D	E	F	G	H	I	J	K
1	FMS(Level3)	TC	WIP	TT	EMP	SR		VF	PF	RF	Progress
2	23	3.05	318	35.09	4	9.2	1	124	25	2	0.376236
3											
4	DMU under evaluation	1.1475	119.64	13.202	1.5	3.461		124	25	2	
5		≥	≥	≥	≥	≥		≤	≤	≤	
6	Evaluation background	0.5355	119.64	7.7206	1.5	3.434		258.6	25	2	
7	FMS(Level1)	TC	WIP	TT	EMP	SR		VF	PF	RF	λ
8	1	1.19	98	12.33	5	5.3		619	88	2	0.117385
9	2	4.91	297	34.84	14	1.1		841	14	4	0
10	5	1.31	377	20.82	3	9.8		628	51	6	0.285026
11	6	3.04	173	38.87	4	1.6		266	13	5	0
12	9	3.06	898	27.67	2	3.9		354	86	5	0
13	10	1.44	423	6.02	10	5.4		694	20	3	0
14	11	2.47	470	4	13	5.3		513	40	5	0
15	12	2.85	87	43.09	8	2.4		884	17	7	0.007867
16	14	1.31	852	86.87	3	0.5		401	18	4	0
17	16	1.99	273	91.08	3	2.5		937	6	3	0
18	18	4.04	106	23.39	11	2.9		615	91	3	0
19	19	3.76	955	54.98	1	9.4		499	46	3	0
20	20	4.76	416	1.55	9	1.5		58	2	6	0
21	21	3.6	660	3.98	6	3.9		592	29	4	0
22	22	3.24	771	52.26	8	1.6		535	61	1	0
23	24	1.6	849	62.83	15	7.3		923	60	3	0

Fig. 9.8 Second degree progress spreadsheet model

Theorem 9.3 For a specific $DMU_q \in \mathbf{E}^{l_o}$, $l_o \in \{1,\ldots, L-1\}$, we have

5. $\Omega_q^*(d) < 1$ for each $d = 1,\ldots,L-l_o$.
6. $\Omega_q^*(d+1) < \Omega_q^*(d)$.

Definition 9.4 $\mathrm{A}_q^*(d) \equiv 1/\Omega_q^*(d)$ is called the (output-oriented) *d-degree* attractiveness of DMU_q from a specific level $\mathbf{E}^{l_o}$.

Note that $\mathrm{A}_q^*(d)$ is the reciprocal of the optimal value to (9.10), therefore $\mathrm{A}_q^*(d) > 1$. The larger the value of $\mathrm{A}_q^*(d)$, the more attractive the DMU_q is. Because this DMU_q makes itself more distinctive from the evaluation context $\mathbf{E}^{l_o+d}$. We are able to rank the DMUs in based upon their attractiveness scores and identify the best one.

Next, consider the following linear programming problem for determining the progress measure for $DMU_q \in \mathbf{E}^{l_o}$, $l_o \in \{2,\ldots, L\}$.

$$
\begin{aligned}
& P_q^*(g) = \max_{\lambda_j, P_q(g)} P_q(g) \qquad g = 1,\ldots,l_o - 1 \\
& \text{subject to} \\
& \sum_{j \in \mathrm{F}(E^{l_o-g})} \lambda_j y_j \geq P_q(g) y_q \\
& \sum_{j \in \mathrm{F}(E^{l_o-g})} \lambda_j x_j \leq x_q \\
& \lambda_j \geq 0 \quad j \in \mathrm{F}(E^{l_o-g})
\end{aligned} \tag{9.11}
$$

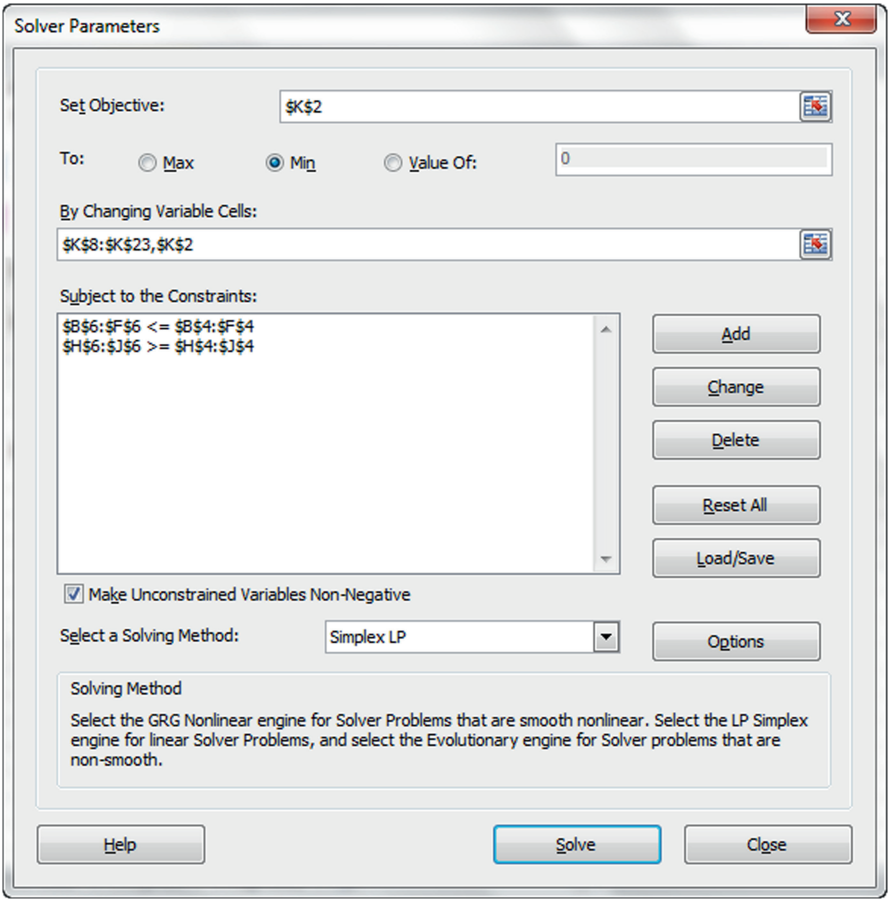


Fig. 9.9 Solver parameters for second degree progress

Similar to Theorem 9.2, we have

Theorem 9.4 For a specific $DMU_q \in \mathbf{E}^{l_o}$, $l_o \in \{2, \ldots, L\}$, we have

7. $P_q^*(g) > 1$ for each $g = 1, \ldots, l_o - 1$.
8. $P_q^*(g+1) > P_q^*(g)$.

Definition 9.5 The optimal value to (9.11), i.e., $P_q^*(g)$, is called the (output-oriented) *g-degree* progress of DMU_q from a specific level $\mathbf{E}^{l_o}$.

For a larger $P_q^*(g)$, more progress is expected for DMU_q. Thus, a smaller value of $P_q^*(g)$ is preferred. To obtain the efficient target, we consider

$$\begin{aligned}
&\max \quad \| S^{+}(g) \|_1 + \| S^{-}(g) \|_1 \qquad g = 1,\ldots,l_o - 1 \\
&\text{subject to} \\
&S^{+}(g) = \sum_{j \in \mathrm{F}(E^{l_o - g})} \lambda_j y_j - P_q^{*}(g) y_q \\
&S^{-}(g) = x_q - \sum_{j \in \mathrm{F}(E^{l_o - g})} \lambda_j x_j \\
&S^{+}(g), S^{-}(g) \ge 0 \\
&\lambda_j \ge 0 \quad j \in \mathrm{F}(E^{l_o - g})
\end{aligned} \tag{9.12}$$

Definition 9.6 (Global Efficient Target and Local Efficient Target) The following point

$$\begin{cases} \hat{y}_q = P_q^{*}(g) y_q + S^{+*}(g) \\ \hat{x}_q = x_q - S^{-*}(g) \end{cases}$$

is the *global efficient target* for $DMU_q \in \mathbf{E}^{l_o}$, $l_o \in \{2,\ldots, L\}$ if $g = l_o - 1$; otherwise, if $g < l_o - 1$, it represents a *local efficient target*, where $P_q^{*}(g)$ is the optimal value to (9.11), and $S^{+*}(g)$ and $S^{-*}(g)$ represent the optimal values in (9.12).

The relationship between the input-oriented and output-oriented context-dependent DEA can be summarized in the following Theorem.

Theorem 9.5

$$H_q^{*}(d) = 1/\Omega_q^{*}(d), \quad \text{and} \quad G_q^{*}(g) = 1/P_q^{*}(g).$$

Theorem 9.5 indicates that the output-oriented attractiveness and progress measures can be obtained from the input-oriented context-dependent DEA. However, Theorem 9.5 is not necessarily true when the frontiers do not exhibit CRS.

Figure 9.10 shows the output-oriented spreadsheet for the first-degree attractiveness for DMUs in the first-level CRS frontier. This spreadsheet is similar to the one shown in Fig. 9.3. The formula for cell B18 is changed to

Cell B18 =INDEX(B2:B17,$G$2,1)

and is copied into cells C18:F18.
The formula for cell H18 is changed to

Cell H18 =$G$5*INDEX(H2:H17,$G$2,1)

where cell G5 represents the output-oriented attractiveness measure. This formula is then copied into cells I18:J18.

	A	B	C	D	E	F	G	H	I	J	K	L
1	FMS (Level1)	TC	WIP	TT	EMP	SR		VF	PF	RF	Attractiveness	input-oriented
2	1	1.19	98	12.33	5	5.3	16	619	88	2	0.190208157	5.257398084
3	2	4.91	297	34.84	14	1.1		841	14	4	0.257524313	3.88312851
4	5	1.31	377	20.82	3	9.8	Score	628	51	6	0.253652805	3.942396767
5	6	3.04	173	38.87	4	1.6	0.628	266	13	5	0.310361728	3.222046761
6	9	3.06	898	27.67	2	3.9		354	86	5	0.269767442	3.706896552
7	10	1.44	423	6.02	10	5.4		694	20	3	0.186997849	5.347655092
8	11	2.47	470	4	13	5.3		513	40	5	0.095638588	10.45603059
9	12	2.85	87	43.09	8	2.4		884	17	7	0.216173255	4.62591916
10	14	1.31	852	86.87	3	0.5		401	18	4	0.208333333	4.8
11	16	1.99	273	91.08	3	2.5		937	6	3	0.299859601	3.334894047
12	18	4.04	106	23.39	11	2.9		615	91	3	0.283369716	3.528958618
13	19	3.76	955	54.98	1	9.4		499	46	3	0.20637561	4.845533835
14	20	4.76	416	1.55	9	1.5		58	2	6	0.034364261	29.1
15	21	3.6	660	3.98	6	3.9		592	29	4	0.108160605	9.245510401
16	22	3.24	771	52.26	8	1.6		535	61	1	0.558720872	1.789802475
17	24	1.6	849	62.83	15	7.3		923	60	3	0.627516581	1.593583388
18	DMU under evaluation	1.6	849	62.83	15	7.3		579.2	37.65	1.88	Attractiveness	
19		≥	≥	≥	≥	≥		≤	≤	≤		
20	Evaluation background	1.6	770	36.206	11.75	7.3		579.2	37.65	2.06		
21	FMS(Level2)	TC	WIP	TT	EMP	SR		VF	PF	RF	λ	
22	3	4.6	418	16.68	12	6.3		555	39	1	0	
23	4	3.69	147	40.83	10	3.8		778	31	2	0.0519893	
24	7	1.83	202	49.67	13	4.3		46	60	4	0.111437381	
25	8	2.07	533	30.07	14	8.8		226	21	4	0	
26	13	4.85	915	54.79	5	2.4		439	58	4	0	
27	15	4.18	924	54.46	4	6		491	27	4	0	
28	17	1.6	983	37.93	13	8.8		709	39	2	0.752643173	

Fig. 9.10 Output-oriented first degree attractiveness spreadsheet model

To obtain the Solver parameters for the model shown in Fig. 9.10, we change the "Min" to "Max" in Fig. 9.4, as shown in Fig. 9.11. The results are reported in cells K2:K17. Cells L2:L17 report the input-oriented attractiveness scores. It can be seen that Theorem 9.5 is true.

Finally, the discussion in this chapter is based upon CRS frontier. Similar discussion can be obtained for other RTS frontiers. However, the related context-dependent DEA may be infeasible. See Chaps. 7 and 10 for the discussion on infeasibility of DEA-type models. The DEAFrontier software allows you to calculate the context-dependent DEA under non-CRS assumptions.

9.5 Solving DEA Using DEAFrontier Software

The context-dependent DEA consists of three functions: (i) Obtain levels, (ii) Calculate context-depend DEA models, and (iii) Unprotect the sheets containing the levels (see Fig. 9.12).

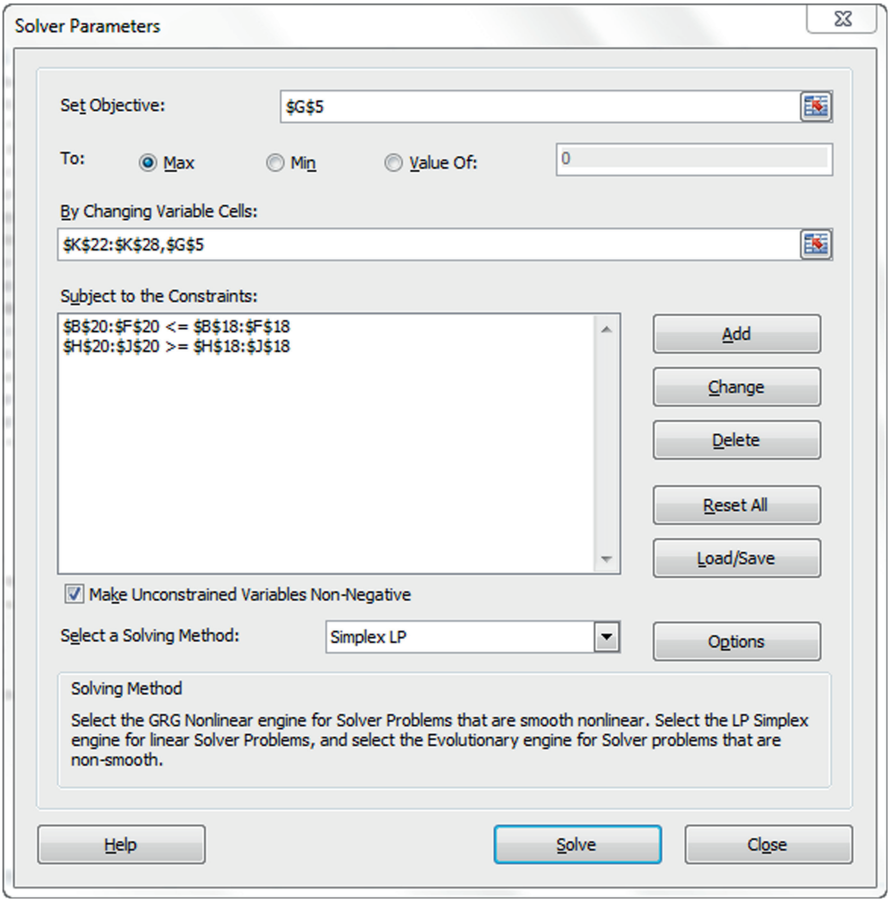
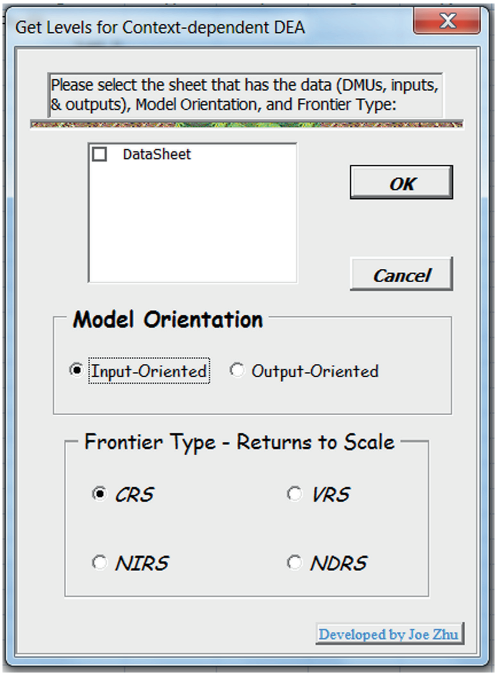


Fig. 9.11 Solver parameters for output-oriented first degree attractiveness

Fig. 9.12 Context-dependent DEA menu

The first function is the stratification model (9.1). It generates all the efficient frontiers—levels (Fig. 9.13). This function will first delete any sheet with a name starting with “Level” and then generate a set of new sheets named as “Level*i*(*Frontier*)” where *i* indicates the level and *Frontier* represents the frontier type. For example, Level1(CRS) means the first level CRS frontier. The “level” sheets are protected for use in the context-dependent DEA. However, they can be unprotected by using the “Unprotect the sheets” menu item. *The format of these level sheets must*

Fig. 9.13 Obtain levels



not be modified. Otherwise, the context-dependent DEA will not run properly and accurately.

Once the efficient frontiers are obtained, the context-dependent DEA can be calculated using the "Context-dependent DEA" submenu item (Fig. 9.14).

The results are reported in the "Context Dependent Result" sheet. In this sheet, the context-dependent scores are the optimal values to model (9.2) (or model (9.7), model (9.10), model (9.11)). To obtain the attractiveness or progress scores, one has to adjust the context-dependent scores based upon Definitions 9.1, 9.2, 9.4, and 9.5.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_9) contains supplementary material, which is available to authorized users.

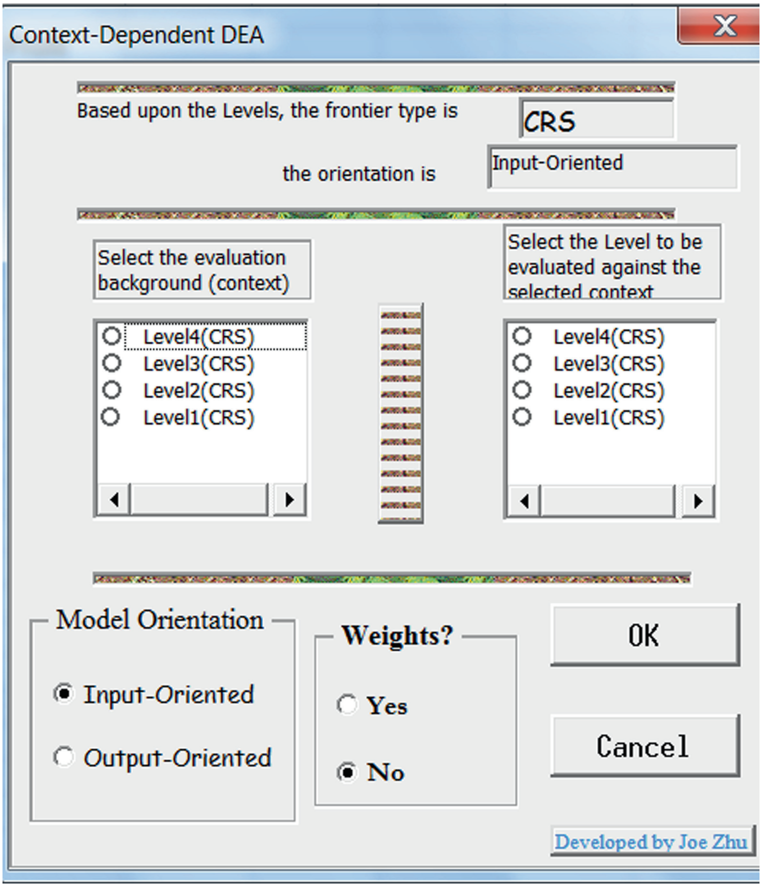


Fig. 9.14 Context-dependent DEA

References

Seiford, L. M., & Zhu, J. (2003). Context-dependent data envelopment analysis: Measuring attractiveness and progress. *OMEGA, 31*(5), 397–480.
Simonson, I., & Tversky, A. (1992). Choice in context: Tradeoff contrast and extremeness aversion. *Journal of Marketing Research, 29,* 281–295.
Tversky, A., & Simonson, I. (1993). Context-dependent preferences. *Management Sciences, 39,* 1179–1189.

Chapter 10
Super Efficiency

10.1 Super-Efficiency DEA Models

When a DMU under evaluation is not included in the reference set of the envelopment models, the resulting DEA models are called super-efficiency DEA models. Charnes et al. (1992) use a super-efficiency model to study the sensitivity of the efficiency classifications. Zhu (1996) and Seiford and Zhu (1998) develop a number of new super-efficiency models to determine the efficiency stability regions (see Chap. 11). Andersen and Petersen (1993) propose using the CRS super-efficiency model in ranking the efficient DMUs. Also, the super-efficiency DEA models can be used in detecting influential observations (Wilson 1995) and in identifying the extreme efficient DMUs (Thrall 1996). Seiford and Zhu (1999) study the infeasibility of various super-efficiency models developed from the envelopment models in Table 11.2. Chapter 11 presents other super-efficiency models that are used in sensitivity analysis.

Table 10.1 presents the basic super-efficiency DEA models based upon the envelopment DEA models. Based upon Table 10.1, we see that the difference between the super-efficiency and the envelopment models is that the DMU_o under evaluation is excluded from the reference set in the super-efficiency models. i.e., the super-efficiency DEA models are based on a reference technology constructed from all other DMUs.

Consider the example in Table 10.1. If we measure the (CRS) super efficiency of DMU2, then DMU2 is evaluated against point A on the new facet determined by DMUs 1 and 3 (see Fig. 10.1). To calculate the (CRS) super efficiency score for DMU2, we use the spreadsheet model shown in Fig. 10.2.

Cell E9 indicates the DMU under evaluation which is excluded from the reference set. Cells F2:F6 are reserved for $\lambda_j\,(j = 1, 2, 3, 4, 5)$,and cell F10 is reserved for the super-efficiency score (θ^{super}).

Cells B11:B13 contain the following formulas

Cell B11 = SUMPRODUCT(B2:B6,F2:F6)
Cell B12 = SUMPRODUCT(C2:C6,F2:F6)
Cell B13 = SUMPRODUCT(E2:E6,F2:F6)

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_10,

Table 10.1 Super-efficiency DEA models

Frontier type	Input-oriented	Output-oriented
CRS	$\min \theta^{\text{super}}$ subject to $\sum_{\substack{j=1\\j\neq o}}^{n} \lambda_j x_{ij} \le \theta^{\text{super}} x_{io} \quad i=1,2,\ldots,m;$ $\sum_{\substack{j=1\\j\neq o}}^{n} \lambda_j y_{rj} \ge y_{ro} \quad r=1,2,\ldots,s;$ $\lambda_j \ge 0 \quad j \neq o.$	$\max \phi^{\text{super}}$ subject to $\sum_{\substack{j=1\\j\neq o}}^{n} \lambda_j x_{ij} \le x_{io} \quad i=1,2,\ldots,m;$ $\sum_{\substack{j=1\\j\neq o}}^{n} \lambda_j y_{rj} \ge \phi^{\text{super}} y_{ro} \quad r=1,2,\ldots,s;$ $\lambda_j \ge 0 \quad j \neq o.$
VRS	Add $\sum_{j\neq o} \lambda_j = 1$	
NIRS	Add $\sum_{j\neq o} \lambda_j \le 1$	
NDRS	Add $\sum_{j\neq o} \lambda_j \ge 1$	

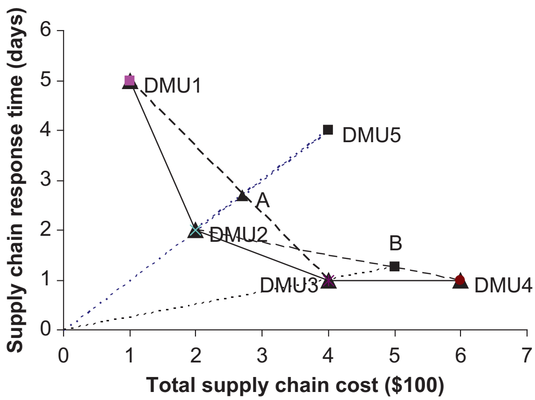


Fig. 10.1 Super-efficiency

Note that in the above formulas, the DMU under evaluation is included in the reference set. In order to exclude the DMU under evaluation from the reference set, we introduce the following formula into cell B14

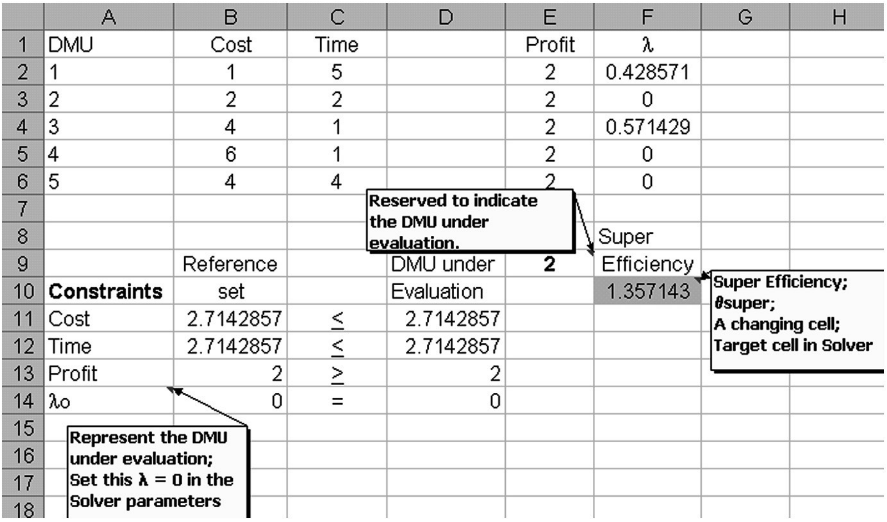
Cell B14 = INDEX(F2:F6,E9,1)

which returns the λ_j for the DMU$_j$ under evaluation. In the Solver parameters shown in Fig. 10.3, we set cell B14 equal to zero.

Cells D11:D13 contain the following formulas

Cell D11 = $F$10*INDEX(B2:B6,E9,1)
Cell D12 = $F$10*INDEX(C2:C6,E9,1)
Cell D13 = INDEX(E2:E6,E9,1)

Based upon Figs. 10.2 and 10.3, the super-efficiency score for DMU2 is 1.357, and the non-zero λ_j in cells F2 and F4 indicate that DMU1 and DMU3 form a new efficient facet.



	A	B	C	D	E	F	G	H
1	DMU	Cost	Time		Profit	λ		
2	1	1	5		2	0.428571		
3	2	2	2		2	0		
4	3	4	1		2	0.571429		
5	4	6	1		2	0		
6	5	4	4		2	0		
7								
8						Super		
9		Reference		DMU under	2	Efficiency		
10	Constraints	set		Evaluation		1.357143		
11	Cost	2.7142857	≤	2.7142857				
12	Time	2.7142857	≤	2.7142857				
13	Profit	2	≥	2				
14	λo	0	=	0				
15								
16								
17								
18								

Fig. 10.2 Input-oriented CRS super-efficiency spreadsheet model

DMU3 is evaluated against B on the new facet determined by DMUs 2 and 4. If we change the value of cell E9 to 3, we obtain the super-efficiency score for DMU3 using the Solver parameters shown in Fig. 10.3. The score is 1.25 (see cell G4 in Fig. 10.4).

If we remove DMU4 or DMU 5 from the reference set, the frontier remains the same. Therefore,the super-efficiency score for DMU4 (DMU5) equals to the input-oriented CRS efficiency score (see Fig. 10.4).

If we measure the super-efficiency of DMU1, DMU1 is evaluated against C on the frontier extended from DMU2 (see Fig. 10.5). It can be seen that C is a weakly efficient DMU in the remaining four DMUs 2, 3, 4 and 5. In fact, we may want to adjust such a super-efficiency score (see Zhu (2001) and Chen and Sherman (2004)).

Although the super-efficiency models can differentiate the performance of the efficient DMUs, the efficient DMUs are not compared to the same "standard". Because the frontier constructed from the remaining DMUs changes for each efficient DMU under evaluation. In fact, the super-efficiency should be regarded as the potential input savings or output surpluses (see Chen 2004, 2005).

10.2 Infeasibility of Super-efficiency DEA Models

Consider the input-oriented VRS super-efficiency model shown in Fig. 10.6. In fact, this is the spreadsheet model for the input-oriented VRS envelopment model except that we introduce the formula "=INDEX(I2:I16, E18,1)" into cell B26. This formula

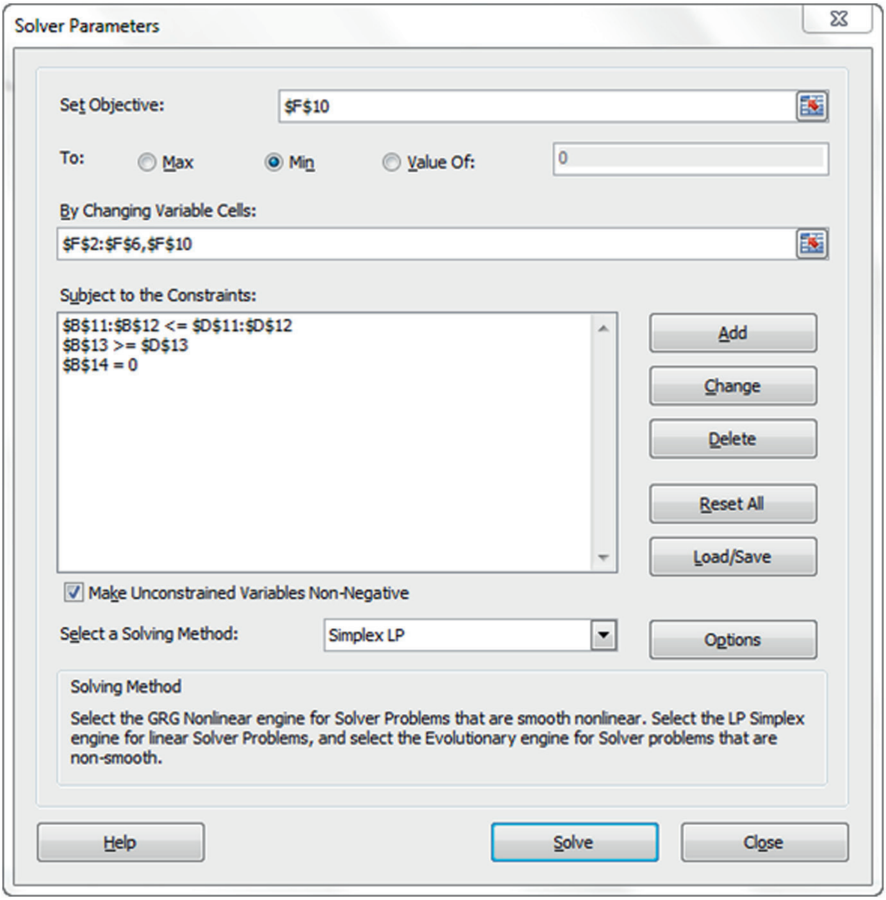
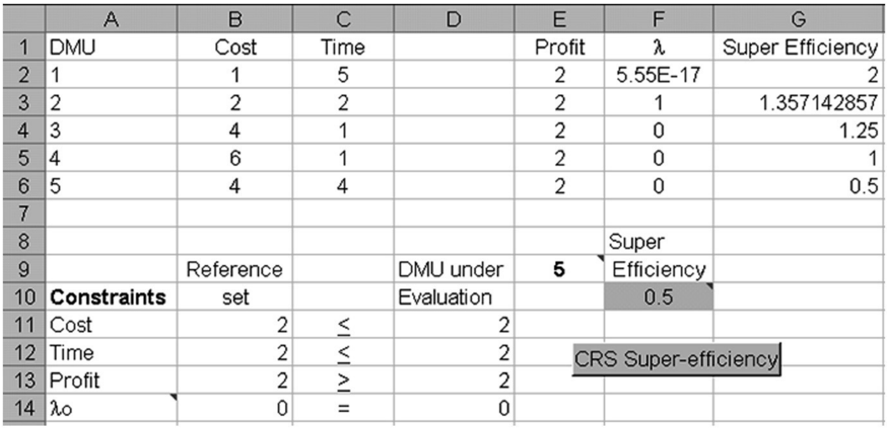


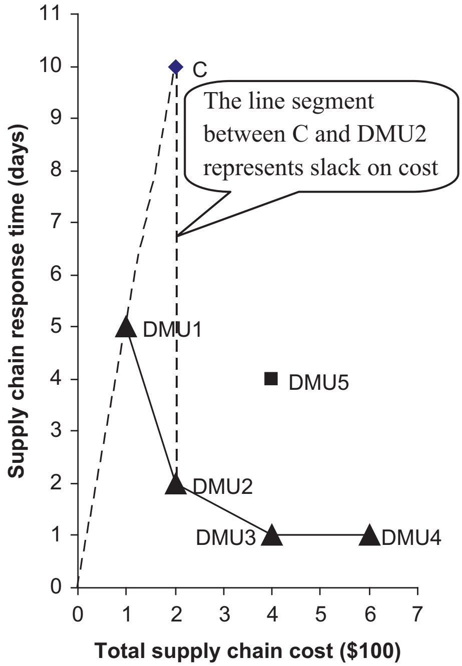
Fig. 10.3 Solver parameters for input-oriented CRS super-efficiency



	A	B	C	D	E	F	G
1	DMU	Cost	Time		Profit	λ	Super Efficiency
2	1	1	5		2	5.55E-17	2
3	2	2	2		2	1	1.357142857
4	3	4	1		2	0	1.25
5	4	6	1		2	0	1
6	5	4	4		2	0	0.5
7							
8						Super	
9		Reference		DMU under	**5**	Efficiency	
10	**Constraints**	set		Evaluation		0.5	
11	Cost	2	≤	2			
12	Time	2	≤	2		CRS Super-efficiency	
13	Profit	2	≥	2			
14	λo	0	=	0			

Fig. 10.4 Super-efficiency scores

Fig. 10.5 Super-efficiency and slacks



	A	B	C	D	E	F	G	H	I	J
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	Super efficiency
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		0	Infeasible
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0	1.751885253
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	1.606521649
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0	Infeasible
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0.77	1.320957592
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	1.009347188
8	Ford Motor	243283	24547	346990		137137	4139		0	0.737555958
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	0.603245345
10	Exxon	91296	40436	82000		110009	6470		0	1.344368672
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	Infeasible
12	Wal-Mart	37871	14762	675000		93627	2740		0.23	1.765155063
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	0.557595838
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0	4.806917693
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0	0.470610997
16	AT&T	88884	17274	299300		79609	139		0	0.533543522
17						Super				
18		Reference		DMU under	15	Efficiency				
19	**Constraints**	set		Evaluation		0.5335435				
20	Assets	47423.482	≤	47423.4824						
21	Equity	8535.6544	≤	9216.4308						
22	Employees	159689.58	≤	159689.576		VRS Super-efficiency				
23	Revenue	150569.21	≥	79609						
24	Profit	791.04056	≥	139						
25	Σλ	1	=	1						
26	λo	0	=	0						

Fig. 10.6 Input-oriented VRS super-efficiency spreadsheet model

is used to exclude the DMU under evaluation from the reference set. That is, one needs to add an additional constraint of “$B$26=0” into the Solver parameters for the input-oriented VRS envelopment spreadsheet model, as shown in Fig. 10.7.

Once we set up the Solver parameters, the calculation is performed by the VBA procedure “SuperEfficiency”.

```
Sub SuperEfficiency()
Dim i As Integer
For i = 1 To 15
Range("E18") = i
SolverSolve UserFinish:=True
If SolverSolve(UserFinish:=True) = 5 Then
Range("J" & i + 1) = "Infeasible"
Else
Range("J" & i + 1) = Range("F19")
End If
Next
End Sub
```

It can be seen that the input-oriented VRS super-efficiency model is infeasible for three VRS efficient companies (Mitsubishi, General Motors, and Royal Dutch/Shell Group). Note that in the VBA procedure “SuperEfficiency”, a VBA statement on infeasibility check is added.

If we consider the output-oriented VRS super-efficiency model, we have the spreadsheet shown in Fig. 10.8. Figure 10.8 is based upon the output-oriented VRS envelopment with an additional formula in cell B26 “=INDEX (I2:I16,E18,1)”. To calculate the output-oriented super-efficiency scores, we need to change the “Min” to “Max” in the Solver parameters shown in Fig. 10.7.

Based upon Fig. 10.8, the output-oriented VRS super-efficiency model is infeasible for five output-oriented VRS efficient companies (Itochu, Sumitomo, Marubeni, Wal-Mart, and Nippon Life Insurance).

Thrall (1996) shows that the super-efficiency CRS model can be infeasible. However, Thrall (1996) fails to recognize that the output-oriented CRS super-efficiency model is always feasible for the trivial solution which has all variables set equal to zero. Moreover, Zhu (1996) shows that the input-oriented CRS super-efficiency model is infeasible if and only if a certain pattern of zero data occurs in the inputs and outputs.

Figure 10.9 illustrates how the VRS super-efficiency model works and the infeasibility for the case of a single output and a single input case. We have three VRS frontier DMUs, A, B and C. AB exhibits IRS and BC exhibits DRS. The VRS super-efficiency model evaluates point B by reference to B′ and B″ on section AC through output-reduction and input-increment, respectively. In an input-oriented VRS super-efficiency model, point A is evaluated against A′. However, there is no referent DMU for point C for input variations. Therefore, the input-oriented VRS

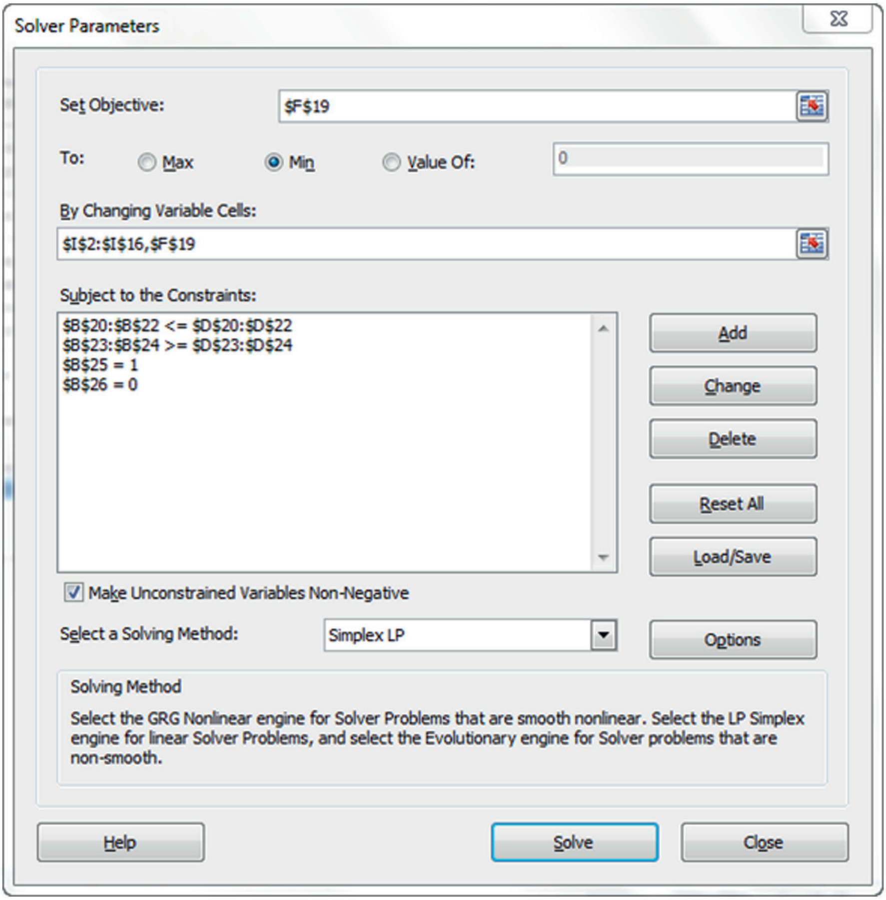


Fig. 10.7 Solver parameters for input-oriented VRS super-efficiency

super-efficiency model is infeasible at point C. Similarly, in an output-oriented VRS super-efficiency model, point C is evaluated against C′. However, there is no referent DMU for point A for output variations. Therefore, the output-oriented VRS super-efficiency model is infeasible at point A. Note that point A is the left most end point and point CB is the right most end point on this frontier.

As in Charnes et al. (1991), the DMUs can be partitioned into four classes E, E′, F and N described as follows. First, E is the set of extreme efficient DMUs. Second, E′ is the set of efficient DMUs that are not extreme points. The DMUs in set E′ can be expressed as linear combinations of the DMUs in set E. Third, F is the set of frontier points (DMUs) with non-zero slack(s). The DMUs in set F are usually called weakly efficient. Fourth, N is the set of inefficient DMUs.

For example, DMUs 1, 2, and 3 in Fig. 10.1 are extreme efficient (in set E), DMU4 is in set F, and DMU5 is in set N.

	A	B	C	D	E	F	G	H	I	J
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	Super efficiency
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		0.869	0.936120353
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0.131	0.937264375
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	Infeasible
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0	0.647119111
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0	Infeasible
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	Infeasible
8	Ford Motor	243283	24547	346990		137137	4139		0	1.158414974
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	1.371588284
10	Exxon	91296	40436	82000		110009	6470		0	0.673147631
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	0.939143546
12	Wal-Mart	37871	14762	675000		93627	2740		0	Infeasible
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	1.898938938
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0	Infeasible
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0	1.892916538
16	AT&T	88884	17274	299300		79609	139		0	2.311193684
17						Super				
18		Reference		DMU under	15	Efficiency				
19	**Constraints**	set		Evaluation		2.3111937				
20	Assets	88884	≤	88884						
21	Equity	10242.181	≤	17274						
22	Employees	41771.582	≤	299300						
23	Revenue	183991.82	≥	183991.818						
24	Profit	342.08119	≥	321.255922						
25	Σλ	1	=	1						
26	λo	0	=	0						

Output-oriented
VRS Super-efficiency

Fig. 10.8 Output-oriented VRS super-efficiency spreadsheet model

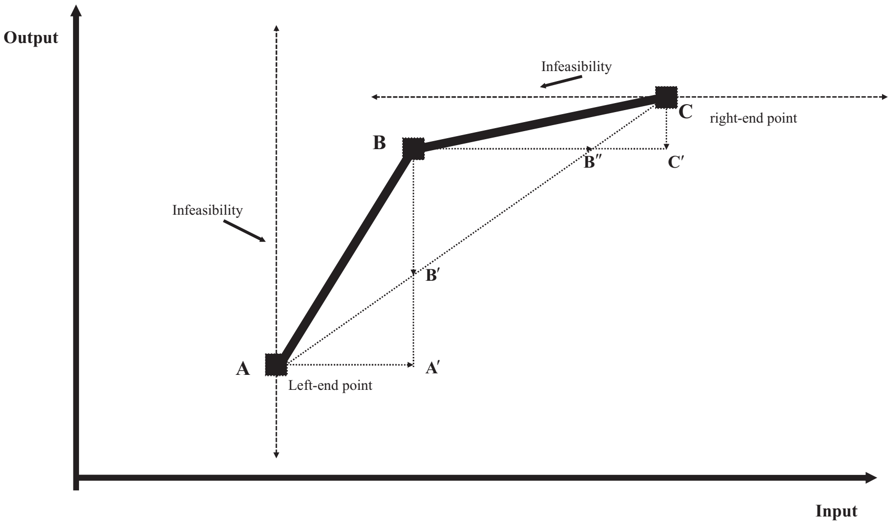


Fig. 10.9 Infeasibility of super-efficiency model

Thrall (1996) shows that if the CRS super-efficiency model is infeasible, or if the super-efficiency score is greater than one for input-oriented model (less than one for output-oriented model), then $DMU_o \in E$. This result can also be applied to other super-efficiency models. i.e., the extreme efficient DMUs can be identified by the super-efficiency models. This finding is important in empirical applications. For example, in the slack-based congestion measures discussed in Chap. 9, if we can know that the data set consists of only extreme efficient DMUs, then the congestion slacks are equal to the DEA slacks.

Note that if a specific $DMU_o \in$ E', F or N and is not included in the reference set, then the efficient frontiers (constructed by the DMUs in set E) remain unchanged. As a result, the super-efficiency DEA models are always feasible and equivalent to the original DEA models when $DMU_o \in$ E', F or N. Thus we only need to consider the infeasibility when $DMU_o \in \mathrm{E}$.

We next study the infeasibility of the VRS, NIRS and NDRS super-efficiency models, where we assume that all data are positive.

From the convexity constraint $\left(\sum_{j\neq o} \lambda_j = 1\right)$ on the intensity lambda variables, we immediately have

Proposition 10.1 $DMU_o \in \mathrm{E}$ under the VRS model *if and only if* $DMU_o \in \mathrm{E}$ under the NIRS model or NDRS model.

Thus in the discussion to follow, we limit our consideration to $DMU_o \in \mathrm{E}$ under the VRS model. We have

Proposition 10.2 Let $\theta^{\text{super}*}$ and $\phi^{\text{super}*}$ denote, respectively, optimal values to the input-oriented and output-oriented super-efficiency DEA models when evaluating an extreme efficient DMU_o, then

(i) Either $\theta^{\text{super}*} > 1$ or the specific input-oriented super-efficiency DEA model is infeasible.

(ii) Either $\phi^{\text{super}*} < 1$ or the specific output-oriented super-efficiency DEA model is infeasible.

Based upon Seiford and Zhu (1999), we next (i) present the necessary and sufficient conditions for the infeasibility of various super-efficiency DEA models in a multiple inputs and multiple outputs situation, and (ii) reveal the relationship between infeasibility and RTS classification. (Note that, in Fig. 10.9, point A is associated with IRS and point C is associated with DRS.)

10.2.1 Output-Oriented VRS Super-Efficiency Model

Suppose each $DMU_j\,(j = 1,2,\ldots,n)$ consumes a vector of inputs, x_j, to produce a vector of outputs, y_j. We have

Theorem 10.1 For a specific extreme efficient $DMU_o = (x_o, y_o)$, the output-oriented VRS super-efficiency model is infeasible *if and only if* $(x_o, \delta y_o)$ is efficient under the VRS envelopment model for any $0 < \delta \leq 1$.

[Proof]: Suppose that the output-oriented VRS super-efficiency model is infeasible and that $(x_o, \delta^o y_o)$ is inefficient, where $0 < \delta^o \leq 1$. Then

$$\begin{aligned}
&\phi_o^{\text{super}*} = \max \ \phi_o^{\text{super}} \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_j \leq x_o \\
&\sum_{j=1}^{n} \lambda_j y_j \geq \phi_o^{\text{super}} (\delta^o y_o) \\
&\sum_{j=1}^{n} \lambda_j = 1
\end{aligned} \tag{10.1}$$

has a solution of $\lambda_j^*(j \neq o)$, $\lambda_o^* = 0$, $\phi_o^{\text{super}*} > 1$. Since $\lambda_o^* = 0$, we have that model (10.1) is equivalent to an output-oriented VRS super-efficiency model and thus the output-oriented VRS super-efficiency model is feasible. A contradiction. This completes the proof of the *only if* part.

To establish the *if* part, we note that if the output-oriented VRS super-efficiency model is feasible, then $\phi^{\text{super}*} < 1$ is the maximum radial reduction of all outputs preserving the efficiency of DMU_o. Therefore, δ cannot be less than $\phi^{\text{super}*}$. Otherwise, DMU_o will be inefficient under the output-oriented VRS envelopment model. Thus, the output-oriented VRS super-efficiency model is infeasible.■

Theorem 10.2 The output-oriented VRS super-efficiency model is infeasible *if and only if* $\hbar^*$, where $\hbar^* > 1$ is the optimal value to (10.2).

$$\begin{aligned}
&\hbar^* = \min \ \hbar \\
&\text{subject to} \\
&\sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j x_j \leq \hbar x_o \\
&\sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j = 1 \\
&\lambda_j \geq 0, j \neq o
\end{aligned} \tag{10.2}$$

[Proof]: We note that for any $\lambda_j (j \neq o)$ with $\Sigma_{j \neq o} \lambda_j = 1$, the constraint $\sum_{j \neq o} \lambda_j y_j \geq \phi^{\text{super}} y_o$ always holds. Thus the output-oriented super-efficiency-VRS is infeasible if and only if there exists no $\lambda_j (j \neq o)$ with $\Sigma_{j \neq o} \lambda_j = 1$ such that

	A	B	C	D	E	F	G	H	I
1	Company	Assets	Equity	Employees			λ	h	Super efficiency
2	Mitsubishi	91920.6	10950	36000			0	0.57426	0.936120353
3	Mitsui	68770.9	5553.9	80000			0	0.8919	0.937264375
4	Itochu	65708.9	4271.1	7182			0	1.20984	Infeasible
5	General Motors	217123.4	23345.5	709000			0	0.25382	0.647119111
6	Sumitomo	50268.9	6681	6193			0.77	1.30639	Infeasible
7	Marubeni	71439.3	5239.1	6702			0	1.00935	Infeasible
8	Ford Motor	243283	24547	346990			0	0.23236	1.158414974
9	Toyota Motor	106004.2	49691.6	146855			0	0.4634	1.371588284
10	Exxon	91296	40436	82000			0	0.54283	0.673147631
11	Royal Dutch/Shell Group	118011.6	58986.4	104000			0	0.42008	0.939143546
12	Wal-Mart	37871	14762	675000			0.23	1.32737	Infeasible
13	Hitachi	91620.9	29907.2	331852			0	0.51532	1.898938938
14	Nippon Life Insurance	364762.5	2241.9	89690			0	1.90513	Infeasible
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400			0	0.38353	1.892916538
16	AT&T	88884	17274	299300			0	0.53354	2.311193684
17									
18		Reference		DMU under	15	h			
19	**Constraints**	set		Evaluation		0.533544			
20	Assets	47423.482	≤	47423.482					
21	Equity	8535.6544	≤	9216.4308		Infeasibility			
22	Employees	159689.58	≤	159689.58					
23	Σλ	1	=	1					
24	λo	0	=	0					

Fig. 10.10 Spreadsheet for infeasibility test (Output-oriented VRS super-efficiency)

$\sum_{j \neq o} \lambda_j x_j \le x_o$ holds. This means that the optimal value to (10.2) is greater than one, i.e., $\hbar^* > 1$.■

Figure 10.10 shows the spreadsheet model for model (10.2) where the output-oriented VRS super-efficiency scores are reported in cells I2:I16.

The spreadsheet shown in Fig. 10.10 is obtained by removing the output constraints from the spreadsheet shown in Fig. 10.6. Figure 10.11 shows the Solver parameters. It can be seen that $\hbar^* > 1$ if and only if model (10.2) is infeasible for a company.

Further, note that the DMU_o is also CRS efficient if and only if CRS prevail. Therefore, if IRS or DRS prevail, then DMU_o must be CRS inefficient. Thus, in this situation, the CRS super-efficiency model is identical to the CRS envelopment model. Based upon Chap. 13, IRS or DRS on DMU_o can be determined by

Lemma 10.1 The RTS for DMU_o can be identified as IRS *if and only if* $\sum_{j \neq 0} \lambda_j^* < 1$ in all optima for the CRS super-efficiency model and DRS *if and only if* $\sum_{j \neq 0} \lambda_j^*$ in all optima for the CRS super-efficiencymodel.

Lemma 10.2 If DMU_o exhibits DRS, then the output-oriented VRS super-efficiency model is feasible and $\phi^{\text{super}*} < 1$, where $\phi^{\text{super}*}$ is the optimal value to the output-oriented VRS super-efficiency model.

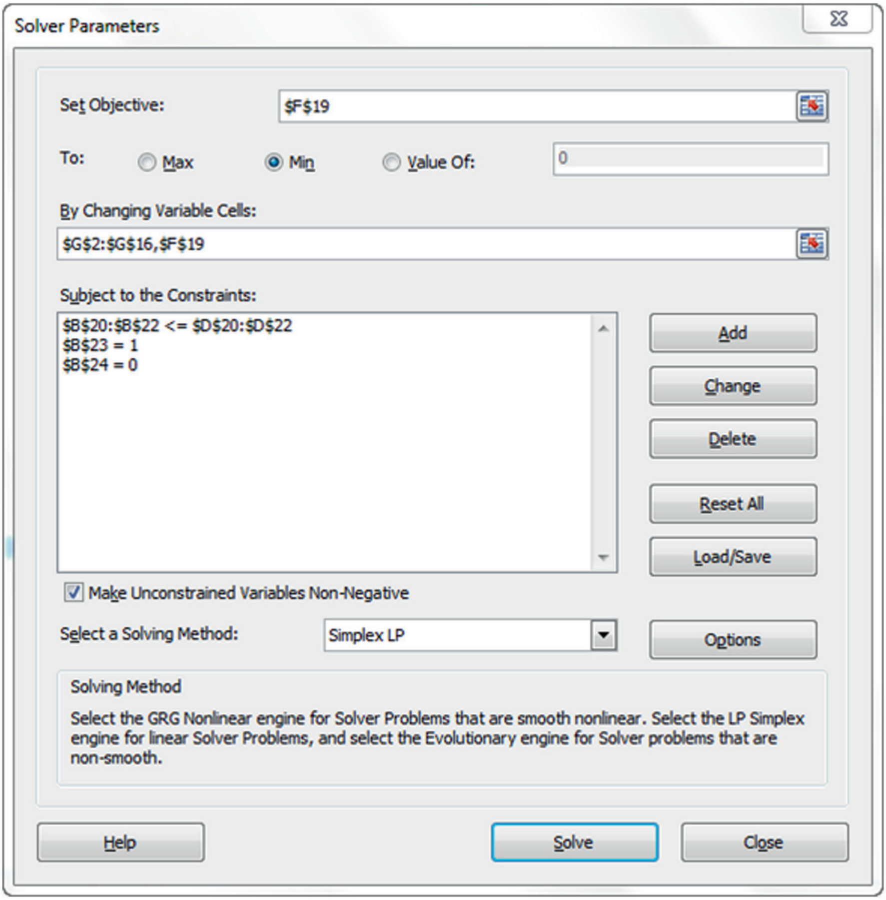


Fig. 10.11 Solver parameters for infeasibility test (Output-oriented)

[Proof]: The output-oriented VRS super-efficiency model is as follows

$$
\begin{aligned}
&\phi^{\text{super}*} = \max \phi^{\text{super}} \\
&\text{subject to} \\
&\sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j x_j \leq x_o \\
&\sum_{\substack{j=1 \\ j \neq 0}}^{n} \lambda_j y_j \geq \phi^{\text{super}} y_o \\
&\sum_{\substack{j=1 \\ j \neq 0}}^{n} \lambda_j = 1; \\
&\phi^{\text{super}}, \lambda_j \geq 0, j \neq o
\end{aligned}
\tag{10.3}
$$

Let $\theta = 1/\phi^{\text{super}}$. Multiplying all constraints in (10.3) by θ yields

$$\begin{aligned}
&\min \theta \\
&\text{subject to} \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \tilde{\lambda}_j x_j \le \theta x_o \\
&\sum_{\substack{j=1 \\ j\neq 0}}^{n} \tilde{\lambda}_j y_j \ge y_o \\
&\sum_{\substack{j=1 \\ j\neq 0}}^{n} \tilde{\lambda}_j = \theta = \frac{1}{\phi^{\text{super}}} \\
&\phi^{\text{super}}, \theta, \tilde{\lambda}_j \ge 0, j \neq o
\end{aligned} \tag{10.4}$$

where $\tilde{\lambda}_j = \theta\lambda_j \ (j \neq o)$.

Since DMU_o exhibits DRS, then by Lemma 10.1, $\sum_{j\neq 0} \lambda_j^* > 1$ in all optima to the following CRS super-efficiency model

$$\begin{aligned}
&\min \theta^{\text{super}} \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j x_j \le \theta^{\text{super}} x_o \\
&\sum_{\substack{j=1 \\ j\neq 0}}^{n} \lambda_j y_j \ge y_o \\
&\lambda_j \ge 0.
\end{aligned} \tag{10.5}$$

Let $\sum_{j\neq 0} \lambda_j^* = \theta$. Obviously $\theta > \theta^{\text{super}}$ is a feasible solution to (10.5). This in turn indicates that $\lambda_j^* \ (j \neq o)$ and θ is a feasible solution to (10.4). Therefore, (10.3) is feasible. Furthermore by Proposition 10.2, we have that $\phi^{\text{super}*} < 1$, where $\phi^{\text{super}*}$ is the optimal value to (10.3).

Theorem 10.3 If the output-oriented VRS super-efficiency model is infeasible, then DMU_o exhibits IRS or CRS.

[Proof]: Suppose that DMU_o exhibits DRS. By Lemma 10.2, the output-oriented VRS super-efficiency modelis feasible. A contradiction.■

Theorems 10.1 and 10.2 indicate that if the output-oriented VRS super-efficiency model is infeasible, then DMU_o is one of the *endpoints*. Moreover, if IRS prevail, then DMU_o is a *left endpoint* (see Fig. 10.9).

10.2.2 Other Output-oriented Super-efficiency Models

Now, consider the output-oriented NIRS and NDRS super-efficiency models. Obviously, we have a feasible solution of $\lambda_j = 0 \ (j \neq o)$ and $\phi^{\text{super}} = 0$ in the output-oriented NIRS super-efficiency model. Therefore, we have

Theorem 10.4 The output-oriented NIRS super-efficiency model is always feasible.

Lemma 10.3 The output-oriented NDRS super-efficiency model is infeasible *if and only if* the output-oriented VRS super-efficiency model is infeasible.

[Proof]: The *only if* part is obvious and hence is omitted. To establish the *if* part, we suppose that the output-oriented NDRS super-efficiency model is feasible. i.e., we have a feasible solution with $\sum_{j\neq o}\lambda_j \geq 1$ for the output-oriented NDRS super-efficiency model. If $\sum_{j\neq o}\lambda_j = 1$, then this solution is also feasible for the output-oriented VRS super-efficiency. If $\sum_{j\neq o}\lambda_j > 1$, let $\sum_{j\neq o}\lambda_j = d > 1$. Then $\sum_{j\neq o}\tilde{\lambda}_j x_j \leq \sum_{j\neq o}\lambda_j x \leq x_o$, where $\tilde{\lambda}_j = \lambda_j / d (j \neq o)$ and $\sum_{j\neq o}\lambda_j = 1$. Therefore $\tilde{\lambda}_j \ (j \neq o)$ is a feasible solution to the output-oriented VRS super-efficiency model. Both possible cases lead to a contradiction. Thus, the output-oriented NDRS super-efficiency model is infeasible if the output-oriented VRS super-efficiency model is infeasible.

On the basis of Lemma 10.3, we have

Theorem 10.5 For a specific extreme efficient $DMU_o = (x_o, y_o)$, we have

(i) The output-oriented NDRS super-efficiency model is infeasible *if and only if* $(x_o, \delta y_o)$ is efficient under the VRS envelopment model for any $0 < \delta \leq 1$.

(ii) The output-oriented NDRS super-efficiency model is infeasible *if and only if* $\hbar^* > 1$, where $\hbar^*$ is the optimal value to (10.2).

If $DMU_o \in \text{E}$ for the NDRS model, then DMU_o exhibits IRS or CRS. By Proposition 10.1, DMU_o also lies on the VRS frontier that satisfies IRS or CRS. i.e., the VRS and NDRS envelopment models are identical for DMU_o. Thus, $(x_o, \delta y_o)$ is also efficient under the NDRS envelopment model for any $0 < \delta \leq 1$.

10.2.3 Input-Oriented VRS Super-Efficiency Model

Theorem 10.6 For a specific extreme efficient $DMU_o = (x_o, y_o)$, the input-oriented VRS super-efficiency model is infeasible *if and only if* $(\chi x_o, y_o)$ is efficient under the VRS envelopment model for any $1 \leq \chi < +\infty$.

[Proof]: Suppose the input-oriented VRS super-efficiency model is infeasible and assume that $(\chi^o x_o, y_o)$ is inefficient, where $1 \leq \chi^o < +\infty$. Then

$$\theta_o^{\text{super}*} = \min \ \theta_o^{\text{super}}$$
subject to
$$\sum_{j=1}^{n} \lambda_j x_j \le \theta_o^{\text{super}} (\chi^o x_o)$$
$$\sum_{j=1}^{n} \lambda_j y_j \ge y_o \tag{10.6}$$
$$\sum_{j=1}^{n} \lambda_j = 1$$

has a solution of $\lambda_j^*(j \ne o)$, $\lambda_o^* = 0$, $\theta_o^{\text{super}*}$. Since $\lambda_o^* = 0$, model (10.6) is equivalent to the input-oriented VRS super-efficiency model. Thus, the input-oriented VRS super-efficiency model is feasible. This completes the proof of *only if* part.

To establish the *if* part, we note that if the input-oriented VRS super-efficiency model is feasible, then $\theta^{\text{super}*} > 1$ is the maximum radial increase of all inputs preserving the efficiency of DMU_o. Therefore, χ cannot be bigger than $\theta^{\text{super}*}$. Otherwise, DMU_o will be inefficient under the input-oriented VRS envelopment model. Thus, the input-oriented VRS super-efficiency model is infeasible.■

Theorem 10.7 The input-oriented super-efficiency-VRS model is infeasible *if and only if* $g^* < 1$, where g^* is the optimal value to (10.7).

$$g^* = \max \ g$$
subject to
$$\sum_{\substack{j=1 \\ j \ne o}}^{n} \lambda_j y_j \ge g y_o \tag{10.7}$$
$$\sum_{\substack{j=1 \\ j \ne o}}^{n} \lambda_j = 1$$
$$\lambda_j \ge 0, j \ne o$$

[Proof]: We note that for any $\lambda_j (j \ne o)$ with $\sum_{j \ne o} \lambda_j = 1$, the constraint $\sum_{j \ne o} \lambda_j x_j \le \theta^{\text{super}} x_o$ always holds. Thus, the input-oriented VRS super-efficiency model is infeasible if and only if $\sum_{j \ne o} \lambda_j y_j \ge y_o$ does not hold for any λ_j $(j \ne o)$ with $\sum_{j \ne o} \lambda_j = 1$. This means that the optimal value to (10.7) is less than one, i.e., $g^* < 1$.■

Figure 10.12 shows the spreadsheet model for model (10.7) where the input-oriented VRS super-efficiency scores are reported in cells I2:I16. This spreadsheet is obtained from the output-oriented VRS super-efficiency model shown in Fig. 10.8. Figure 10.13 shows the Solver parameters. It can be seen that $g^* < 1$ if and only if model (10.7) is infeasible for a company.

	A	B	C	D	E	F	G	H	I
1	Company	Revenue	Profit	λ				g	Super efficiency
2	Mitsubishi	184365.2	346.2	1				0.9843	Infeasible
3	Mitsui	181518.7	314.8	0				1.0157	1.751885253
4	Itochu	169164.6	121.2	0				1.0899	1.606521649
5	General Motors	168828.6	6880.7	0				0.7623	Infeasible
6	Sumitomo	167530.7	210.5	0				1.1005	1.320957592
7	Marubeni	161057.4	156.6	0				1.1447	1.009347188
8	Ford Motor	137137	4139	0				1.26	0.737555958
9	Toyota Motor	111052	2662.4	0				1.5777	0.603245345
10	Exxon	110009	6470	0				1.0667	1.344368672
11	Royal Dutch/Shell Group	109833.7	6904.6	0				0.9965	Infeasible
12	Wal-Mart	93627	2740	0				1.8493	1.765155063
13	Hitachi	84167.1	1468.8	0				2.1126	0.557595838
14	Nippon Life Insurance	83206.7	2426.6	0				2.0813	4.806917693
15	Nippon Telegraph & Telephone	81937.2	2209.1	0				2.124	0.470610997
16	AT&T	79609	139	0				2.3159	0.533543522
17						Super			
18		Reference		DMU under	15	Efficiency			
19	**Constraints**	set		Evaluation		2.315884			
20	Revenue	184365.2	≥	184365.2					
21	Profit	346.2	≥	321.90786		Infeasibility			
22	Σλ	1	=	1					
23	λo	0	=	0					

Fig. 10.12 Spreadsheet for infeasibility test (Input-oriented VRS super-efficiency)

Lemma 10.4 If DMU_o exhibits IRS, then the input-oriented VRS super-efficiency model is feasible and $\theta^{\text{super}*} > 1$, where $\theta^{\text{super}*}$ is the optimal value to the input-oriented VRS super-efficiency model.

[Proof]: Let $\vartheta = 1/\theta^{\text{super}}$, then the input-oriented VRS super-efficiency model becomes

$$\begin{aligned}
&\max \vartheta \\
&\text{subject to} \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \hat{\lambda}_j x_j \le x_o; \\
&\sum_{\substack{j=1 \\ j\neq 0}}^{n} \hat{\lambda}_j y_j \ge \vartheta y_o; \\
&\sum_{\substack{j=1 \\ j\neq 0}}^{n} \hat{\lambda}_j = \vartheta = \frac{1}{\theta^{\text{super}}}; \\
&\theta^{\text{super}}, \vartheta, \hat{\lambda}_j \ge 0.
\end{aligned} \tag{10.8}$$

where $\hat{\lambda}_j = \vartheta \lambda_j \; (j \neq o)$.

Since DMU_o exhibits IRS, then by Lemma 10.1, $\sum_{j\neq 0} \lambda_j^* < 1$ in all optima to the following output-oriented CRS super-efficiency model

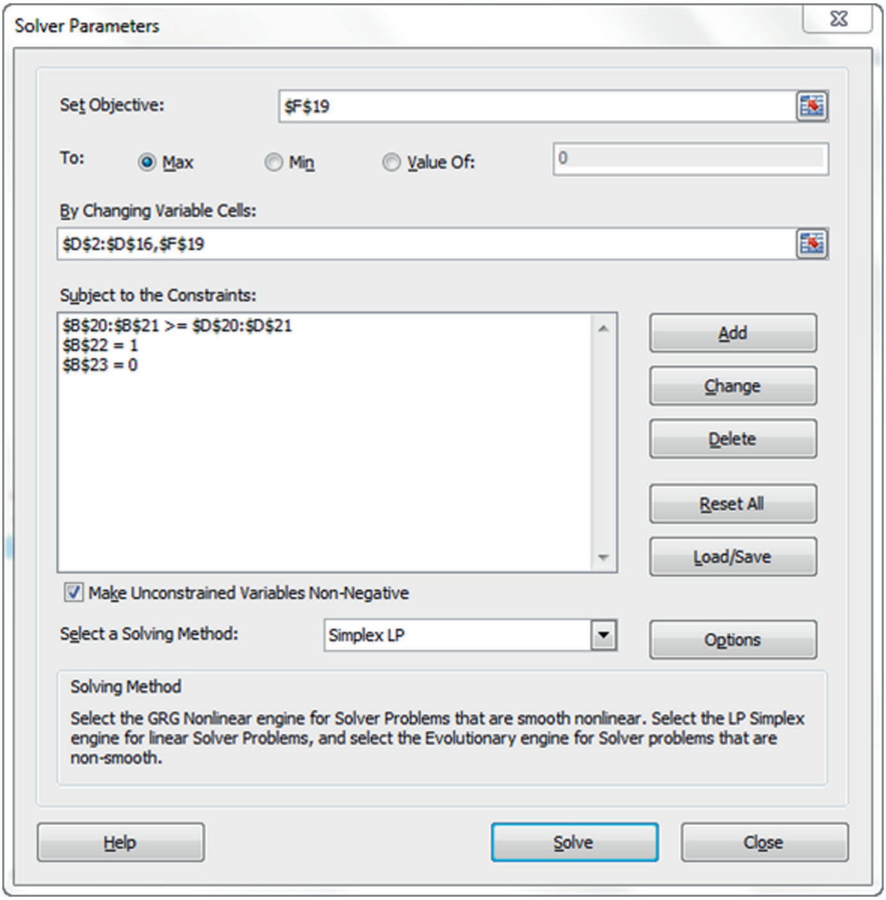


Fig. 10.13 Solver parameters for infeasibility test (Input-oriented)

$$
\begin{aligned}
&\max \phi^{\text{super}} \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j x_j \le x_o \\
&\sum_{\substack{j=1 \\ j\neq 0}}^{n} \lambda_j y_j \ge \phi^{\text{super}} y_o \\
&\phi^{\text{super}}, \lambda_j \ge 0.
\end{aligned}
\tag{10.9}
$$

Let $\sum_{j\neq 0} \lambda_j^* = \vartheta < 1$. Since DMU_o is CRS inefficient, therefore $\phi^{\text{super}} > 1$ and hence $\phi^{\text{super}} > \vartheta$ is a feasible solution to (10.9). This in turn indicates that ϑ and $\lambda_j^*(j \neq o)$ with $\sum_{j\neq 0} \lambda_j^* = \vartheta$ is a feasible solution to (10.8). Therefore, the input-oriented

VRS super-efficiency model is feasible. Furthermore, by Proposition 10.2, we have that $\phi^{\text{super*}} > 1$, where $\phi^{\text{super*}}$ is the optimal value to the input-oriented VRS super-efficiency model.■

Theorem 10.8 If the input-oriented VRS super-efficiency model is infeasible, then DMU_o exhibits DRS or CRS.

[Proof]: If DMU_o exhibits IRS, then by Lemma 10.4, the input-oriented VRS super-efficiency model is feasible. A contradiction.■

Theorems 10.6 and 10.7 indicate that if the input-oriented VRS super-efficiency model is infeasible, then DMU_o is one of the *endpoints*. Furthermore, if DRS prevail, then DMU_o is an *right endpoint* (see Fig. 10.9).

10.2.4 Other Input-oriented Super-efficiency Models

Now, consider the input-oriented NIRS and NDRS super-efficiency models.

Theorem 10.9 The input-oriented NDRS super-efficiency model is always feasible.

[Proof]: Since $\sum_{j \neq o} \lambda_j \geq 1$ in the input-oriented DNRS super-efficiency model, there must exist some $\tilde{\lambda}_j$ with $\sum_{j \neq o} \tilde{\lambda}_j > 1$ such that $\sum_{j \neq o} \tilde{\lambda}_j y_j \geq y_o$ holds. Note that $\sum_{j \neq o} \tilde{\lambda}_j x_j \leq \theta^{\text{super}} x_o$ can always be satisfied by a proper θ^{super}. Thus, the input-oriented NDRS super-efficiency model is always feasible.■

Lemma 10.5 The input-oriented NIRS super-efficiency model is infeasible *if and only if* the input-oriented VRS super-efficiency model is infeasible.

[Proof]: The *only if* part is obvious and hence is omitted. To establish the *if* part, we suppose that the input-oriented NIRS super-efficiency model is feasible. i.e., we have a feasible solution with $\sum_{j \neq o} \lambda_j \leq 1$ for the input-oriented NIRS super-efficiency model. If $\sum_{j \neq o} \lambda_j = 1$, then this solution is also feasible for the output-oriented VRS super-efficiency model. If $\sum_{j \neq o} \lambda_j < 1$, let $\sum_{j \neq o} \lambda_j = e < 1$. Then $\sum_{j \neq o} \hat{\lambda}_j y_j \geq \sum_{j \neq o} \lambda_j y_j \geq y_o$, where $\hat{\lambda}_j = \lambda_j / e (j \neq o)$ and $\sum_{j \neq o} \hat{\lambda}_j = 1$. Therefore $\hat{\lambda}_j (j \neq o)$ is a feasible solution to the output-oriented VRS super-efficiency model. Both possible cases lead to a contradiction. Thus, the output-oriented NIRS super-efficiency model is infeasible if the output-oriented VRS super-efficiency model is infeasible.■

On the basis of this Lemma 10.5, we have

Theorem 10.10 For a specific extreme efficient $DMU_o = (x_o, y_o)$, we have

(i) The input-oriented NIRS super-efficiency model is infeasible *if and only if* $(\chi x_o, y_o)$ is efficient under the VRS envelopment model for any $1 \leq \chi < +\infty$.

(ii) The input-oriented NIRS super-efficiency model is feasible *if and only if* $g^* < 1$, where g^* is the optimal value to (10.7).

Table 10.2 Super-efficiency DEA models and infeasibility

Super-efficiency models		Infeasibility	RTS
Output-oriented	VRS	Theorem 10.2 (Model (10.2))	DRS
	NIRS	Always feasible	Always feasible
Input-oriented	NDRS	Lemma 10.3, Theorem 10.2	Corollary 10.1 (i)
	VRS	Theorem 10.7 (Model (10.7))	IRS
	NIRS	Lemma 10.5, Theorem 7	Always feasible
	NDRS	Always feasible	Corollary 10.1 (ii)

If $DMU_o \in E$ under the NIRS model, then DMU_o exhibits DRS or CRS. By Proposition 10.1, the DMU_o also lies on the VRS frontier that satisfies DRS or CRS. i.e., the VRS and NIRS envelopment models are identical for DMU_o. Thus $(\chi x_o, y_o)$ is also efficient under the NIRS envelopment model for any $1 \le \chi < +\infty$.

Furthermore, Theorems 10.3 and 10.8 demonstrate that the possible infeasibility of the output-oriented and input-oriented VRS super-efficiency models can only occur at those extreme efficient DMUs exhibiting IRS (or CRS) and DRS (or CRS), respectively. Note that IRS and DRS are not allowed in the NIRS and NDRS models, respectively. Therefore, we have the following corollary.

Corollary 10.1 (i) If $DMU_o \in E$ exhibits DRS, then all output-oriented super-efficiency DEA models are feasible.

(ii) If $DMU_o \in E$ exhibits IRS, then all input-oriented super-efficiency DEA models are feasible.

By Theorems 10.1 and 10.6, we know that infeasibility indicates that the inputs of an extreme efficient DMU_o can be proportionally increased without limit or that the outputs can be decreased in any positive proportion, while preserving the efficiency of DMU_o. This indicates that the efficiency of DMU_o is always stable under the proportional data changes.

Models (10.2) and (10.7) are useful in the determination of infeasibility while Theorems 10.1 and 10.6 are useful in the sensitivity analysis of efficiency classifications. Table 10.2 summarizes the relationship between infeasibility and the super-efficiency DEA models.

Finally, we note that the super-efficiency VRS models can also be used to estimate RTS. This is a possible new usage of the super-efficiency DEA models.

10.3 Models for Dealing with Infeasibility

A number of studies have tried to solve the problem of VRS super-efficiency model's infeasibility. Lovell and Rouse (2003) suggest using a user-defined scaling factor to make the VRS super-efficiency model feasible. Yet, as indicated in Cook et al. (2009), it is possible that Lovell and Rouse's (Lovell and Rouse 2003) approach assigns the user-defined scaling factor as the super-efficiency score for all DMUs having infeasible solutions. Cook et al. (2009) develop a modified VRS super-efficiency model for efficient DMUs that are infeasible under the standard VRS super-efficiency model. Cook et al. (2009) further define a super-efficiency score with respect to both input and output super-efficiencies.

In fact, as pointed out by Chen (2005), one needs to use both input- and output-oriented super-efficiency models to fully characterize the super-efficiency when infeasibility occurs. Chen (2005) further suggests that one should integrate the input and output super-efficiency scores by solving both the input- and output-oriented VRS super-efficiency models.

Lee et al. (2011) develop a two-stage super-efficiency calculation. Similar to Seiford and Zhu (1998), Lee et al. (2011) point out that the infeasibility of input-oriented super-efficiency occurs when the outputs of the evaluated DMU is outside the production possibility set spanned by the outputs of the remaining DMUs and the infeasibility of output-oriented super-efficiency occurs when the inputs of the evaluated DMU is outside the production possibility set spanned by the inputs of the remaining DMUs. As indicated in Seiford and Zhu (1999) and Chen (2005), infeasibility in the input-oriented super-efficiency can indicate that a particular efficient DMU under evaluation exhibits super-efficiency performance only in outputs. Infeasibility in the output-oriented super-efficiency can indicate that a particular efficient DMU under evaluation exhibits super-efficiency performance only in inputs. Chen (2005) points out that super-efficiency can be regarded as input saving/output surplus achieved by an efficient DMU.

In this section, we present the two-stage procedure developed by Lee et al. (2011). In their first stage, one seeks to simultaneously test whether a VRS super-efficiency model is infeasible, and detect output surpluses (input savings) when infeasibility occurs in the input-oriented (output-oriented) VRS super-efficiency model. Then, in a second stage calculation, a modified VRS super-efficiency model is proposed to calculate the super-efficiency for all the efficient DMUs.

If super-efficiency only exists in inputs (or outputs), Lee, Chu and Zhu's (2011) output-oriented (or input-oriented) super-efficiency model may actually indicate inefficient performance. In other words, infeasibility may imply inefficient performance. This is consistent with the findings in Chen (2005) and Cook et al. (2009).

Like the approach in Cook et al. (2009), when infeasibility occurs, Lee, Chu and Zhu's (Lee et al. 2011) approach may require that (i) both inputs and outputs be decreased to reach the frontier formed by the remaining DMUs under the input-orientation and (ii) both inputs and outputs be increased to reach the frontier formed by the remaining DMUs under the output-orientation.

Note that Lee, Chu and Zhu's (2011) model provides VRS super-efficiency scores that are equivalent to those arising from the VRS super-efficiency model when feasibility is present. When the VRS super-efficiency model is infeasible, their model determines a referent (benchmark) DMU formed by the remaining DMUs and yields a score that characterizes the super-efficiency in inputs and outputs.

In the first stage, Lee et al. (2011) propose to calculate the following model for an efficient DMU_k

$$
\begin{aligned}
&\min \ \sum_{r=1}^{s} s_r \\
\text{s.t.} \ &\sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j y_{rj} + s_r y_{rk} \geq y_{rk} \quad r = 1, 2, \ldots, s \\
&\sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j = 1 \\
&\lambda_j \geq 0, j \neq k \\
&s_r \geq 0, r = 1, 2, \ldots, s
\end{aligned}
\tag{10.10}
$$

This model seeks to determine potential surpluses in individual outputs. We note that model (10.10k is similar to the Seiford and Zhu's (1999) model for testing for infeasibility. Lee et al. (2011) show the following theorem.

Theorem 10.11 Let $(s_1^*, \ldots, s_s^*)$ denote a set of optimal solution in (10.10). Then the input-oriented VRS super efficiency model is feasible if and only if $s_r^* = 0$ for $r = 1, \ldots, s$.

Theorem 10.11 indicates that the input-oriented VRS super-efficiency model is infeasible if and only if there exists some $s_r^* > 0$. Note that these $s_r^* y_{rk}$ are not the output slacks in the standard DEA approach, but represent the output surpluses in DMU_k compared to the frontier formed by the rest of DMUs.

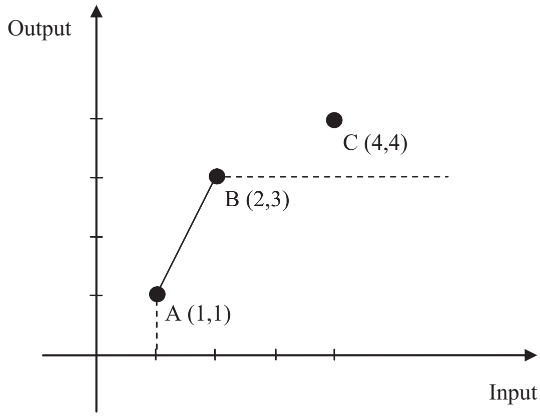
Lee et al. (2011) then establish the following modified VRS super-efficiency model which is always feasible

$$
\begin{aligned}
&\min \ \hat{\theta} \\
\text{s.t.} \ &\sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j x_{ij} \leq \hat{\theta} x_{ik} \quad i = 1, 2, \ldots, m \\
&\sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j y_{rj} + s_r^* y_{rk} \geq y_{rk} \quad r = 1, 2, \ldots, s \\
&\sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j = 1 \\
&\lambda_j \geq 0, j \neq k
\end{aligned}
\tag{10.11}
$$

where $(s_1^*, \ldots, s_s^*)$ are optimal solutions in model (10.10).

Let $\hat{\theta}^*$ be the optimal solution of (10.11) and θ^* be the optimal value to the standard (input-oriented) VRS super-efficiency model. Obviously $\hat{\theta}^* = \theta^*$ when the standard VRS super-efficiency model is feasible, indicating that model (10.11)

Fig. 10.14 Numerical example for super-efficiency



yields the identical super-efficiency score when the standard VRS super-efficiency model is feasible.

We would expect that for efficient DMUs, their input-oriented super-efficiency scores should be greater than one. Such expectation is realistic for the CRS assumption. Under VRS assumption, because of the possible infeasibility, such expectation may not be met due to the fact that an efficient DMU needs to decrease both its inputs and outputs to reach the frontier formed by the rest of DMUs. To further illustrate this point, we consider a simple numerical example from Lee et al. (2011), as shown in Fig. 10.14 where we have three efficient DMUs, A(1,1), B(2,3) and C(4,4).

For DMU C we have infeasibility in the standard VRS super-efficiency model, $s^* = 1/4$ in model (10.10) and $\hat{\theta}^* = 0.5(< 1)$ in model (10.11). This is because DMU B is identified as its benchmark. To reach DMU B, it has to decrease its both input and output.

In a similar manner, we can develop output-oriented VRS super-efficiency model that is always feasible. We first solve the following linear programming problem which seeks to determine potential input savings ($t_i^* x_{ik}$) in the efficient DMU_k compared to the frontier formed by the rest of DMUs:

$$\begin{aligned}
& \min \sum_{i=1}^{m} t_i \\
\text{s.t.} \quad & \sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j x_{ij} - t_i x_{ik} \le x_{ik} \quad i = 1, 2, \ldots, m \\
& \sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j = 1 \\
& \lambda_j \ge 0, j \neq k \\
& t_i \ge 0, i = 1, 2, \ldots, m
\end{aligned} \tag{10.12}$$

Let t_i^* be a set of optimal solution in model (10.12). The standard output-oriented VRS super-efficiency model is feasible if and only if $t_i^* = 0$ for $i = 1, \ldots, m$. We then establish the following output-oriented VRS super-efficiency model which is always feasible.

$$\begin{aligned} & \max \hat{\beta} \\ \text{s.t. } & \sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j x_i^j - t_i^* x_i^k \leq x_i^k \quad i = 1, 2, \ldots, m \\ & \sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j y_r^j \geq \hat{\beta} y_r^k \quad r = 1, 2, \ldots, s \\ & \sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j = 1 \\ & \lambda_j \geq 0, j \neq k \end{aligned} \tag{10.13}$$

It is likely that when infeasibility occurs, the output-oriented super-efficiency score from model (10.13) is greater than one, indicating that output super-efficiency does not exist. Consider DMU A in Fig. 10.14. The standard output oriented super-efficiency model is infeasible for DMU A. We have $t^* = 1$ in model (10.12) and $\hat{\beta}^* = 3(>1)$ in model (10.13). This is because DMU A is projected onto DMU B, and DMU A has to increase both its input and output to reach DMU B. In other words, DMU A has super efficiency in input, but not output. Interested reader is referred to Lee et al. (2011) for two illustrative applications to US cities and Japanese companies of the above approach.

10.4 Zero Data and Infeasibility

In the above discussion, it is assumed that all input and output data are positive. In fact, as discovered in Lee and Zhu (2012), zero data in inputs/outputs can lead to infeasibility in both VRS and CRS situations. In fact, Thrall (1996) and Zhu (1996) point out that the CRS super-efficiency model can also be infeasible when an efficient DMU has zero input values. To address such issue, Lee and Zhu (2012) extend the work of Lee et al. (2011), so that the revised model is feasible when zero data exist in inputs. We should point out that zero output data will not lead to infeasibility of the output-oriented super-efficiency models developed in Lee et al. (2011), Chen and Liang (2011), and Cook et al. (2009). This is because the output side of the constraints can always be satisfied. Therefore, the work of Lee and Zhu (2012) only assumes that some inputs are zero for some efficient DMUs.

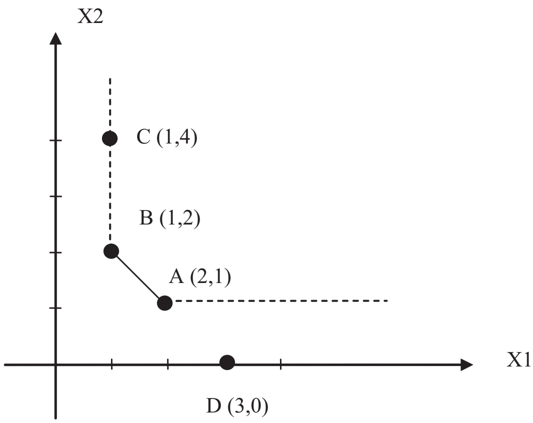


Fig. 10.15 Numerical example for zero data and super-efficiency

Consider a simple numerical example in Lee and Zhu (2012) as shown in Fig. 10.15 where all DMUs have the same output. DMU D is infeasible under input-oriented VRS super-efficiency model. Because all DMUs have the same output in Fig. 10.15. Figure 10.15 can be applied to both CRS and VRS situations, namely, the VRS super-efficiency model is equivalent to the CRS super-efficiency model. Therefore, DMU D is infeasible under both the CRS and VRS super-efficiency models.

Now, let us consider the new super-efficiency DEA model (10.11) developed by Lee et al. (2011) or the equivalent "one model" approach by Chen and Liang (2011).

$$\begin{aligned}
&\min \ \tau + M * \sum_{r=1}^{s} \beta_r \\
\text{s.t.} \ &\sum_{\substack{j=1 \\ j\neq k}}^{n} \lambda_j x_{ij} \le (1+\tau) x_{ik} \quad i = 1,2,\ldots,m \\
&\sum_{\substack{j=1 \\ j\neq k}}^{n} \lambda_j y_{rj} \ge (1-\beta_r) y_{kr} \quad r = 1,2,\ldots,s \\
&\sum_{\substack{j=1 \\ j\neq k}}^{n} \lambda_j = 1 \\
&\lambda_j \ge 0, j \neq k, \beta_r \ge 0
\end{aligned} \tag{10.14}$$

where M is a user-defined large positive number, and in Cook et al. (2009), M is set equal to 10^5. The (modified) super-efficiency score is defined as $1 + \tau^* + \frac{1}{|R|} \sum_{r \in R} \frac{1}{1-\beta_r^*}$ where R is the set of $\beta_r^* > 0$, defined in Lee et al. (2011).

Due to the zero input in D, D is still infeasible under model (10.14). To address this issue, Lee and Zhu (2012) denote $x_i^{\max} = \max_{k=1}^{n}\{x_{ik}\}$ and consider the following modified version of model (10.14)

$$
\begin{aligned}
& \min \ \tau + M * \left(\sum_{r=1}^{s} \beta_r + \sum_{i=1}^{m} t_i \right) \\
\text{s.t.} \ & \sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j x_{ij} - t_i x_i^{\max} \leq (1+\tau) x_{ik} \quad i = 1,2,\ldots,m \\
& \sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j y_{rj} \geq (1-\beta_r) y_{kr} \quad r = 1,2,\ldots,s \\
& \sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j = 1 \\
& \lambda_j \geq 0, j \neq k, \beta_r, t_i \geq 0, \tau \ \textit{is unrestricted}
\end{aligned}
\tag{10.15}
$$

The only difference between (10.14) and (10.15) is the added $t_i x_i^{\max}$ in input constraints. Infeasibility occurs when (some of) input constraints of (10.14) are violated when the evaluated DMU has zero inputs. The DMU D in Fig. 10.15 is such an example. To remedy such situation, we deduct the term $t_i x_i^{\max}$ from the left hand side of the input constraints so that these constraints will not be violated when zero input occurs. The amount to be deducted in Fig. 10.15 for DMU D is 1, the vertical distance from D to horizontal dashed line above D. The reasons for expressing the deducted amount in terms of $t_i x_i^{\max}$ are two folds. The first is unit invariant, whose rationale is as follows. Assume the ith input is scaled by a factor b. The first constraint of model (10.15) becomes

$$
\sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j b x_{ij} - t_i b x_i^{\max} \leq b(1+\tau) x_{ik} \quad i = 1,2,\ldots,m
$$

which is equivalent to the original constraint. Therefore, the optimal solution t_i^* remains the same, which means the optimal solution of model (10.15) is unit invariant.The second reason for using $t_i x_i^{\max}$ is that $t_i x_i^{\max}$ will not be zero when x_i^k is zero. If we replace $t_i x_i^{\max}$ with $t_i x_i^k$, the amount deducted $t_i x_i^k$ will be zero when x_i^k is zero. In Fig. 10.15, $t_2^* x_2^{\max} = 1$ for DMU D, where $t_2^* = 0.25, x_2^{\max} = 4$. Note that the output constraints in (10.14) hold even when some outputs has zero. Therefore, the output constraints remain unchanged. This is also the reason why the output-oriented super-efficiency models developed in Lee et al. (2011), Chen and Liang (2011), and Cook et al. (2009) are always feasible if outputs have zero data.

Denote $R = \{r \mid \beta_r^* > 0\}$ and $I = \{i \mid t_i^* > 0\}$. Lee and Zhu (2012) then define the following indexes

Input savings index:

$$\hat{i} = \begin{cases} 0 & if \;\; I = \phi \\ \dfrac{\sum_{i\in I}\left(\dfrac{1+t_i^*}{1}\right)}{|I|} & if \; I \neq \phi \end{cases}$$

Output surplus index:

$$o = \begin{cases} 0 & if \;\; R = \phi \\ \dfrac{\sum_{r\in R}\left(\dfrac{1}{1-\beta_r^*}\right)}{|R|} & if \; R \neq \phi \end{cases}$$

Then, the super-efficiency score can be defined as

$$\breve{\theta} = 1+\tau^* + o + \hat{i}$$

For D, model (10.15) yields $1+\tau^* = 0.666667, t_1^* = 0, t_2^* = 0.25, \beta_1^* = 0$, and the super-efficiency score is 1.916667. Model (10.15) projects D to A in Fig. 10.15.

The efficiency measure $\breve{\theta}$ consists of three parts: the radial efficiency $1+\tau^*$, the output surplus index o, and the input saving index $\hat{i}$. The input saving index is defined to be $\dfrac{\sum_{i\in I}\left(\dfrac{1+t_i^*}{1}\right)}{|I|}$ when $\{i \mid t_i^*\}$ is not empty, where $\dfrac{\sum_{i\in I}\left(\dfrac{1+t_i^*}{1}\right)}{|I|} = \dfrac{\sum_{i\in I}\left(\dfrac{x_i^{\max} + t_i^* x_i^{\max}}{x_i^{\max}}\right)}{|I|}$ reflects how far the DMU k is below the dashed horizontal efficient boundary (see the dashed line through A in Fig. 10.15). When a DMU falls below the dashed horizontal line in Fig. 10.15, like DMU D, its efficiency of the original VRS is 1 which implies that its supper efficiency should be no less than 1, which is guaranteed by our input saving index. For example, for DMU D in Fig. 10.15, its radial efficiency, output surplus index and input saving index are 0.666667, 0 and 1.25 respectively,which means that to get to the projection A, DMU D should scale down its input by 0.666667 and then move upward by 1 ($t_i^* x_i^{\max} = 1$). In the original VRS model, DMU D is efficient, which implies that DMU D should have super efficiency not less than 1. The super efficiency of D is 1.91667 because its input saving index is 1.25, which means that DMU D uses less input than its projection A. Note that input saving index is greater than 1 if input saving exits.

The output surplus index $\frac{\sum_{r \in R}\left(\frac{1}{1-\beta_r^*}\right)}{|R|}$ reflects how far the DMU k is above the dashed efficient boundary (see the dashed line in Fig. 10.14). For DMU C in Fig. 10.14, its radial efficiency, input saving index and output surplus index are 0.5, 0 and 4/3 respectively. Since C is efficient in the original VRS model, the super efficiency should be no less than 1. The super efficiency of C is 11/6, which is greater than 1 because of its output surplus index.

10.5 Slack-Based Super Efficiency

Under the radial DEA models, super-efficiency DEA models are obtained simply by removing the DMU under evaluation from the reference set. However, the above procedure cannot be applied directly to non-radial DEA models (e.g., the slack-based models discussed in Chap. 5) to yield the super-efficiency versions. As demonstrated in Tone (2002), for non-radial or slacks-based DEA models, one needs to identify the efficient DMUs first and then modify the DEA model.

Suppose that DMU_o is an efficient DMU, and the slacks-based super-efficiency of DMU_o is defined as the optimal value δ_o^* to the following problem (Tone 2002).

$$
\begin{aligned}
\delta_o^* = \min \delta_o &= \frac{\frac{1}{m}\sum_{i=1}^{m} \bar{x}_{io} / x_{io}}{\frac{1}{s}\sum_{r=1}^{s} \bar{y}_{ro} / y_{ro}} \\
s.t.\ \bar{x}_{io} &\geq \sum_{j=1,\, j\neq o}^{n} \lambda_j x_{ij}, i = 1,2,\ldots,m \\
\bar{y}_{ro} &\leq \sum_{j=1,\, j\neq o}^{n} \lambda_j y_{rj}, r = 1,2,\ldots,s \\
\bar{x}_{io} &\geq x_{io}, i = 1,2,\ldots,m \\
\bar{y}_{ro} &\leq y_{ro}, r = 1,2,\ldots,s \\
\lambda_j, \bar{y}_{ro} &\geq 0, j = 1,2,\ldots,n, j \neq o, r = 1,2,\ldots,s
\end{aligned}
\tag{10.16}
$$

In Tone's (2002) super-efficiency model (10.16), a fractional objective function is used, and additional constraints of $\bar{x}_{io} \geq x_{io}$ and $\bar{y}_{ro} \leq y_{ro}$ are introduced. Due to the objective function, the model also requires positive input and output data for efficient DMUs, i.e. $x_{ij} > 0$ and $y_{rj} > 0$. Using the Charnes-Cooper transformation, model (10.16) can be transformed into a linear program.

Du et al. (2010) show that we can also build super efficiency models based upon the additive DEA model (Charnes et al. 1982) or slack-based models discussed in Chap. 5. To facilitate the development, we consider the following slack-based model

$$\begin{aligned}
&\max \sum_{i=1}^{m} s_{io}^{-} + \sum_{r=1}^{s} s_{ro}^{+} \\
&s.t. \sum_{j=1}^{n} \lambda_j x_{ij} + s_{io}^{-} = x_{io}, i = 1,2,...,m \\
&\qquad \sum_{j=1}^{n} \lambda_j y_{rj} - s_{ro}^{+} = y_{ro}, r = 1,2,...,s \\
&\qquad \lambda_j, s_{io}^{-}, s_{ro}^{+} \geq 0, j = 1,2,...,n, i = 1,2,...,m, r = 1,2,...,s
\end{aligned} \tag{10.17}$$

where s_{io}^{-} and s_{ro}^{+} are input and output slacks.

Suppose DMU_o is efficient. To obtain the super-efficiency of DMU_o under model (10.17), we cannot simply modify model (10.17) by removing DMU_o from the reference set. If we do that, the resulting model may not have a feasible solution.

For an efficient DMU_o under the additive DEA model (10.17), we have the following super-efficiency model:

$$\begin{aligned}
&\alpha_o^{*} = \min \alpha_o = \sum_{i=1}^{m} t_{io}^{-} + \sum_{r=1}^{s} t_{ro}^{+} \\
&s.t. \sum_{j=1, j \neq o}^{n} \lambda_j x_{ij} \leq x_{io} + t_{io}^{-}, i = 1,2,...,m \\
&\qquad \sum_{j=1, j \neq o}^{n} \lambda_j y_{rj} \geq y_{ro} - t_{ro}^{+}, r = 1,2,...,s \\
&\qquad \lambda_j, t_{io}^{-}, t_{ro}^{+} \geq 0, j = 1,2,...,n, j \neq o, i = 1,2,...,m, r = 1,2,...,s
\end{aligned} \tag{10.18}$$

After DMU_o is removed from the reference set of model (10.17), we need to modify the constraints and objective of model (10.18) to get the super-efficiency model. The constraints should be modified because we need to increase the inputs and decrease the outputs for DMU_o to reach the frontier constructed by the remaining DMUs. We change the objective from maximization to minimization so that the resulting model is bounded.

We can also use a different objective function for model (10.18) so that the resulting model is unit invariant, for example,

$$
\begin{aligned}
\beta_o^* = \min \beta_o &= \frac{1}{m+s}\left(\sum_{i=1}^{m}\frac{t_{io}^-}{x_{io}} + \sum_{r=1}^{s}\frac{t_{ro}^+}{y_{ro}}\right) \\
s.t.\ x_{io} + t_{io}^- &\ge \sum_{j=1, j\neq o}^{n} \lambda_j x_{ij}, i = 1,2,\ldots,m \\
y_{ro} - t_{ro}^+ &\le \sum_{j=1, j\neq o}^{n} \lambda_j y_{rj}, r = 1,2,\ldots,s \\
\lambda_j, t_{io}^-, t_{ro}^+ &\ge 0, j = 1,2,\ldots,n, j \neq o, i = 1,2,\ldots,m, r = 1,2,\ldots,s
\end{aligned}
\tag{10.19}
$$

Now, if we let $\left\{\alpha_o^*; \lambda_j^*(\alpha), j = 1,2,\ldots,n, j \neq o; t_{io}^{-*}(\alpha), i = 1,2,\ldots,m; t_{ro}^{+*}(\alpha), r = 1,2,\ldots,s\right\}$ and $\left\{\beta_o^*; \lambda_j^*(\beta), j = 1,2,\ldots,n, j \neq o; t_{io}^{-*}(\beta), i = 1,2,\ldots,m; t_{ro}^{+*}(\beta), r = 1,2,\ldots,s\right\}$ be an optimal solution to models (10.18) and (10.19), respectively. Then we can define

$$
\hat{\delta}_o^*(\alpha) = \frac{\frac{1}{m}\sum_{i=1}^{m}\left(x_{io} + t_{io}^{-*}(\alpha)\right)/x_{io}}{\frac{1}{s}\sum_{r=1}^{s}\left(y_{ro} - t_{ro}^{+*}(\alpha)\right)/y_{ro}}
\tag{10.20}
$$

and

$$
\hat{\delta}_o^*(\beta) = \frac{\frac{1}{m}\sum_{i=1}^{m}\left(x_{io} + t_{io}^{-*}(\beta)\right)/x_{io}}{\frac{1}{s}\sum_{r=1}^{s}\left(y_{ro} - t_{ro}^{+*}(\beta)\right)/y_{ro}}
\tag{10.21}
$$

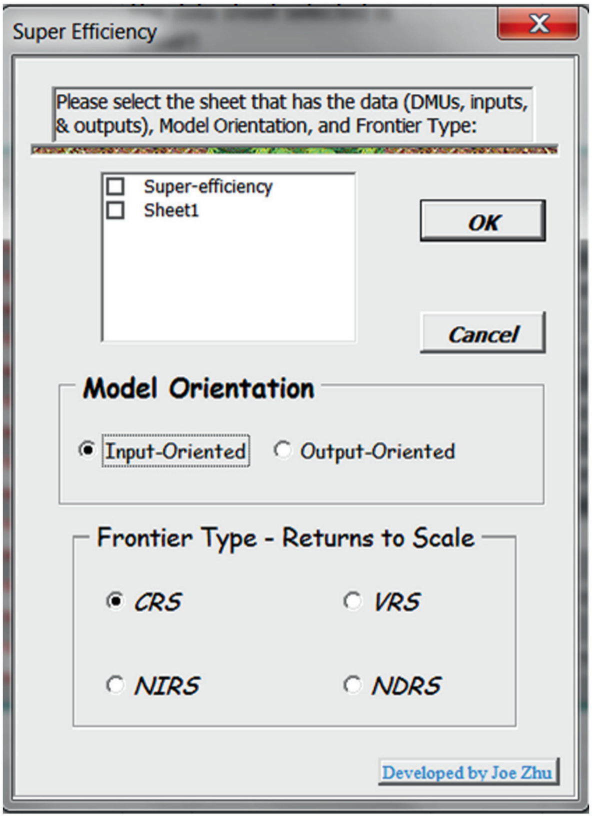
which are in the exact format of the objective function used in model (10.16). Note that $\hat{\delta}_o^*(\alpha) \ge 1$ and $\hat{\delta}_o^*(\beta) \ge 1$. Thus we can use $\hat{\delta}_o^*(\alpha)$ and $\hat{\delta}_o^*(\beta)$ as the super-efficiency scores for model (10.18) and (10.19), respectively.

Note that the models are developed under the CRS assumption. If we add $\sum_{j=1, j\neq o}^{n} \lambda_j = 1$ into the models, we obtain slacks-based super-efficiency models for VRS. Du et al. (2010) show that the slacks-based super-efficiency models are always feasible under CRS or VRS assumption.

10.6 Solving Super-Efficiency Using DEA Frontier

To run the super-efficiency models presented in Table 10.1, select the "Super-efficiency" menu item. You will be prompted a form shown in Fig. 10.16 for specifying the super-efficiency models. The results are reported in the "Super-efficiency" sheet.

Fig. 10.16 Super efficiency models



To run the slack-based super efficiency, the user selects the slack-based super efficiency menu item. The user needs to select a frontier type. There are two options for the objective function, as shown in Fig. 10.17. The efficiency scores defined in (10.20) and (10.21) are reported in sheet "Slack Super". This sheet also reports the individual slacks.

Note that results for inefficient DMUs are not reported, as they are the same if the standard slacks-based model (Chap. 5) is used.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_10) contains supplementary material, which is available to authorized users.
Caption of the data object (Excelfiles.zip 711 kb)

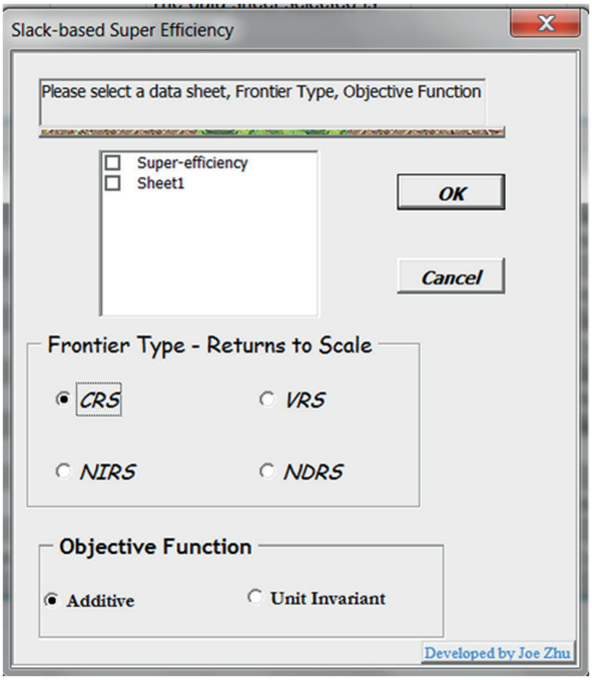


Fig. 10.17 Slack-base super efficiency models

References

Andersen, P., & Petersen, N. C. (1993). A procedure for ranking efficient units in data envelopment analysis. *Management Science, 39,* 1261–1264.

Charnes, A., Cooper, W. W., Seiford, L. M., & Stutz, J. (1982). A multiplicative model for efficiency analysis. *Socio-Economic Planning Sciences, 16*(5), 223–224.

Charnes, A., Cooper, W. W., & Thrall, R. M. (1991). A structure for classifying and characterizing efficiencies and inefficiencies in DEA. *Journal of Productivity Analysis, 2,* 197–237.

Charnes, A., Haag, S., Jaska, P., & Semple, J. (1992). Sensitivity of efficiency classifications in the additive model of data envelopment analysis. *International Journal of Systems Science, 23,* 789–798.

Chen, Y. (2004). Ranking efficient units in DEA. *OMEGA, 32,* 213–219.

Chen, Y. (2005). Measuring super-efficiency in DEA in the presence of infeasibility. *European Journal of Operational Research, 161,* 545–551.

Chen, Y., & Liang, L. (2011). Super-efficiency DEA in the presence of infeasibility: One model approach. *European Journal of Operational Research, 213*(1), 359–360.

Chen, Y., & Sherman, D. H. (2004). The benefits of non-radial vs. radial super-efficiency DEA: An application to burden-sharing amongst NATO member nations. *Socio-Economic Planning Sciences, 38*(4), 307–320.

Cook, W. D., Liang, L., Zha, Y., & Zhu, J. (2009). A modified super-efficiency DEA model for infeasibility. *Journal of Operational Research Society, 60*, 276–281.

Du, J., Liang, L., & Zhu, J. (2010). A slacks-based measure of super-efficiency in data envelopment analysis: A comment. *European Journal of Operational Research, 204,* 694–697.

Lee, H.-S., & Zhu, J. (2012). Super-efficiency infeasibility and zero data in DEA. *European Journal of Operational Research, 216,* 429–433.

Lee, H.-S., Chu, C.-W., & Zhu, J. (2011). Super-efficiency DEA in the presence of infeasibility. *European Journal of Operational Research, 212,* 141–147.

Lovell, C. A. K., & Rouse, A. P. B. (2003). Equivalent standard DEA models to provide super-efficiency scores. *Journal of the Operational Research Society, 54*(1), 101–108.

Seiford, L. M., & Zhu, J. (1998). Stability regions for maintaining efficiency in data envelopment analysis. *European Journal of Operational Research, 108*(1), 127–139.

Seiford, L. M., & Zhu, J. (1999). Infeasibility of super efficiency data envelopment analysis models. *INFOR, 37*(2), 174–187.

Thrall, R. M. (1996). Duality, classification and slacks in DEA. *Annals of Operations Research, 66,* 109–138.

Tone, K. (2002). A slacks-based measure of super-efficiency in data envelopment analysis. *European Journal of Operational Research, 143,* 32–41.

Wilson, P. (1995). Detecting influential observations in data envelopment analysis. *Journal of Productivity Analysis, 6,* 27–45.

Zhu, J. (1996). Robustness of the efficient DMUs in data envelopment analysis. *European Journal of Operational Research, 90*(3), 451–460.

Zhu, J. (2001). Super-efficiency and DEA sensitivity analysis. *European Journal of Operational Research, 129*(2), 443–455.

Chapter 11
Sensitivity Analysis

11.1 DEA Sensitivity Analysis

One important issue in DEA which has been studied by many DEA researchers is the efficiency sensitivity to perturbations in the data. Some DEA sensitivity studies focus on the sensitivity of DEA results to the variable and model selection, e.g., Ahn and Seiford (1993). Most of the DEA sensitivity analysis studies focus on the misspecification of efficiency classification of a test DMU. However, note that DEA is an extremal method in the sense that all extreme points are characterized as efficient. If data entry errors occur for various DMUs, the resulting isoquant may vary substantially. We say that the calculated frontiers of DEA models are stable if the frontier DMUs that determine the DEA frontier remain on the frontier after particular data perturbations are made.

By updating the inverse of the basis matrix associated with a specific efficient DMU in a DEA linear programming problem, Charnes et al. (1985a) study the sensitivity of DEA model to a single output change. This is followed by a series of sensitivity analysis articles by Charnes and Neralic in which sufficient conditions preserving efficiency are determined (see, e.g. Charnes and Neralic (1990)).

Another type of DEA sensitivity analysis is based on super-efficiency DEA models. Charnes et al. (1992), Rousseau and Semple (1995) and Charnes et al. (1996) develop a super-efficiency DEA sensitivity analysis technique for the situation where simultaneous proportional change is assumed in all inputs and outputs for a specific DMU under consideration. This data variation condition is relaxed in Zhu (1996) and Seiford and Zhu (1998a) to a situation where inputs or outputs can be changed individually and the entire (largest) stability region which encompasses that of Charnes et al. (1992) is obtained. As a result, the condition for preserving efficiency of a test DMU is necessary and sufficient.

The DEA sensitivity analysis methods we have just reviewed are all developed for the situation where data variations are only applied to the test efficient DMU and the data for the remaining DMUs are assumed fixed. Obviously, this assumption may not be realistic, since possible data errors may occur in each DMU. Seiford and Zhu (1998b) generalize the technique in Zhu (1996) and Seiford and Zhu (1998a) to the worst-case scenario where the efficiency of the test DMU is deteriorating while

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_11,

the efficiencies of the other DMUs are improving. In their method, same maximum percentage data change of a test DMU and the remaining DMUs is assumed and sufficient conditions for preserving an extreme efficient DMU's efficiency are determined. Note that Thompson et al. (1994) use the SCSC (strong complementary slackness condition) multipliers to analyze the stability of the CRS model when the data for all efficient and all inefficient DMUs are simultaneously changed in opposite directions and in same percentages. Although the data variation condition is more restrictive in Seiford and Zhu (1998b) than that in Thompson et al. (1994), the super-efficiency based approach generates a larger stability region than the SCSC method. Also, the SCSC method is dependent upon a particular SCSC solution, among others, and therefore the resulting analysis may vary (see Cooper et al. 2001).

Seiford and Zhu (1999) (Chap. 10) develop the necessary and sufficient conditions for infeasibility of various super-efficiency DEA models. Although the super-efficiency DEA models employed in Charnes et al. (1992) and Charnes et al. (1996) do not encounter the infeasibility problem, the models used in Seiford and Zhu (1998a) do. Seiford and Zhu (1998a) discover the relationship between infeasibility and stability of efficiency classification. That is, infeasibility means that the efficiency of the test DMU remains stable to data changes in the test DMU. Furthermore, Seiford and Zhu (1998b) show that this relationship is also true for the simultaneous data change case and other DEA models, such as the VRS model and the additive model of Charnes et al. (1985b). This finding is critical since super-efficiency DEA models in Seiford and Zhu (1998b) are frequently infeasible for real-world data sets, indicating efficiency stability with respective to data variations in inputs/outputs associated with infeasibility.

Zhu (2001) extends the results in Seiford and Zhu (1998a, b) to a situation when different data variations are applied to the test DMU and the remaining DMUs, respectively.

In this chapter, we focus on the DEA sensitivity analysis methods based upon super-efficiency DEA models that are developed by Zhu (1996, 2001) and Seiford and Zhu (1998a, b). For the DEA sensitivity analysis based upon the inverse of basis matrix, the reader is referred to Neralic (1994).

Since an increase of any output or a decrease of any input cannot worsen the efficiency of DMU_o, we restrict our attention to decreases in outputs and increases in inputs for DMU_o. We consider proportional increases of inputs or proportional decreases of outputs of the form

$$\hat{x}_{io} = \beta_i x_{io} \qquad \beta_i \geq 1, i = 1,\ldots,m \tag{11.1}$$

$$\hat{y}_{ro} = \alpha_r y_{ro} \quad 0 < \alpha_r \leq 1, r = 1,\ldots,s \tag{11.2}$$

where x_{io} $(i = 1, 2,\ldots,m)$ and y_{ro} $(r = 1,\ldots, s)$ are respectively, the inputs and outputs for a specific extreme efficient $DMU_o = DMU_{j_o}$ among n DMUs.

Zhu (1996) provides a super-efficiency model to compute a stability region in which DMU_o remains efficient. Specifically, for an increase in inputs of form (11.1), this model is given by

$$
\begin{aligned}
&\min \beta_k^o \qquad \text{for each } k = 1,\ldots,m \\
&\text{subject to} \\
&\sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j x_{kj} \leq \beta_k^o x_{ko} \\
&\sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j x_{ij} \leq x_{io} \qquad i \neq k \\
&\sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r = 1,\ldots,s \\
&\lambda_j, \beta_k^o \geq 0
\end{aligned}
\tag{11.3}
$$

where x_{ij} and y_{rj} are the ith input and rth output of DMU_j $(j = 1,\ldots,n)$, respectively. It can be seen that model (11.3) is developed from the input-oriented measure-specific model—CRS kth input-specific model.

Zhu's (1996) approach requires two assumptions: (i) the hyperplane constructed by the m hypothetical observations obtained from model (11.3) is not dominated by other DMUs and (ii) model (11.3) is feasible. However, in real word situations, these two assumptions may not be satisfied.

Note that any increase of input or any decrease of output will cause the DMUs in set E' (efficient but not extreme efficient) to become inefficient. For those DMUs in set F (weakly efficient with non-zero slacks), the amount of inputs (or outputs) which have non-zero slacks can be increased (or decreased) without limit, and these DMUs will remain in the set F. However, for inputs and outputs which have no slack, any input increase of (11.1) or any output decrease of (11.2) will cause these DMUs to become inefficient. Therefore, the sensitivity issue of DMUs in set E' or F is straightforward if not trivial. Thus, we first focus on the efficiency of the DMUs in set E, i.e., the extreme efficient DMUs.

11.2 Stability Region[1]

11.2.1 Input Stability Region

For $DMU_o \in E$, we first suppose that (11.3) is feasible for each input and consider input changes of form (11.1). As shown in Zhu (1996), the optimal value to (11.3),

[1] Part of the material in this section is adapted from European Journal of Operational Research, Vol 108, Seiford, L.M. and Zhu, J., Stability regions for Maintaining Efficiency in DEA, 127–139, 1998, with permission from Elsevier Science.

β_k^{o*}, gives the maximum possible increase for each individual input which allows DMU_o to remain efficient with the other inputs and all outputs held constant. Also, (11.3) provides m hypothetical frontier points (efficient DMUs) when DMU_o is excluded from the reference set. The kth point is generated by increasing the kth input from x_{ko} to $\beta_k^{o*} x_{ko}$ and holding all other inputs and outputs constant. We denote these k hypothetical observations by

$$DMU(\beta_k^{o*}) = (x_{1o}, \ldots, \beta_k^{o*} x_{ko}, \ldots, x_{mo}, y_{1o}, \ldots, y_{so}) \tag{11.4}$$

Consider the following linear programming problem

$$\begin{aligned} & \min \sum_{i=1}^{m} \rho_i^o \\ & \text{subject to} \\ & \sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j x_{ij} \le \rho_i^o x_{io} \qquad i = 1, \ldots, m \\ & \sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j y_{rj} \ge y_{ro} \qquad r = 1, \ldots, s \\ & \rho_i^o \ge 1, \lambda_j \ge 0. \end{aligned} \tag{11.5}$$

This model determines the smallest summation of the proportions to move DMU_o to the boundary of the convex hull of the other DMUs.

Lemma 11.1 Denote the optimal solution for (11.5) by $\rho_i^{o*}(i = 1, 2, \ldots, m)$. For $i = 1, 2, \ldots, m$, we have $\beta_i^{o*} \ge \rho_i^{o*} \ge 1$.

[Proof]: Suppose for some i_o, $\beta_{i_o}^{o*} < \rho_{i_o}^{o*}$, then $\sum_{j \neq o} \lambda_j x_{ij} \le \beta_{i_o}^{o*} x_{i_o o} < \rho_{i_o}^{o*} x_{i_o o}$. Therefore, any optimal solution to (11.3) is a feasible solution to (11.5). Thus, $m - 1 + \beta_{i_o}^{o*} \ge \sum_{i=1}^{m} \rho_i^{o*} > m - 1 + \beta_{i_o}^{o*}$. A contradiction. ■

Associated with $\rho_i^{o*}(i = 1, 2, \ldots, m)$, m additional points (or DMUs) can be generated as

$$DMU(\rho_k^{o*}) = (x_{1o}, \ldots, \rho_k^{o*} x_{ko}, \ldots, x_{mo}, y_{1o}, \ldots, y_{so}) \tag{11.6}$$

Theorem 11.1 For $DMU_o = (x_{1o}, \ldots, x_{mo}, y_{1o}, \ldots, y_{so})$, denote an increase of inputs of form (11.1) by $DMU_o(\beta_1, \ldots, \beta_m) = (\beta_1 x_{1o}, \ldots, \beta_m x_{mo}, y_{1o}, \ldots, y_{so})$ and define $\Omega^o = \{(\beta_1, \ldots, \beta_m) \mid 1 \le \beta_i \le \rho_i^{o*}, i = 1, \ldots, m\}$. If $(\beta_1, \ldots, \beta_m) \in \Omega^o$, then $DMU_o(\beta_1, \ldots, \beta_m)$ remains efficient.

[Proof]: Suppose $(\tilde{\beta}_1,\ldots,\tilde{\beta}_m)\in\Omega^o$ and DMU_o with inputs of $\tilde{\beta}_i x_{io}\,(I = 1,\ldots,m)$ is inefficient. In fact, (11.5) is equivalent to the following linear programming problem where $\rho_i^o x_{io} = x_{io} + \delta_i^o$ (and $\tilde{\beta}_i x_{io} = x_{io} + \tilde{\delta}_i$ in which $0 \le \tilde{\delta}_i \le \delta_i^{o*}$)

$$
\begin{aligned}
&\min \sum_{i=1}^{m} \delta_i^o \\
&\text{subject to} \\
&\sum_{\substack{j=1\\ j\ne o}}^{n} \lambda_j x_{ij} - \delta_i^o \le x_{io} \qquad i = 1,\ldots,m \\
&\sum_{\substack{j=1\\ j\ne o}}^{n} \lambda_j y_{rj} \ge y_{ro} \qquad r = 1,\ldots,s \\
&\delta_i^o, \lambda_j \ge 0.
\end{aligned}
$$

$\sum_{i=1}^{m}\delta_i^o$ is $\sum_{i=1}^{m}\delta_i^{o*}$ at optimality when $\sum_{i=1}^{m}\rho_i^o = \sum_{i=1}^{m}\rho_i^{o*}$, and there exist $\lambda_j\ (j \ne 0)$, s_i^-, $s_r^+ \ge 0$ that satisfy

$$
\begin{aligned}
&\sum_{\substack{j=1\\ j\ne o}}^{n} \lambda_j x_{ij} + s_i^- = x_{io} + \tilde{\delta}_i \qquad i = 1,\ldots,m \\
&\sum_{\substack{j=1\\ j\ne o}}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \qquad r = 1,\ldots,s \\
&\lambda_j, s_i^-, s_r^+ \ge 0 \qquad j = 1,\ldots,n
\end{aligned}
$$

violating the optimality of $\sum_{i=1}^{m}\delta_i^{o*}$. Thus, $DMU_o(\beta_1,\ldots,\beta_m)$ with inputs of $\tilde{\beta}_i x_{io}\,(i = 1,\ldots,m)$ is efficient. ■

Definition 11.1 A region of allowable input increases is called an **Input Stability Region** *if and only if* DMU_o remains efficient after such increases occur.

The input stability region (ISR) determines by how much all of DMU_o's inputs can be increased before DMU_o is within the convex hull of the other DMUs. From Lemma 11.1 and Theorem 11.1 we know that (i) Ω^o is only a subset of ISR and (ii) the sets $\{\beta_i \mid 1 \le \beta_i \le \beta_i^{o*}\} i = 1,\ldots,m$, form part of the boundary of ISR.

If the input hyperplane constructed by the *m* points, $DMU(\beta_k^{o*})$, associated with the optimal values to (11.3), is not dominated by other DMUs except DMU_o, i.e., that input hyperplane is a new efficient facet when excluding DMU_o, then the following set Γ^o is precisely the ISR (Zhu 1996)

$$
\Gamma^o = \{(\beta_1,\ldots,\beta_m) \mid 1 \le \beta_i \le \beta_i^{o*}, i = 1,\ldots,m \text{ and } B_1^o\beta_1 + \ldots + B_m^o\beta_m \le 1\}
$$

where B_1^o, …, B_m^o are parameters determined by the following system of equations

$$\begin{cases} \beta_1^{o*} B_1^o + \quad B_2^o \quad + \ldots + \quad B_m^o = 1 \\ B_1^o + \beta_2^{o*} B_2^o \quad + \ldots + \quad B_m^o = 1 \\ \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\ B_1^o + \quad B_2^0 \quad + \ldots + \beta_m^{o*} B_m^o = 1 \end{cases}$$

Zhu (1996) shows the following result

Theorem 11.2 In the case of input increases of form (11.1), for any extreme efficient DMU_o, if the m points, $DMU(\beta_k^{o*})$, which are associated with the optimal β values to (11.3), determine an efficient input hyperplane, then DMU_o remains efficient *if and only if* $(\beta_1, \ldots, \beta_m) \in \Gamma^o$.

Next, suppose that the hyperplane constructed by the m points in (11.4) is dominated by some other DMUs which are inefficient when including DMU_o. In this case, the ISR is no longer the set of Γ^o. Thus, we develop the following procedure.

Initiation (t = 0) Solve model (11.3) for each k, k = 1,…, m. If the input hyperplane, which is determined by the m points of $DMU(\beta_k^{o*})$ in (11.4), is not dominated by other DMUs, then we obtain the ISR defined by Γ^o. Otherwise solve model (11.5). Associated with the optimal solutions to (11.5), ρ_i^{o*}, we obtain m new points, $DMU(\rho_i^{o*})\,(i = 1, 2, \ldots, m)$ as given in (11.6) and Ω^o.

Iteration t = 1, 2,…, T At iteration t, for each point of iteration t-1, say point p, which is associated with the optimal ρ values to (11.5), we solve model (11.3) at each new kth input, k = 1, 2,…, m and apply the *Stopping Rule*. (a) If the rule is satisfied for a particular point p, then we have a similar set Γ_p^t determined by the optimal β values, say β_k^{p*}, k = 1,…, m. We continue for the remaining points. (b) Otherwise solve model (11.5) for point p to obtain m new points and a similar set Ω_p^t determined by the optimal ρ values, say ρ_k^{p*}. Apply iteration t + 1 to each of these m new points.

Stopping Rule If the input hyperplane determined by the$m$ points that are associated with the m optimal β values is not dominated by other DMUs, then iteration stops.

From the above procedure, we see that if a Γ-like set is obtained, then the iteration stops at a specific point. i.e., the Γ-like set indicates the termination of the iteration.

Theorem 11.3 The input stability region is a union of Ω^o and some Ω_p^t and some Γ_p^t.

[Proof]: Obviously, DMU_o remains efficient when its input increases $(\beta_1, \ldots, \beta_m)$ belong to Ω^oor any of the Ω_p^t or Γ_p^t. Conversely, from the iterations we know that the ISR is connected. Since the input increases occurred in Ω-like sets, DMU_o is first moved to a particular point p which is used to construct a Γ-like set. By Theorem

11.2, we know that the sets of Γ_p^t are the boundary sets of the ISR. This means that if further input increases are not in this kind of set, then DMU_o will become inefficient. Therefore, if DMU_o remains efficient, then the input increase of form (11.1) must be in Ω^o or any of the Ω_p^t or Γ_p^t. ■

11.2.2 Output Stability Region

Similarly, Seiford and Zhu (1998a) develop a sensitivity analysis procedure for output decreases of (11.2). For a specific extreme efficient DMU_o, we consider the following linear program (Zhu 1996)

$$
\begin{aligned}
&\min \alpha_k^o \qquad \text{for each } k = 1,\ldots,s \\
&\text{subject to} \\
&\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{kj} \geq \alpha_k^o y_{ro} \\
&\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r \neq k \\
&\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \leq x_{io} \qquad i = 1,\ldots,m \\
&\lambda_j, \alpha_k^o \geq 0
\end{aligned}
\tag{11.7}
$$

Model (11.7) is a super-efficiency model based upon the CRS kth output specific model. The optimal values to (11.7), α_k^{o*}, $k=1,\ldots, s$, give s hypothetical frontier points (or DMUs) and a set Λ^o defined as follows

$$DMU(\alpha_k^{o*}) = (x_{10},\ldots,x_{mo}, y_{10},\ldots, \alpha_k^{o*} y_{ko},\ldots y_{so})$$

and

$$\Lambda^o = \{(\alpha_1,\ldots,\alpha_m) \mid \alpha_r^{o*} \leq \alpha_r \leq 1, r = 1,\ldots,s \text{ and } A_1^o \alpha_1 + \ldots + A_s^o \alpha_s \geq 1\}$$

in which the parameters of A_r^o are determined by the following system of equations

$$
\begin{cases}
\alpha_1^{o*} A_1^o + A_2^o + \ldots + A_s^o = 1 \\
A_1^o + \alpha_2^{o*} A_2^o + \ldots + A_s^o = 1 \\
\ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\
A_1^o + A_2^o + \ldots + \alpha_s^{o*} A_s^o = 1
\end{cases}
$$

Here, we rewrite the result of Zhu (1996) as the following theorem

Theorem 11.4 In the case of output decreases of form (11.2), for any extreme efficient DMU_o, if the s points, $DMU(\alpha_k^{o*})$, which are associated with the optimal α values to (11.7), determine an efficient output hyperplane, then DMU_o remains efficient *if and only if* $(\alpha_1, \ldots, \alpha_s) \in \Lambda^o$.

Definition 11.2 A region of allowable output decreases is called an **Output Stability Region** *if and only if* DMU_o remains efficient after such decreases occur.

Now, suppose that the output hyperplane constructed by the s points, $DMU(\alpha_k^{o*})$, is dominated by some other DMUs which are originally inefficient, then the output stability region (OSR) is not the set Λ^o. We consider the following linear programming problem

$$\begin{aligned}
&\max \sum_{r=1}^{s} \varphi_r^o \\
&\text{subject to} \\
&\sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j y_{rj} \geq \varphi_r^o y_{ro} \qquad r = 1, \ldots, s \\
&\sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j x_{ij} \leq x_{io} \qquad i = 1, \ldots, m \\
&\lambda_j \geq 0, \varphi_r^o \leq 1.
\end{aligned} \tag{11.8}$$

Similar to Theorem 11.1, we have

Theorem 11.5 For a decrease in outputs of form (11.2), if

$$(\alpha_1, \ldots, \alpha_s) \in \Psi^o = \{(\alpha_1, \ldots, \alpha_m) \mid \ \varphi_r^{o*} \leq \alpha_r \leq 1, r = 1, \ldots, s\}$$

then DMU_o remains efficient.

We also have the following s new points that associated with the optimal solutions, φ_r^{o*}, of (11.8)

$$DMU(\varphi_k^{o*}) = (x_{10}, \ldots, x_{mo}, y_{10}, \ldots, \varphi_k^{o*} y_{ko}, \ldots y_{so})$$

To obtain the output stability region, we apply model (11.7) and model (11.8) at each iteration in the procedure for input stability region until no Λ-like sets can be obtained. Similarly, we have

Theorem 11.6 The output stability region is a union of Λ-like sets and Ψ-like sets.

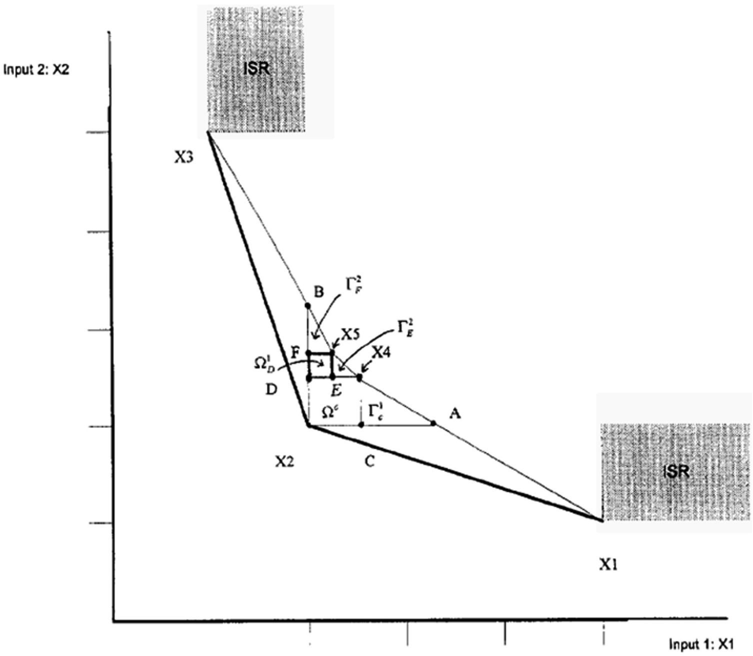


Fig. 11.1 Geometrical presentation of input stability region

11.2.3 Geometrical Presentation of Input Stability Region

We now illustrate the sensitivity analysis procedure geometrically for the following five DMUs with a single output and two inputs. For convenience, we suppose the five DMUs produce an equal amount of output and thus omit the output quantities in the following discussion. With the help of Fig. 11.1, we will see how to keep track of newly generated points (DMUs) by the procedure.

It is obvious that DMUs 1, 2, and 3 are extreme efficient, and DMUs 4 and 5 are inefficient. Let $DMU_o = DMU2(x_{10} = 2, x_{20} = 2)$, i.e., we consider the robustness of the efficiency of DMU2 when the two inputs increase.

Initiation (t=0) First we solve model (11.3) for DMU_o (point X_2), that is

$$
\begin{aligned}
&\beta_1^{o*} = \min \beta_1^o \\
&\text{subject to} \\
&5\lambda_1 + \lambda_3 + \frac{5}{2}\lambda_4 + \frac{9}{4}\lambda_5 \le 2\beta_1^o \\
&\lambda_1 + 5\lambda_3 + \frac{5}{2}\lambda_4 + \frac{11}{4}\lambda_5 \le 2 \\
&\lambda_1 + \lambda_3 + \lambda_4 + \lambda_5 \ge 1 \\
&\lambda_1, \lambda_3, \lambda_4, \lambda_5, \beta_1^o \ge 0
\end{aligned}
$$

	A	B	C	D	E	F
1	**DMUs**	**input 1**	**input 2**		**output**	**λ**
2	**1**	**5**	**1**		**1**	**0.333333**
3	**2**	**2**	**2**		**1**	**0**
4	**3**	**1**	**5**		**1**	**0**
5	**4**	**2 1/2**	**2 1/2**		**1**	**0.666667**
6	**5**	**2 1/4**	**2 3/4**		**1**	**0**
7		**DMU under**				
8		**consideration**		2	β	
9	Constraints				**1 2/3**	
10	input1	3.333333	≤	3 1/3	**point A**	
11	input2	2	≤	2		
12	output	1	≥	1		
13	DMUo	0	=	0		

Fig. 11.2 Spreadsheet for input stability region (Input 1)

Figure 11.2 shows the spreadsheet model. Cell D8 is reserved to indicate the DMU under consideration, and is equal to 2 now, indicating DMU2. Cells F2:F6 are reserved to indicate λ_j. Cell E9 represents β_1. This spreadsheet model is developed from the spreadsheet model for measure-specific models. The formulas for the spreadsheet shown in Fig. 11.2 are

Cell B10=SUMPRODUCT(B2:B6,F2:F6)
Cell B11=SUMPRODUCT(C2:C6,F2:F6)
Cell B12=SUMPRODUCT(E2:E6,F2:F6)
Cell B13=INDEX(F2:F6,D8,1)
Cell D10=E9*INDEX(B2:B6,D8,1)
Cell D11=INDEX(C2:C6,D8,1)
Cell D12=INDEX(E2:E6,D8,1)
Cell D13=0

Figure 11.3 shows the Solver parameters for the spreadsheet shown in Fig. 11.2. We have $\beta_1^{o*} = 5/3$ (see cell E9 in Fig. 11.2). For k = 2, we have the spreadsheet model shown in Fig. 11.4. The formulas for cells B10:B13 and cells D12:D13 remain the same. We need to change the formulas in cells D10:D11 to

Cell D10=INDEX(B2:B6,D8,1)
Cell D11=E9*INDEX(C2:C6,D8,1)

Using the Solver parameters shown in Fig. 11.3, we obtain $\beta_2^{o*} = 8/5$ (see cell E9 in Fig. 11.4). Furthermore, we have the following two newly generated points associated with the optimal β values (cells D10:D11 in Figs. 11.2 and 11.4)

$$\begin{cases} A = (\beta_1^{o*} x_{10},\ x_{20}) = (\dfrac{10}{3}, 2) \\ B = (x_{10},\ \beta_2^{o*} x_{20}) = (2, \dfrac{16}{5}) \end{cases}$$

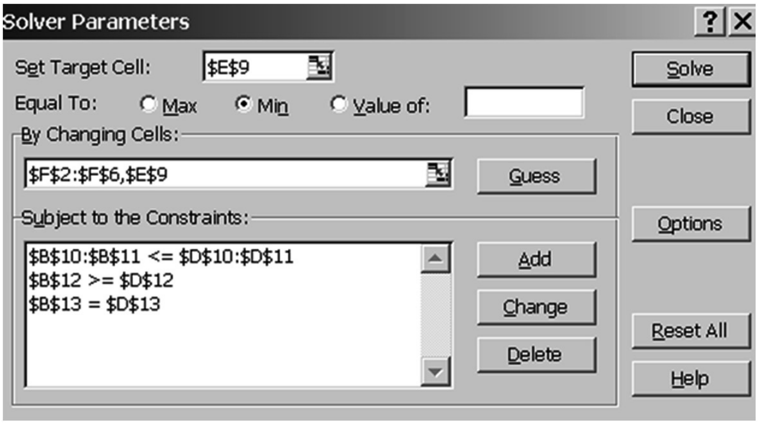


Fig. 11.3 Solver parameters for input stability region

	A	B	C	D	E	F
1	DMUs	input 1	input 2		output	λ
2	1	5	1		1	0
3	2	2	2		1	0
4	3	1	5		1	0.2
5	4	2 1/2	2 1/2		1	0
6	5	2 1/4	2 3/4		1	0.8
7						
8		DMU under consideration		2	β	
9	Constraints				1 3/5	
10	input1	2	≤	2		
11	input2	3.2	≤	3 1/5	Point B	
12	output	1	≥	1		
13	DMUo	0	=	0		

Fig. 11.4 Spreadsheet for input stability region (Input 2)

Obviously, the input hyperplane (line segment AB) constructed by A and B is dominated by DMU4 and DMU5. Thus, we solve model (11.5) for DMU_o, that is

$$\min \rho_1^o + \rho_2^o$$

subject to

$$5\lambda_1 + \lambda_3 + \frac{5}{2}\lambda_4 + \frac{9}{4}\lambda_5 \le 2\rho_1^o$$

$$\lambda_1 + 5\lambda_3 + \frac{5}{2}\lambda_4 + \frac{11}{4}\lambda_5 \le 2\rho_2^o$$

$$\lambda_1 + \lambda_3 + \lambda_4 + \lambda_5 \ge 1$$

$$\lambda_1, \lambda_3, \lambda_4, \lambda_5 \ge 0, \rho_1^o, \rho_2^o \ge 1$$

	A	B	C	D	E	F	G
1	DMUs	input 1	input 2		output	λ	
2	1	5	1		1	0	
3	2	2	2		1	0	
4	3	1	5		1	0	
5	4	2 1/2	2 1/2		1	1	
6	5	2 1/4	2 3/4		1	0	
7							
8		DMU under consideration		2	ρ1	ρ2	
9	Constraints				1 1/4	1 1/4	2 1/2
10	input1	2.5	≤	2 1/2			
11	input2	2.5	≤	2 1/2		Target cell	
12	output	1	≥	1			
13	DMUo	0	=	0			

Fig. 11.5 Spreadsheet for input stability region (Model (11.5))

Figure 11.5 shows the spreadsheet model for model (11.5). Cell E9 and Cell F9 represent ρ_1 and ρ_2, respectively. The target cell G9 (=E9+F9) represents the objective function of model (11.5). We change the formulas of cells D10:D11 to

Cell D10=E9*INDEX(B2:B6,D8,1)
Cell D11=F9*INDEX(C2:C6,D8,1)

Figure 11.6 shows the Solver parameters for model (11.5). We obtain $\rho_1^{o*} = \rho_2^{o*} = 5/4$. Moreover, we have $\Omega^o = \{(\beta_1, \beta_2) \mid 1 \le \beta_1 \le 5/4,\ 1 \le \beta_2 \le 5/4\}$ as shown in Fig. 11.1 and obtain the following two additional points associated with optimal ρ values

$$\begin{cases} C = (\rho_1^{o*} x_{10},\ x_{20}) = (x_{10}^C,\ x_{20}^C) = (5/2,\ 2) \\ D = (x_{10},\ \rho_2^{o*} x_{20}) = (x_{10}^D,\ x_{20}^D) = (2,\ 5/2) \end{cases}$$

Iteration: t=1 For the first point C, we solve model (11.3)

$$\begin{aligned} &\min \beta_1^C \\ &\text{subject to} \\ &5\lambda_1 + \lambda_3 + \frac{5}{2}\lambda_4 + \frac{9}{4}\lambda_5 \le \beta_1^C x_{10}^C = \frac{5}{2}\beta_1^C \\ &\lambda_1 + 5\lambda_3 + \frac{5}{2}\lambda_4 + \frac{11}{4}\lambda_5 \le x_{20}^C = 2 \\ &\lambda_1 + \lambda_3 + \lambda_4 + \lambda_5 \ge 1 \\ &\lambda_1, \lambda_3, \lambda_4, \lambda_5, \beta_1^C \ge 0 \end{aligned}$$

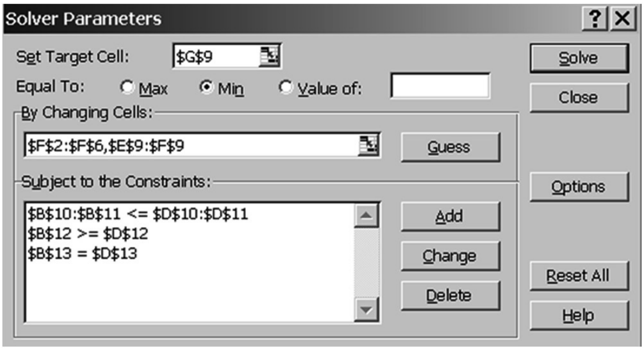


Fig. 11.6 Solver parameters for model (11.5)

We have $\beta_1^{C*} = 4/3$. Similarly, $\beta_2^{C*} = 5/4$. The two corresponding new points are as follows

$$\begin{cases} A \;=\; (\beta_1^{C*} x_{10}^C,\; x_{20}^C) = (\frac{10}{3},\; 2) \\ X_4 = (x_{10}^C,\; \beta_2^{C*} x_{20}^C) = (\frac{5}{2},\; \frac{5}{2}) \end{cases}$$

The input hyperplane constructed by these two points (line segment AX_4) is not dominated by other DMUs, therefore the iteration for point C stops and we have the following results.

Let $\hat{x}_{10} = c_1 x_{10}^C = c_1 \beta_1^{o*} x_{10}$ and $\hat{x}_{20} = c_2 x_{20}^C = c_2 x_{20}$. By Zhu (1996), we have

$$\Gamma_c^1 = \{(c_1, c_2) \mid\; 1 \le c_k \le \beta_k^{C*}, k = 1,2 \;\text{ and }\; B_1^C c_1 + B_2^C c_2 \le 1\}$$

where B_1^C and B_2^C are determined as follows

$$\begin{cases} B_1^C \beta_1^{C*} + B_2^C = 1 \\ B_1^C + B_2^C \beta_2^{C*} = 1 \end{cases} \Rightarrow \begin{cases} \frac{4}{3} B_1^C + B_2^C = 1 \\ B_1^C + \frac{5}{4} B_2^C = 1 \end{cases} \Rightarrow \begin{cases} B_1^C = \frac{3}{8} \\ B_2^C = \frac{1}{2} \end{cases}$$

Hence,

$$\Gamma_c^1 = \{(c_1 \rho_1^{o*}, c_2) \mid \rho_1^{o*} \le c_1 \rho_1^{o*} \le \rho_1^{o*} \beta_1^{C*} = \beta_1^{o*}, 1 \le c_2 \le \beta_2^{C*}, \text{ and } \frac{3}{8} \rho_1^{o*} c_1 + \frac{1}{2} \rho_1^{o*} c_2 \le \rho_1^{o*}\}$$

Let $\beta_1 = c_1\rho_1^{o*}$ and $\beta_2 = c_2$. Then $\Gamma_c^1 = \{(\beta_1, \beta_2) \mid 5/4 \le \beta_1 \le 5/3, 1 \le \beta_2 \le 5/4, 3\beta_1 + 5\beta_2 \le 10\}$.

Next, for the second point D, solving model (11.3) when k=1 and k=2 yields $\beta_1^{D*} = 5/4$ and $\beta_2^{D*} = 32/25$, respectively. Associated with these two optimal β values, we have two new points

$$\begin{cases} X_4 = (\beta_1^{D*}x_{10}^D,\ x_{20}^D) = (\frac{5}{2},\ \frac{5}{2}) \\ B = \ (x_{10}^D,\ \beta_2^{D*}x_{20}^D) = (2,\ \frac{16}{5}) \end{cases}$$

The input hyperplane determined by these two points (X_4 and B) is dominated by DMU5, therefore we compute model (11.5) for point D,

$$\begin{aligned} &\min \rho_1^D + \rho_2^D \\ &\text{subject to} \\ &5\lambda_1 + \ \lambda_3 + \frac{5}{2}\lambda_4 + \frac{9}{4}\lambda_5 \le \rho_1^D x_{10}^D = 2\rho_1^D \\ &\lambda_1 + 5\lambda_3 + \frac{5}{2}\lambda_4 + \frac{11}{4}\lambda_5 \le \rho_2^D x_{20}^D = \frac{5}{2}\rho_2^D \\ &\lambda_1 + \ \lambda_3 + \ \lambda_4 + \ \lambda_5 \ge 1 \\ &\lambda_1, \lambda_3, \lambda_4, \lambda_5 \ge 0, \rho_1^D, \rho_2^D \ge 1 \end{aligned}$$

We have $\rho_1^{D*} = 9/8$ and $\rho_2^{D*} = 11/10$. Next we compute Ω_D^1.

First, let $\hat{x}_{10} = d_1 x_{10}^D = d_1 x_{10}$ and $\hat{x}_{20} = d_2 x_{20}^D = d_2\rho_2^{o*}x_{20}$. By Theorem 11.1, we have

$$\Omega_D^1 = \{(d_1, d_2) \mid 1 \le d_k \le \rho_k^{D*}, k = 1, 2\}$$

and

$$\Omega_D^1 = \{(d_1, \rho_2^{o*}d_2) \mid 1 \le d_1 \le \rho_1^{D*}, \rho_2^{o*} \le \rho_2^{o*}d_2 \le \rho_2^{o*}\rho_1^{D*}\}$$

and further $\Omega_D^1 = \{(\beta_1, \beta_2) \mid 1 \le \beta_1 \le 9/8, 5/4 \le \beta_2 \le 11/8\}$.

Associated with the two optimal values of ρ_1^{D*} and ρ_2^{D*}, we now have the following two points

$$\begin{cases} E = (\rho_1^{D*}x_{10}^D, x_{20}^D) = (x_{10}^E, x_{20}^E) = (\frac{9}{4}, \frac{5}{2}) \\ F = (x_{10}^D, \rho_2^{D*}x_{20}^D) = (x_{10}^F, x_{20}^F) = (2, \frac{11}{4}) \end{cases}$$

Iteration t = 2 For the point E generated from the point D in the first iteration, we obtain, by solving model (11.3), $\beta_1^{E*} = 10/9$, $\beta_2^{E*} = 11/10$, and two corresponding points

$$\begin{cases} X_4 = (\beta_1^{E*} x_{10}^E, x_{20}^E) = (\frac{5}{2}, \frac{5}{2}) \\ X_5 = (x_{10}^E, \beta_2^{E*} x_{20}^E) = (\frac{9}{4}, \frac{11}{4}) \end{cases}$$

The input hyperplane constructed by the two points of X_4 and X_5 is not dominated by other DMUs, therefore the iteration stops.

Let $\hat{x}_{10} = e_1 x_{10}^E = e_1 \rho_1^{D*} x_{10}^D = e_1 \rho_1^{D*} x_{10}$ and $\hat{x}_{20} = e_2 x_{20}^E = e_2 x_{20}^D = e_2 \rho_2^{o*} x_{20}$. Similar to Γ^o, we have $\Gamma_E^2 = \left\{(e_1, e_2) \mid 1 \le e_k \le \beta_k^{E*}, \text{ k} = 1,2 \text{ and } B_1^E e_1 + B_2^E e_2 \le 1\right\}$ in which B_1^E and B_2^E are determined by

$$\begin{cases} B_1^E \beta_1^{E*} + B_2^E = 1 \\ B_1^E + B_2^E \beta_2^{E*} = 1 \end{cases} \Rightarrow \begin{cases} \frac{10}{9} B_1^E + B_2^E = 1 \\ B_1^E + \frac{11}{10} B_2^E = 1 \end{cases} \Rightarrow \begin{cases} B_1^E = \frac{9}{20} \\ B_2^E = \frac{1}{2} \end{cases}$$

Thus, $\Gamma_E^2 = \{(e_1 \rho_1^{D*}, e_2 \rho_2^{o*}) \mid \rho_1^{D*} \le e_1 \rho_1^{D*} \le \rho_1^{D*} \beta_1^{E*} = \rho_1^{o*}, \rho_2^{o*} \le e_2 \rho_2^{o*} \le \beta_2^{E*} \rho_2^{o*}$, and $(9/20)\rho_1^{D*} \rho_2^{o*} e_1 + (1/2) \rho_1^{D*} \rho_2^{o*} e_2 \le \rho_1^{D*} \rho_2^{o*}\} = \left\{(\beta_1, \beta_2) \mid 9/8 \le \beta_1 \le 5/4, 5/4 \le \beta_2 \le 11/8, 18\beta_1 + 2\beta_2 \le 45\right\}$, where $\beta_1 = e_1 \rho_1^{D*}$ and $\beta_2 = e_2 \rho_2^{o*}$.

For the point F, we have $\beta_1^{F*} = 9/8$ and $\beta_2^{F*} = 64/55$, and two corresponding points

$$\begin{cases} X_5 = (\beta_1^{F*} x_{10}^F, x_{20}^F) = (\frac{9}{4}, \frac{11}{4}) \\ B = (x_{10}^F, \beta_2^{F*} x_{20}^F) = (2, \frac{16}{5}) \end{cases}$$

The input hyperplane constructed by these two points of X_5 and B is not dominated by other DMUs, therefore the iteration stops.

Let $\hat{x}_{10} = f_1 x_{10}^F = f_1 x_{10}$ and $\hat{x}_{20} = f_2 x_{20}^F = f_2 \rho_2^{D*} x_{20}^D = f_2 \rho_2^{D*} \rho_2^{o*} x_{20}$. Similar to Γ^o, we have $\Gamma_F^2 = \{(f_1, f_2) \mid 1 \le f_k \le \beta_k^{F*}, k = 1,2 \text{ and } B_1^F f_1 + B_2^F f_2 \le 1\}$ in which B_1^F and B_2^F are the solutions to the following system of equations

$$\begin{cases} B_1^F \beta_1^{F*} + B_2^F = 1 \\ B_1^F + B_2^F \beta_2^{F*} = 1 \end{cases} \Rightarrow \begin{cases} \frac{9}{8} B_1^F + B_2^F = 1 \\ B_1^F + \frac{64}{55} B_2^F = 1 \end{cases} \Rightarrow \begin{cases} B_1^F = \frac{9}{17} \\ B_2^F = \frac{55}{136} \end{cases}$$

Table 11.1 DMUs for illustration of input stability region

DMU	$1(X_1)$	$2(X_2)$	$3(X_3)$	$4(X_4)$	$5(X_5)$
input 1 x_1	5	2	1	5/2	9/4
input 2 x_2	1	2	5	5/2	11/4

Thus, $\Gamma_F^2 = \{(\beta_1, \beta_2) \mid 1 \le \beta_1 \le 9/8,\ \ 11/8 \le \beta_2 \le 8/5,\ \ 9\beta_1 + 5\beta_2 \le 17\}$. Finally, we obtain the following input stability region for DMU2 (point X_2) as shown in Fig. 11.1.

$$ISR = \Omega^o \cup \Omega_D^1 \cup \Gamma_C^1 \cup \Gamma_E^2 \cup \Gamma_F^2$$

11.3 Infeasibility and Stability

The previous sensitivity analysis procedure is developed under the assumption that model (11.3) (or model (11.7)) is feasible. However, this may not be always the case. For example, if we calculate (11.3) for DMU3 in Table 11.1, then we have $\beta_1^* = 2$ for the first input but infeasibility for the second input. If we calculate (11.3) for DMU1, then we have infeasibility for the first input. Figure 11.7 presents the results for the three efficient DMUs 1, 2, and 3. The calculation is performed by a VBA procedure “InputStabilityRegion”.

```
Sub InputStabilityRegion()
Dim i As Integer
For i = 1 To 3
Range("D8") = i
SolverSolve UserFinish:=True
If SolverSolve(UserFinish:=True) = 5 Then
Range("G" & i + 1) = "Infeasible"
Else
Range("G" & i + 1) = Range("E9")
End If
Next
End Sub
```

Note that, in fact, we can increase infinitely the amount of DMU3’s second input (DMU1’s first input) while maintaining the efficiency of DMU3.

Theorem 11.7 For an efficient DMU_o, an increase of the kth input only, model (11.3) is infeasible, *if and only if*, the amount of kth input of DMU_o can be increased without limitation while maintaining the efficiency of DMU_o.

	A	B	C	D	E	F	G
1	**DMUs**	**input 1**	**input 2**		**output**	λ	**ISR**
2	**1**	**5**	**1**		**1**	**0**	Infeasible
3	**2**	**2**	**2**		**1**	**1**	1 2/3
4	**3**	**1**	**5**		**1**	**0**	2
5	**4**	**2 1/2**	**2 1/2**		**1**	**0**	
6	**5**	**2 1/4**	**2 3/4**		**1**	**0**	
7							
8				3	β		
9	Constraints				**2**		
10	input1	2	≤	2			
11	input2	2	≤	5	Input1		
12	output	1	≥	1			
13	DMUo	0	=	0			

	A	B	C	D	E	F	G
1		**input 1**	**input 2**		**output**	λ	**ISR**
2	**1**	**5**	**1**		**1**	**0**	2
3	**2**	**2**	**2**		**1**	**0.5**	1 3/5
4	**3**	**1**	**5**		**1**	**0**	Infeasible
5	**4**	**2 1/2**	**2 1/2**		**1**	**0**	
6	**5**	**2 1/4**	**2 3/4**		**1**	**0**	
7							
8				3	β		
9	Constraints				**1/5**		
10	input1	1	≤	1			
11	input2	1	≤	1	Input2		
12	output	0.5	≥	1			
13	DMUo	0	=	0			

Fig. 11.7 Optimal β

[Proof]: The *if* part is obvious from the fact that if (11.3) is feasible, then the optimal value to (11.3) gives the maximum increase proportion of the kth input. Therefore, the amount of kth input cannot be infinitely increased.

To establish the *only if* part we suppose that the kth input is increased by $M \geq 1$ and DMU_o is inefficient. By substituting DMU_o into CRS envelopment model, we obtain an optimal solution $\theta^* \leq 1$, $\lambda_o^* = 0, \lambda_j^* (j \neq o)$, in which $\theta^* \leq 1$ implies $DMU_o \in F$. Therefore,

$$\begin{cases} \sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j^* x_{kj} \leq \theta^* M x_{ko} \\ \sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j^* x_{ij} \leq \theta^* x_{io} \leq x_{io} \qquad i \neq k \end{cases}$$

This means that $\lambda_j^*(j \neq o), \beta_k = \theta^* M$ is a feasible solution to (11.3) and leads to a contradiction. Since M is arbitrary, the amount of the kth input can be infinitely increased while maintaining DMU_o's efficiency. ■

As the Theorem 11.7 indicates, if (11.3) is infeasible, then $\beta_k^{o*} = +\infty$. Thus, in this situation, we must modify the sensitivity analysis procedure. Because we are unable to express the new frontier point associated with $\beta_k^{o*} = +\infty$, and further, to apply the stopping rule. Note that if we here assume that all data are positive, then model (11.5) is always feasible. But in the case of infeasibility, (11.5) does not perform well. For example, if we apply (11.5) to DMU3, we obtain $\rho_1^* = 2$ and $\rho_2^* = 1$. i.e., $\rho_i^{o*} = 1$ relative to the unbounded input i. Consequently, we are unable to determine the stability region. Thus, from a computational point of view, in this situation, we apply model (11.5) with $\rho_i^o = \theta_o$ $(i = 1, \ldots, m)$. That is,

$$
\begin{aligned}
&\min \theta_o \\
&\text{subject to} \\
&\sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j x_{ij} \leq \theta_o x_{io} \qquad i = 1, \ldots, m \\
&\sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r = 1, \ldots, s \\
&\lambda_j \geq 0 \qquad j = 1, \ldots, n.
\end{aligned}
\tag{11.9}
$$

At each point, in each iteration, we first apply (11.9) when (11.3) is infeasible, and then, for the newly generated points, we apply (11.3). If (11.3) is feasible, we use the procedure suggested previously. If (11.3) is still infeasible, then apply (11.9) again (go to next iteration). We can, in fact, regard infeasibility as the rejection of the stopping rule, and then we calculate model (11.9) instead of (11.5) to generate new frontier DMUs for the next iteration. In this situation, the set Ω^o obtained from (11.9) corresponds to the $\infty - norm$ in Charnes et al. (1992).

This general procedure for the infeasibility case is stated below

Step 1: Solve model (11.9).
Step 2: Solve (11.3) for the newly generated points by (11.9):

$DMU(\theta_o^*) = (x_{1o}, \ldots, \theta_o^* x_{ko}, \ldots, x_{mo}, y_{1o}, \ldots, y_{so}),\ k = 1, \ldots, m.$

a. If (11.3) is feasible, then go to the procedure given in Sect. 11.2.1;
b. If (11.3) is infeasible, then go to step 1.

Note that infeasibility often occurs in real world situations. In theory, one can always use this general procedure to determine the ISR. However, in practice one may use this procedure to approximate the ISR due to the fact that some inputs' amount can be infinitely increased. For example, it is obvious that the IRS for DMU3 in Table 11.1 is

$$ISR = \{(\beta_1,\ \beta_2) \mid\ 1 \le \beta_1 < 2,\ \ 1 \le \beta_2 < +\infty\} \tag{11.10}$$

which is the shaded region shown in Fig. 11.1. Furthermore, we have

Theorem 11.8 For the two-input case, one of the two optimal β values in (11.3) is equal to the corresponding optimal value to (11.9), *if and only if*, $ISR = \{(\beta_1,\ \beta_2) \mid\ 1 \le \beta_i < \beta_i^{o*}, i = 1,2\}$, where one of the β_i^{o*} is finite and the other is $+\infty$.

[Proof]:Without loss of generality, assume that β_1^{o*} is finite and $\beta_2^{o*} = +\infty$.

Suppose that $ISR = \{(\beta_1, \beta_2) \mid 1 \le \beta_1 < \beta_1^{o*},\ \ 1 \le \beta_2 < +\infty\}$. Let $\beta_1 = \beta_1^{o*}$, then $DMU_o$with $(\beta_1^{o*} x_{1o},\ \beta_1^{o*} x_{2o})$ is a frontier point. Therefore $\theta_o^* = \beta_1^{o*}$. This completes the proof of the *if* part.

Suppose $\theta_o^* = \beta_1^{o*}$. Obviously, if DMU_o with inputs of $(\beta_1 x_{1o},\ \beta_2 x_{2o})$ is in set E, then $(\beta_1, \beta_2) \in ISR = \{(\beta_1, \beta_2) \mid 1 \le \beta_1 < \beta_1^{o*} = \theta_o^*,\ \ 1 \le \beta_2 < +\infty\}$. Next, note that the original DMU_o belongs to set E, therefore $\theta_o^* > 1$. By θ_o^* and β_1^{o*}, we obtain two frontier points $A = (\theta_o^* x_{1o},\ \theta_o^* x_{2o})$ and $B = (\beta_1^{o*} x_{1o},\ x_{2o}) = (\theta_o^* x_{1o},\ x_{2o})$. Thus,

$\theta_o^* = \beta_1^{o*} \Rightarrow A \in F$ with nonzero slack on the second input $\Rightarrow (\beta_1^{o*} x_{1o},\ \beta_2 x_{2o}) \in F$, where $1 \le \beta_2 < +\infty \Rightarrow (\beta_1 x_{1o},\ \beta_2 x_{2o}) \in E$, where $1 \le \beta_1 < \beta_1^{o*}, 1 \le \beta_2 < +\infty$.

Therefore, if $(\beta_1, \beta_2) \in ISR$, then DMU_o preserves its efficiency. This completes the *only if* part. ■

By the proof of Theorem 11.8 and the result of Theorem 11.7, we can easily obtain

Corollary 11.1 For the two-input case, if one of the two optimal β values in (11.3) is equal to the corresponding optimal value to (11.9), then (11.3) is infeasible for the other input.

Note that equality is not held in the right hand side of the inequalities in (11.10) of β_i. Otherwise, DMU_o will be in set F. For instance, if $\beta_1 = 2$ in (11.10), then DMU3 (X3) is moved into set F. However, if we only consider weak efficiency, then the equality can be imposed. Because the efficiency ratings are equal to one for the DMUs in set F.

Finally, the above discussion and development holds for the output case when (11.7) is infeasible. That is,

Theorem 11.9 For an efficient DMU_o, an increase of the *k*th output only, model (11.7) is infeasible, *if and only if*, the amount of *k*th output of DMU_o can be increased without limitation while maintaining the efficiency of DMU_o.

Theorems 11.7 and 11.9 indicate that if model (11.3) or (11.7) is infeasible, then the test DMU remains efficient when data variations are applied to the specific input or output. This conclusion is also true when the data variations are applied to both the test DMU and the remaining DMUs.

11.4 Simultaneous Data Change[2]

Zhu (2001) shows that a particular super-efficiency score can be decomposed into two data perturbation components of a particular test DMU and the remaining DMUs. Also, necessary and sufficient conditions for preserving a DMU's efficiency classification are developed when various data changes are applied to all DMUs. As a result, DEA sensitivity analysis can be easily applied if we employ various super-efficiency DEA models.

We rewrite the input-oriented CRS envelopment model and its dual as

$$\begin{aligned}
&\theta_{CRS}^{o*} = \min \theta_{CRS}^{o} \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = \theta_{CRS}^{o} x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \qquad r = 1,2,\ldots,s; \\
&\theta_{CRS}^{o},\ \lambda_j, s_i^-, s_r^+ \geq 0.
\end{aligned} \tag{11.11}$$

$$\begin{aligned}
&\max \sum_{r=1}^{s} u_r y_{ro} \\
&\text{subject to} \\
&\sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0, \qquad j = 1,\ldots,n \\
&\sum_{i=1}^{m} v_i x_{io} = 1 \\
&u_r, v_i \geq 0.
\end{aligned} \tag{11.12}$$

We also present the input-oriented and output-oriented CRS super-efficiency models

$$\begin{aligned}
&\theta_o^{\text{super}*} = \min \theta_o^{\text{super}} \\
&\text{subject to} \\
&\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \leq \theta_o^{\text{super}} x_{io} \qquad i = 1,2,\ldots,m \\
&\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r = 1,2,\ldots,s \\
&\theta_o^{\text{super}}, \lambda_j (j \neq o) \geq 0.
\end{aligned} \tag{11.13}$$

[2] Part of the material in this section is adapted from European Journal of Operational Research, Vol 129, Zhu, J., Super-efficiency and DEA Sensitivity Analysis, 443–455, 2001, with permission from Elsevier Science.

$$\begin{aligned}
&\phi_o^{\text{super}*} = \max\ \phi_o^{\text{super}} \\
&\text{subject to} \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j y_{rj} \geq \phi_o^{\text{super}} y_{ro} \quad r=1,\ldots,s \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j x_{ij} \leq x_{io} \quad i=1,\ldots,m \\
&\phi_o^{\text{super}},\ \lambda_j (j\neq o) \geq 0.
\end{aligned} \tag{11.14}$$

In order to simultaneously consider the data changes for other DMUs, we suppose increased output and decreased input for all other DMUs. i.e., our discussion is based on a worst-case scenario in which efficiency of DMU_o declines and the efficiencies of all other $DMU_j (j \neq o)$ improve.

Let **I** and **O** denote respectively the input and output subsets in which we are interested. i.e., we consider the data changes in set **I** and set **O**. Then the simultaneous data perturbations in input/output of all $DMU_j (j \neq o)$ and DMU_o can be written as

Percentage data perturbation (variation)

For DMU_o

$$\begin{cases} \hat{x}_{io} = \delta_i x_{io} & \delta_i \geq 1,\ i \in \mathbf{I} \\ \hat{x}_{io} = x_{io} & i \notin \mathbf{I} \end{cases} \text{ and } \begin{cases} \hat{y}_{ro} = \tau_r y_{ro} & 0 < \tau_r \leq 1,\ r_r \in \mathbf{O} \\ \hat{y}_{ro} = y_{ro} & r \notin \mathbf{O} \end{cases}$$

For $DMU_j (j \neq o)$

$$\begin{cases} \hat{x}_{ij} = x_{ij} / \tilde{\delta}_i & \tilde{\delta}_i \geq 1,\ i \in \mathbf{I} \\ \hat{x}_{ij} = x_{ij} & i \notin \mathbf{I} \end{cases} \text{ and } \begin{cases} \hat{y}_{rj} = y_{rj} / \tilde{\tau}_r & 0 < \tilde{\tau}_r \leq 1,\ r \in \mathbf{O} \\ \hat{y}_{rj} = y_{rj} & r \notin \mathbf{O} \end{cases}$$

where (^) represents adjusted data. Note that the data perturbations represented by δ_i and $\tilde{\delta}_i$ (or τ_r and $\tilde{\tau}_r$) can be different for each $i \in \mathbf{I}$ (or $r \in \mathbf{O}$).

Lemma 11.2 Suppose $DMU_o \in$ set F with non-zero input/output slack values associated with set **I**/set **O**. Then DMU_o with inputs of $\hat{x}_{io}$ and outputs of $\hat{y}_{ro}$ as defined above still belongs to set F when other DMUs are fixed.

[Proof]: Applying the complementary slackness theorem for models (11.10) and (11.11), we have $s_i^{-*} v_i^* = s_r^{+*} u_r^* = 0$. Since $s_i^{-*} \neq 0$ for $i \in \mathbf{I}$ and $s_r^{+*} \neq 0$ for $r \in \mathbf{O}$, we have $v_i^* = 0$ for $i \in \mathbf{I}$ and $u_r^* = 0$ for $r \in \mathbf{O}$. Therefore, v_i^* and u_r^* is a feasible solution to (11.11) for DMU_o with inputs of $\hat{x}_{io}$ and outputs of $\hat{y}_{ro}$. Note that $\sum_{r=1}^{s} u_r^* \hat{y}_{ro} = \sum_{r \notin \mathbf{O}} u_r^* \hat{y}_{ro} = \sum_{r \notin \mathbf{O}} u_r^* y_{ro} = 1$ indicating that the maximum value of 1 is achieved. Therefore, DMU_o still belongs to set F. ■

11.4.1 Sensitivity Analysis Under CRS

We first modify models (11.12) and (11.13) to the following two super-efficiency DEA models that are based upon the measure-specific models

$$
\begin{aligned}
&\theta_{\mathbf{I}}^{o*} = \min \theta_{\mathbf{I}}^{o} \\
&\text{subject to} \\
&\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \leq \theta_{\mathbf{I}}^{o} x_{io} \qquad i \in \mathbf{I} \\
&\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \leq x_{io} \qquad i \notin \mathbf{I} \\
&\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r = 1,2,\ldots,s \\
&\theta_{\mathbf{I}}^{o}, \lambda_j (j \neq o) \geq 0.
\end{aligned}
\tag{11.15}
$$

and

$$
\begin{aligned}
&\phi_{\mathbf{O}}^{o*} = \max \phi_{\mathbf{O}}^{o} \\
&\text{subject to} \\
&\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \geq \phi_{\mathbf{O}}^{o} y_{ro} \qquad r \in \mathbf{O} \\
&\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r \notin \mathbf{O} \\
&\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \leq x_{io} \qquad i=1,\ldots,m \\
&\phi_{\mathbf{O}}^{o}, \lambda_j (j \neq o) \geq 0.
\end{aligned}
\tag{11.16}
$$

If $\mathbf{I} = \{k\}$, $k \in \{1, \ldots, m\}$ and $\mathbf{O} = \{l\}$, $l \in \{1, \ldots, s\}$, then optimal values of $\theta_{\mathbf{I}}^{o*} = \theta_k^{o*} (k = 1,\ldots,m)$ and $\phi_{\mathbf{O}}^{o*} = (\phi_l^{o*})(l = 1,\ldots,s)$ are the optimal values to models (11.3) and (11.7), respectively.

Models (11.14) and (11.15) measure the maximum increase rate of inputs associated with **I** and the maximum decrease rate of outputs associated with **O**, respectively, required for DMU_o to reach the frontier of $DMU_j (j \neq o)$ when other inputs and outputs are kept at their current levels. For example, consider B in Table 11.2 (Fig. 11.8) and model (11.14). If $\mathbf{I} = \{1\}$, $\theta_{\{1\}}^{B*} = 14/9$ indicates that B

Table 11.2 Sample DMUs

DMU	y	x_1	x_2	$\theta_o^{\text{super}*} = \theta_{\mathbf{I}=\{1,2\}}^{o*}$	$\theta_{\mathbf{I}=\{1\}}^{o*}$	$\theta_{\mathbf{I}=\{2\}}^{o*}$
A	1	2	5	15/13	5/4	7/5
B	1	3	3	26/21	14/9	17/12
C	1	6	2	3/2	infeasible	3/2
D	1	2	7	1	1	5/7

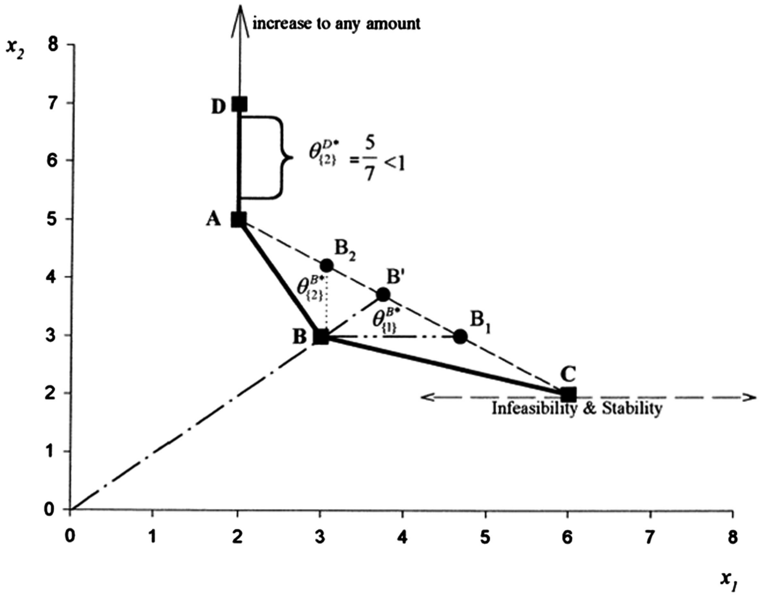


Fig. 11.8 Super-efficiency and sensitivity analysis

reaches B_1 by changing its x_1 to $(14/9)\times 3 = 14/3$. If $\mathbf{I} = \{2\}$, $\theta_{\{2\}}^{B*} = 17/12$ indicates that B reaches B_2 by changing its x_2 to $(17/12)\times 3 = 17/14$. If $\mathbf{I} = \{1, 2\}$, $\theta_{\{1,2\}}^{B*} = \theta_B^{\text{super}*} = 26/21$ gives the input increase rate for B in order to reach B′.

Associated with the optimal values in models (11.12), (11.13), (11.14) and (11.15), we have

Lemma 11.3
1. If $\theta_o^{\text{super}} = 1$, then $\theta_{\mathbf{I}}^{o*} \le 1$.
2. If $\phi_o^{\text{super}*} = 1$, then $\phi_{\mathbf{O}}^{o*} \ge 1$.

[Proof]: The proof is obvious from the fact that $\theta_o^{\text{super}} = 1$ is a feasible solution to (11.4) and $\phi_o^{\text{super}*} = 1$ is a feasible solution to (11.5). ■

Lemma 11.4
1. If $\theta_o^{\text{super}} = 1$ and $\theta_{\mathbf{I}}^{o*} < 1$, then $DMU_o \in F$.
2. If $\phi_o^{\text{super}*} = 1$ and $\phi_{\mathbf{O}}^{o*} > 1$, then $DMU_o \in F$.

[Proof]: (i) $\theta_o^{\text{super}} = 1$ indicates that $DMU_o \in E' \cup F$. $\theta_{\mathbf{I}}^{o*} < 1$ further indicates that there are non-zero slack values in x_{io} for $i \in \mathbf{I}$. Thus, $DMU_o \in F$.

(ii) The proof is similar to that of (i). ■

Theorem 11.10

1. If $\theta_o^{\text{super}} = 1$ and $\theta_{\mathbf{I}}^{o*} < 1$, then for any $\delta_i \geq 1$ and $\tilde{\delta}_i \geq 1$ $(i \in \mathbf{I})$, DMU_o remains in set F.
2. If $\phi_o^{\text{super}*} = 1$ and $\phi_{\mathbf{O}}^{o*} > 1$, then for any $0 < \tau_r \leq 1$ and $0 < \tilde{\tau}_r \leq 1$ $(r \in \mathbf{O})$, DMU_o remains in set F.

[Proof]: (i) From Lemma 11.4, we know that $DMU_o \in$ F with non-zero slack values in x_{io} for $i \in \mathbf{I}$. Based upon Lemma 11.2 and the proof of Lemma 11.2, we know that for any $\delta_i \geq 1$ and $\tilde{\delta}_i \geq 1$, with an objective function value of 1, v_i^* and u_r^* is a feasible solution to (11.11) in which inputs are replaced by $\hat{x}_{ij}$ for $i \in \mathbf{I}$ and x_{ij} for $i \notin \mathbf{I}$. Thus, DMU_o remains in set F after input data changes set $\mathbf{I}$ in all DMUs.

(ii) The proof is similar to that of (i). ■

In fact, Lemma 11.2 and Theorem 11.10 indicate that the classification of DMUs in set F is stable under any data perturbations in all DMUs occurred in inputs (outputs) which have non-zero slack values in DMU_o. For example, if $\mathbf{I}$ = {2}, then model (11.14) yields $\theta_{\{2\}}^{D*} = 5/7 < 1$ for D indicating that D has non-zero slack value in its second input. From Fig. 11.8, it is clear that D can increase its x_2 to any amount and still belongs to set F while other DMUs, A, B and C decrease their amount of x_2. This finding is very useful for the sensitivity analysis of the DMUs in set F.

Theorem 11.10 gives the sufficient condition for $DMU_o \in$ set F to preserve its efficiency classification. By Lemma 11.3, we immediately have

Corollary 11.2

1. If for any $\delta_i \geq 1$ and $\tilde{\delta}_i \geq 1$ $(i \in \mathbf{I})$, DMU_o remains in set F, then (a) $\theta_o^{\text{super}} = 1$ and $\theta_{\mathbf{I}}^{o*} < 1$, or (b) $\theta_o^{\text{super}} = 1$ and $\theta_{\mathbf{I}}^{o*} = 1$.
2. If for any $0 < \tau_r \leq 1$ and $0 < \tilde{\tau}_r \leq 1$ $(r \in \mathbf{O})$, DMU_o remains in set F, then (a) $\phi_o^{\text{super}*} = 1$ and $\phi_{\mathbf{O}}^{o*} > 1$, or (b) $\phi_o^{\text{super}*} = 1$ and $\phi_{\mathbf{O}}^{o*} = 1$.

Corollary 11.2 implies that for $DMU_o \in$ set F, some inputs without slack values may also be increased while preserving the efficiency of DMU_o. For example, consider two DMUs: $\text{DMU}_1 = (y, x_1, x_2, x_3)$ and $\text{DMU}_2 = (y, x_1, x_2, \pi x_3)$, where $\pi > 1$, a constant. Obviously, $\text{DMU}_1 \in$ set E and $\text{DMU}_2 \in$ set F with non-zero slack value on the third input. Now, let $\mathbf{I}$ = {2, 3}. We have that DMU_2 with $(y, x_1, \delta x_2, \delta\pi x_3)$ $(\delta > 1)$ remains in set F while DMU_1 is changed to $(y, x_1, x_2/\tilde{\delta}, x_3/\tilde{\delta}) \in$ set E $(\tilde{\delta} > 1)$. In this situation, $\theta_{\{2,3\}}^{2*} = 1$ in (11.14).

From Lemma 11.3, we know that $\theta_{\mathbf{I}}^{o*}$ or $\phi_{\mathbf{O}}^{o*}$ may also be equal to one. Obviously, in this situation, $DMU_o \in$ set E′ or set F and our approach indicates that no data variations are allowed in DMU_o and other DMUs. In fact, any data perturbation defined above will change the efficiency classification of DMUs in set E′. Note also that $\theta_{\{1\}}^{D*} = 1$ for D in Table 11.2. Thus, any data variation in the first input will let D become non-frontier point (see Fig. 11.8).

Furthermore, from Lemma 11.3, we have

Corollary 11.3 Infeasibility of model (11.14) or model (11.15) can only be associated with extreme efficient DMUs in set E.

[Proof]: Lemma 11.3 implies that models (11.14) and (11.15) are always feasible for DMUs in set E′ or set F. Also, models (11.14) and (11.15) are always feasible for non-frontier DMUs. Therefore, infeasibility of models (11.14) and (11.15) may only occur for extreme efficient DMUs in set E. ■

Seiford and Zhu (1998b) show that infeasibility of a super-efficiency DEA model means stability of the efficiency classification of DMU_o with respect to the changes of corresponding inputs and (or) outputs in all DMUs. We summarize Seiford and Zhu's (1998b) finding as the following theorem.

Theorem 11.11

1. If a specific super-efficiency DEA model associated with set I is infeasible, if and only if for any $\delta_i \geq 1$ and $\tilde{\delta}_i \geq 1$ $(i \in \mathbf{I})$, DMU_o remains extreme efficient.
2. If a specific super-efficiency DEA model associated with set O is infeasible, if and only if for any $0 < \tau_r \leq 1$ and $0 < \tilde{\tau}_r \leq 1$ $(r \in \mathbf{O})$, DMU_o remains extreme efficient.

Theorem 11.11 indicates that, for example, if mode (11.14) is infeasible, then DMU_o will still be extreme efficient no matter how much its inputs associated with set **I** are increased while the corresponding inputs of other DMUs are decreased. Consider C in Table 11.2. If **I** = {1}, then model (11.14) is infeasible. (Note that model (11.12) is feasible for C.) From Fig. 11.8, it is clear that C will remain extreme efficient if its first input is increased to any amount while DMUs A, C, and D decrease their amount of x_1.

In the discussion to follow, we assume that super-efficiency DEA models (11.14) and (11.15) are feasible. Otherwise, the efficiency classification of DMU_o is stable to data perturbations in all DMUs by Theorem 11.11.

Lemma 11.5

1. If model (11.14) is feasible and $\theta_o^{\sup er} > 1$ then $\theta_{\mathbf{I}}^{o*} > 1$.
2. If model (11.15) is feasible and $\phi_o^{\sup er*} < 1$ then $\phi_{\mathbf{O}}^{o*} < 1$.

[Proof]: (i) Suppose $\theta_{\mathbf{I}}^{o*} \leq 1$. Then the input constraints of (11.14) turn into

$$\begin{cases} \sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j x_{ij} \leq \theta_{\mathbf{I}}^{o*} x_{io} \leq x_{io} & i \in \mathbf{I} \\ \sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j x_{ij} \leq x_{io} & i \notin \mathbf{I} \end{cases}$$

which indicates that $\theta_o^{\sup er} = 1$ is a feasible solution to (11.12). Therefore, $\theta_o^{\sup er*} \leq 1$. A contradiction. Thus, $\theta_{\mathbf{I}}^{o*} > 1$.

(ii) The proof is similar to that of (i). ■

Lemma 11.5 indicates that if $DMU_o \in$ set E and model (11.14) (or model (11.15)) is feasible, then $\theta_{\mathbf{I}}^{o*}$ must be greater than one (or $\phi_{\mathbf{O}}^{o*}$ must be less than one). We next study the efficiency stability of extreme efficient DMUs and we relax the assumption that same percentage change holds for data variation of DMU_o and $DMU_j\,(j \neq o)$ and generalize the results in Seiford and Zhu (1998b).

Theorem 11.12 Suppose $\theta_o^{\sup er*} > 1$ and $\phi_o^{\sup er*} < 1$, then

1. If $1 \leq \delta_i \tilde{\delta}_i < \theta_{\mathbf{I}}^{o*}$ for $i \in \mathbf{I}$, then DMU_o remains extreme efficient. Furthermore, if equality holds for $\delta_i \tilde{\delta}_i = \theta_{\mathbf{I}}^{o*}$, i.e., $1 \leq \delta_i \tilde{\delta}_i \leq \theta_{\mathbf{I}}^{o*}$, then DMU_o remains on the frontier, where $\theta_{\mathbf{I}}^{o*}$ is the optimal value to (11.14).
2. If $\phi_{\mathbf{O}}^{o*} < \tau_r \tilde{\tau}_r \leq 1$ for $r \in \mathbf{O}$, then DMU_o remains extreme efficient. Furthermore, if equality holds for $\tau_r \tilde{\tau}_r = \phi_{\mathbf{O}}^{o*}$, i.e., $\phi_{\mathbf{O}}^{o*} < \tau_r \tilde{\tau}_r \leq 1$, then DMU_o remains on the frontier, where $\phi_{\mathbf{O}}^{o*}$ is the optimal value to (11.15).

[Proof]: (i) Note that from Lemma 11.5, $\theta_{\mathbf{I}}^{o*} > 1$. Now suppose $1 \leq \delta_i^o \tilde{\delta}_i^o < \theta_{\mathbf{I}}^{o*}$, and DMU_o is not extreme efficient when $\hat{x}_{io} = \delta_i^o x_{io}$ and $\hat{x}_{ij} = x_{ij} / \tilde{\delta}_i^o$, $\theta_{\mathbf{I}}^{o*}$. Then, there exist $\lambda_j \;\; (j \neq o) \geq 0$ and $\theta_o^{\sup er*} \leq 1$ in (11.12) such that

$$\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j \frac{x_{ij}}{\tilde{\delta}_i^o} \leq \theta_o^{\sup er*} \delta_i^o x_{io} \qquad i \in \mathbf{I}$$

$$\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \leq \theta_o^{\sup er*} x_{io} \;\; \leq x_{io} \qquad i \notin \mathbf{I}$$

$$\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r = 1, 2, \ldots, s.$$

This means that $\lambda_j (j \neq o) \geq 0$ and $\theta_o^{\sup er*} \delta_i^o \tilde{\delta}_i^o$ is a feasible solution to (11.14). But $\theta_o^{\sup er*} \delta_i^o \tilde{\delta}_i^o < \theta_o^{\sup er*} \theta_{\mathbf{I}}^{o*} \leq \theta_{\mathbf{I}}^{o*}$ violating the optimality of $\theta_{\mathbf{I}}^{o*}$. Thus, if $1 \leq \delta_i^o \tilde{\delta}_i^o < \theta_{\mathbf{I}}^{o*}$, then DMU_o remains extreme efficient.

Next, if $\delta_i^o \tilde{\delta}_i^o = \theta_{\mathbf{I}}^{o*}$, then we assume DMU_o is not a frontier when $\hat{x}_{io} = \delta_i^o x_{io}$ and $\hat{x}_{ij} = x_{ij} / \tilde{\delta}_i^o$, $i \in \mathbf{I}$. Thus, we have $\theta_o^{\sup er*} < 1$ in (11.12). Now we have $\theta_o^{\sup er*} \delta_i^o \tilde{\delta}_i^o \leq \theta_o^{\sup er*} \theta_{\mathbf{I}}^{o*} < \theta_{\mathbf{I}}^{o*}$ violating the optimality of $\theta_{\mathbf{I}}^{o*}$. Thus, if $1 \leq \delta_i^o \tilde{\delta}_i^o \leq \theta_{\mathbf{I}}^{o*}$, then DMU_o remains on the frontier.

(ii) The proof is similar to (i), but is based upon (11.13) and (11.15). ■

Theorem 11.12 indicates that the optimal value to a super-efficiency DEA model can actually be decomposed into a data perturbation component (e) for DMU_o and a data perturbation component ($\tilde{e}$) for the remaining DMUs, $DMU_j \; (j \neq o)$. Define

$$\Omega^o = \begin{cases} \theta_{\mathbf{I}}^{o*} & \text{if } e = \delta \text{ and } \tilde{e} = \tilde{\delta} \\ \phi_{\mathbf{O}}^{o*} & \text{if } e = \tau \text{ and } \tilde{e} = \tilde{\tau} \end{cases}$$

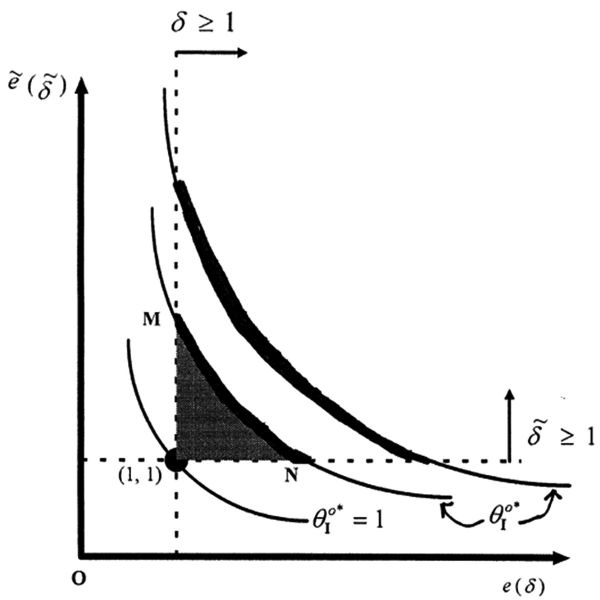


Fig. 11.9 Input variations

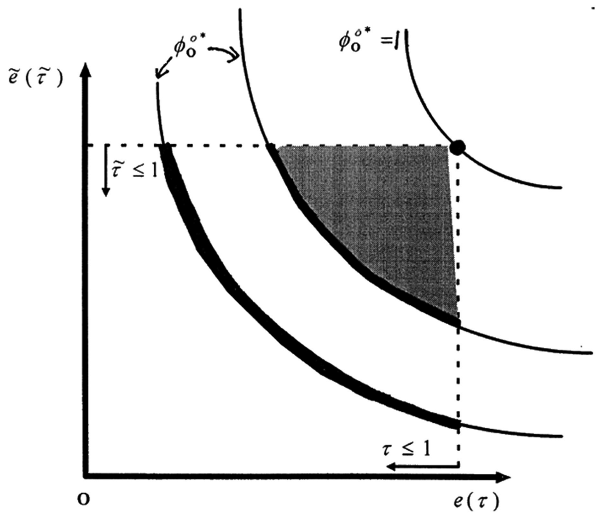


Fig. 11.10 Output variations

Then, the data perturbation can be expressed in a quadratic function,

$$e\,\tilde{e} = \Omega^{o} \tag{11.17}$$

Function (11.16) gives an upper boundary for input changes and a lower boundary for output changes. Figures 11.9 and 11.10 illustrate the admissible regions for e and $\tilde{e}$. For example, in Fig. 11.9, since $\delta \geq 1$ and $\tilde{\delta}_i \geq 1$, only part of the function $\delta\tilde{\delta} = \theta_{\mathrm{I}}^{o*}$ forms the upper boundary of a admissible region for δ and $\tilde{\delta}$. Any data variations fall below MN and above lines $\delta = 1$ and $\tilde{\delta} = 1$ will preserve the frontier status of DMU_o. The bigger the $\theta_{\mathbf{I}}^{o*}$ (or the smaller the $\phi_{\mathbf{O}}^{o*}$), the larger the input (output) variation regions will be. In fact, the function given by (11.16) defines the maximum percentage change rates for DMU_o and DMU_j ($j \neq o$).

Theorem 11.12 gives sufficient conditions for preserving efficiency. The following theorem implies necessary conditions for preserving efficiency of an extreme efficient DMU_o.

Theorem 11.13 Suppose $\theta_o^{\text{super}*} > 1$ and $\phi_o^{\text{super}*} < 1$, then

1. If $\delta_i \tilde{\delta}_i > \theta_{\mathbf{I}}^{o*}$ for $i \in \mathbf{I}$, then DMU_o will not be extreme efficient, where $\theta_{\mathbf{I}}^{o*}$ is the optimal value to (11.14).
2. If $\tau_r \tilde{\tau}_r < \phi_{\mathbf{O}}^{o*}$ for $r \in \mathbf{O}$, then DMU_o will not be extreme efficient, where $\phi_{\mathbf{O}}^{o*}$ is the optimal value to (11.15).

[Proof]: (i) We assume that DMU_o remains extreme efficient after the data changes in all DMUs with $\delta_i \tilde{\delta}_i > \theta_{\mathbf{I}}^{o*}$. Consider the input constraints associated with set **I** in (11.14),

$$\sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j \frac{x_{ij}}{\tilde{\delta}_i} \leq \hat{\theta}_{\mathbf{I}}^{o} \delta_i x_{io}, \, i \in \mathbf{I} \tag{11.18}$$

where $\hat{\theta}_{\mathbf{I}}^{o}$ is the objective function in (11.14).

Equation (11.17) is equivalent to

$$\sum_{\substack{j=1 \\ j \neq o}}^{n} \lambda_j x_{ij} \leq \hat{\theta}_{\mathbf{I}}^{o} \delta_i \tilde{\delta}_i x_{io}, \, i \in \mathbf{I}$$

Let $\hat{\theta}_{\mathbf{I}}^{o*}$ be the optimal value. Obviously, $\hat{\theta}_{\mathbf{I}}^{o*} = \theta_{\mathbf{I}}^{o*} / \delta_i \tilde{\delta}_i < 1$ where $\theta_{\mathbf{I}}^{o*}$ is the optimal value to (11.14). On the basis of Lemma 11.5 (i), $\hat{\theta}_{\mathbf{I}}^{o*}$ must be greater than one in (11.14) with input constraints of (11.17). A contradiction.

(ii) The proof is similar to (i), but is based upon (11.17). ■

Theorem 11.13 indicates that input (output) data perturbations in all DMUs beyond the variation regions prescribed by function (11.16) will change the efficiency classification of extreme efficient DMUs.

Note that $\delta_i \tilde{\delta}_i = \theta_{\mathbf{I}}^{o*}$ (or $\tau_r \tilde{\tau}_r = \phi_{\mathbf{O}}^{o*}$) may or may not keep the efficiency classification of an extreme efficient DMU_o. For example, in Fig. 11.8, A remains extreme efficient if $\delta_2 \tilde{\delta}_2 = \theta_{\{2\}}^{A*} = 7/5$. (In this situation, A coincides D and both become extreme efficient.) However, if we consider C and if C's second input is increased to $\theta_{\{2\}}^{C*} x_{2C} = (3/2) \times 2 = 3$, then C becomes a member of set F along the ray BB_1 in Fig. 11.8. (In this situation, we assume $\delta_2 = \theta_{\{2\}}^{C*} = 3/2$ for C and $\tilde{\delta}_2 = 1$ for the remaining DMUs of A, B and D.)

Turning to point A again. If we are only interested in whether a DMU remains on the frontier, rather than in its original efficiency classification, then we may still increase A's second input after A coincides D. We can find this "extra" data perturbation by applying a very small data perturbation to the changed DMU_o and then applying model (11.14) or (11.15). For example, we apply a data perturbation of ε to $\hat{x}_{2A}$ which is the new input value when $\delta_2\tilde{\delta}_2 = \theta_{\{2\}}^{A*} = 7/5$. If we use models (11.12) and (11.14), then we know that this changed DMU A with it second input equal to $\hat{x}_{2A} + \varepsilon$ is now in set F, and therefore A can still increase it x_2 to any amount larger than 7 and remains on the frontier. Note that, in this case, A may no longer be extreme efficient. In fact, $\delta_2\tilde{\delta}_2 = \theta_{\{2\}}^{A*} = 7/5$ prescribes a point on line segment AB including A and B. If $\delta_2\tilde{\delta}_2 > 7/5$, then A and D switch their positions. Namely, A becomes a weakly efficient DMU and D becomes an extreme efficient DMU.

Above developments consider the input changes or output changes in all DMUs. Next we consider the following modified DEA model for simultaneous variations of inputs and outputs

$$
\begin{aligned}
&\Gamma^* = \min \Gamma \\
&\text{subject to} \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j x_{ij} \le (1+\Gamma) x_{io} && i \in \mathbf{I} \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j x_{ij} \le x_{io} && i \notin \mathbf{I} \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j y_{rj} \ \ge \ (1-\Gamma) y_{ro} && r \in \mathbf{O} \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j y_{rj} \ \ge \ y_{ro} && r \notin \mathbf{O} \\
&\lambda_j (j \neq o) \ge 0, \ \ \Gamma \ \ \textit{unrestricted}
\end{aligned}
\tag{11.19}
$$

If $\mathbf{I} = \{1, 2, \ldots, m\}$ and $\mathbf{O} = \{1, 2, \ldots, s\}$, then (11.18) is identical to the model of Charnes et al. (1996) when variations in the data are only applied to DMU_o. Note that if DMU_o is a frontier point, then $\Gamma \ge 0$.

Theorem 11.14 Suppose DMU_o is a frontier point. If $1 \le \delta_i\tilde{\delta}_i \le 1+\Gamma^*$ and $1-\Gamma^* \le \tau_r\tilde{\tau}_r \le 1$, then DMU_o remains as a frontier point, where Γ^* is the optimal value to (11.18).

[Proof]: Equivalently we prove that if $\delta_i\tilde{\delta}_i = 1+\Gamma^*$ and $\tau_r\tilde{\tau}_r = 1-\Gamma^*$, then DMU_o still remains on the frontier. We assume that after the data changes, DMU_o

is a nonfrontier point, and therefore can be enveloped by the adjusted DMU_o $(j \neq o)$. Thus,

$$\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j \frac{x_{ij}}{\tilde{\delta}_i} \leq \delta_i x_{io} \qquad i \in \mathbf{I}$$

$$\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \leq x_{io} \qquad i \notin \mathbf{I}$$

$$\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j \frac{y_{rj}}{\tilde{\tau}_r} \geq \tau_r y_{ro} \qquad r \in \mathbf{O}$$

$$\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r \notin \mathbf{O}$$

That is

$$\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \leq (1+\Gamma^*) x_{io} \qquad i \in \mathbf{I}$$

$$\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \leq x_{io} \qquad i \notin \mathbf{I}$$

$$\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \geq (1-\Gamma^*) y_{ro} \qquad r \in \mathbf{O}$$

$$\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r \notin \mathbf{O}$$

This means that the adjusted DMU_o with $(1+\Gamma^*)x_{io}$ $(i \in \mathbf{I})$, x_{io} $(i \notin \mathbf{I})$, $(1-\Gamma^*)y_{ro}$ $(r \in \mathbf{O})$ and y_{ro} $(r \notin \mathbf{O})$ can be enveloped by the original DMU_j $(j \neq o)$. However, by Charnes et al. (1996), we know that proportional changes to inputs and outputs respectively within the computed values of $(1+\Gamma^*)$ and $(1-\Gamma^*)$ cannot change the efficiency of DMU_o when the remaining DMU_j $(j \neq o)$ are fixed. Therefore, this leads to a contradiction and completes the proof. ■

The result in Theorem 11.14 generalizes the finding of Charnes et al. (1996) to the situation where variations in the data are applied to all DMUs. Similar to Theorem 11.13, for an extreme efficient DMU_o, if $\delta_i \tilde{\delta}_i > 1+\Gamma^*$ and $\tau_r \tilde{\tau}_r < 1-\Gamma^*$, then DMU_o will not remain extreme efficient.

11.4.2 Sensitivity Analysis Under VRS

It is obvious that the results in the previous section hold for the VRS frontier DMUs if we add the additional constraint of $\sum_{j\neq o}\lambda_j = 1$ into models (11.12), (11.13), (11.14) and (11.15), respectively.

Because of the translation invariance property resulted from the convex constraint of $\sum_{j\neq o}\lambda_j = 1$ in the VRS models, we are able to discuss the simultaneous absolute data changes in all DMUs. That is,

Absolute Data Perturbations (Variations)

For DMU_o

$$\begin{cases} \hat{x}_{io} = x_{io} + \alpha_i & \alpha_i \geq 0,\ i \in \mathbf{I} \\ \hat{x}_{io} = x_{io} & i \notin \mathbf{I} \end{cases} \text{and} \begin{cases} \hat{y}_{ro} = y_{ro} - \beta_r & \beta_r \geq 0,\ r \in \mathbf{O} \\ \hat{y}_{ro} = y_{ro} & r \notin \mathbf{O} \end{cases}$$

For DMU_j $(j \neq o)$

$$\begin{cases} \hat{x}_{ij} = x_{ij} - \tilde{\alpha}_i & \tilde{\alpha}_i \geq 0,\ i \in \mathbf{I} \\ \hat{x}_{ij} = x_{ij} & i \notin \mathbf{I} \end{cases} \text{and} \begin{cases} \hat{y}_{rj} = y_{rj} + \tilde{\beta}_r & \tilde{\beta}_r \geq 0,\ r \in \mathbf{O} \\ \hat{y}_{rj} = y_{rj} & r \notin \mathbf{O} \end{cases}$$

where (^) represents adjusted data. Note that the data changes defined above are not only applied to all DMUs, but also different in various inputs and outputs. In this case the sensitivity analysis results are also suitable to the slack-based models.

We modify model (11.18) to the following linear programming problem

$$\begin{aligned} &\gamma^* = \min\ \gamma \\ &\text{subject to} \\ &\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \leq x_{io} + \gamma \qquad i \in \mathbf{I} \\ &\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \leq x_{io} \qquad i \notin \mathbf{I} \\ &\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \geq y_{ro} - \gamma \qquad r \in \mathbf{O} \\ &\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r \notin \mathbf{O} \\ &\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j = 1 \\ &\gamma,\ \lambda_j (j\neq o) \geq 0. \end{aligned} \tag{11.20}$$

If $\mathbf{I} = \{1, 2,\ldots, m\}$ and $\mathbf{O} = \{1, 2,\ldots, s\}$, then model (11.19) is used by Charnes et al. (1992) to study the sensitivity of efficiency classifications in the additive model via L_∞ norm when variations in the data are only applied to DMU_o.

Theorem 11.15 Suppose DMU_o is a frontier point. If $0 \le \alpha_i + \tilde{\alpha}_i \le \gamma^*$ $(i \in \mathbf{I})$, $0 \le \beta_r + \tilde{\beta}_r \le \gamma^*$ $(r \in \mathbf{O})$, then DMU_o remains as a frontier point, where γ^* is the optimal value to (11.19).

[Proof]: The proof is similar to that of Theorem 11.13 by noting that $\sum_{j \ne o} \lambda_j = 1$. ■

If $\mathbf{O} = \varnothing$, then (11.19) only considers absolute changes in inputs. If $\mathbf{I} = \varnothing$, then (10) only considers absolute changes in output. For different choices of subsets $\mathbf{I}$ and $\mathbf{O}$, we can determine the sensitivity of DMU_o to the absolute changes of different sets of inputs or (and) outputs when DMU_o's efficiency is deteriorating and DMU_j's $(j \ne o)$ efficiencies are improving.

We may change the objective function of (11.19) to "minimize $\sum_{i \in \mathbf{I}} r_i^- + \sum_{r \in \mathbf{O}} \gamma_r^+$" and obtain the following super-efficiency DEA model

$$
\begin{aligned}
&\min \sum_{i \in \mathbf{I}} \gamma_i^- + \sum_{r \in \mathbf{O}} \gamma_r^+ \\
&\text{subject to} \\
&\sum_{\substack{j=1 \\ j \ne o}}^{n} \lambda_j x_{ij} \le x_{io} + \gamma_i^- \qquad i \in \mathbf{I} \\
&\sum_{\substack{j=1 \\ j \ne o}}^{n} \lambda_j x_{ij} \le x_{io} \qquad i \notin \mathbf{I} \\
&\sum_{\substack{j=1 \\ j \ne o}}^{n} \lambda_j y_{rj} \ge y_{ro} - \gamma_r^+ \qquad r \in \mathbf{O} \\
&\sum_{\substack{j=1 \\ j \ne o}}^{n} \lambda_j y_{rj} \ge y_{ro} \qquad r \notin \mathbf{O} \\
&\sum_{\substack{j=1 \\ j \ne o}}^{n} \lambda_j = 1 \\
&\gamma_i^-, \gamma_r^+, \lambda_j (j \ne o) \ge 0
\end{aligned}
\tag{11.21}
$$

We then obtain a generalized model under L_1 norm. The results in Charnes et al. (1992) are generalized to the situation of data changes in all DMUs by the following Theorem.

Theorem 11.16 Suppose DMU_o is a frontier point. If $0 \le \alpha_i + \tilde{\alpha}_i \le \gamma_i^{-*}$ $(i \in \mathbf{I})$, $0 \le \beta_r + \tilde{\beta}_r \le \gamma_r^{+*}$ $(r \in \mathbf{O})$, then DMU_o remains as a frontier point, where γ_i^{-*} $(i \in \mathbf{I})$ and γ_r^{+*} $(r \in \mathbf{O})$ are optimal values in (11.20).

Table 11.3 Measure-specific super-efficiency DEA models

Frontier type	Input-Oriented	Output-Oriented
CRS	$\theta_{\mathbf{I}}^{o*} = \min \theta_{\mathbf{I}}^{o}$ subject to $\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \le \theta_{\mathbf{I}}^{o} x_{io} \quad i \in \mathbf{I}$ $\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \le x_{io} \quad i \notin \mathbf{I}$ $\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \ge y_{ro} \quad r = 1,2,\ldots,s$ $\theta_{\mathbf{I}}^{o}, \lambda_j (j \neq o) \ge 0.$	$\phi_{\mathbf{O}}^{o*} = \max \phi_{\mathbf{O}}^{o}$ subject to $\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \ge \phi_{\mathbf{O}}^{o} y_{ro} \quad r \in \mathbf{O}$ $\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j y_{rj} \ge y_{ro} \quad r \notin \mathbf{O}$ $\sum_{\substack{j=1\\ j\neq o}}^{n} \lambda_j x_{ij} \le x_{io} \quad i=1,\ldots,m$ $\phi_{\mathbf{O}}^{o}, \lambda_j (j \neq o) \ge 0.$
VRS	Add $\sum_{j\neq o} \lambda_j = 1$	
NIRS	Add $\sum_{j\neq o} \lambda_j \le 1$	
NDRS	Add $\sum_{j\neq o} \lambda_j \ge 1$	

[Proof]: The proof is similar to that of Theorem 11.13 and is omitted. ■

Similar to Theorem 11.13, for an extreme efficient DMU_o, if $\alpha_i + \tilde{\alpha}_i > \gamma_i^{-*}$ and $\beta_r + \tilde{\beta}_r > \gamma_r^{+*}$ then DMU_o will not remain extreme efficient.

11.4.3 Spreadsheet Models for Sensitivity Analysis

The current chapter presents a new approach for the sensitivity analysis of DEA models by using various super-efficiency DEA models. The sensitivity analysis approach simultaneously considers the data perturbations in all DMUs, namely, the change of the test DMU and the changes of the remaining DMUs. The data perturbations in the test DMU and the remaining DMUs can be different when all remaining DMUs work at improving their efficiencies against the deteriorating of the efficiency of the test efficient DMU. It is obvious that larger (smaller) optimal values to the input-oriented (output-oriented) super-efficiency DEA models presented in the current study correspond to greater stability of the test DMU in preserving efficiency when the inputs and outputs of all DMUs are changed simultaneously and unequally.

By using super-efficiency DEA models based upon the measure-specific models, the sensitivity analysis of DEA efficiency classification can be easily achieved. Since the approach uses optimal values to various super-efficiency DEA models, the results are stable and unique. By the additional constraint on $\sum_{j\neq o} \lambda_j$, the approach can easily be modified to study the sensitivity of other DEA models. Table 11.3 presents the measure-specific super-efficiency DEA models.

	A	B	C	D	E	F	G	H	I	J
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	Stability
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		0	Infeasible
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0	1.75189
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	Infeasible
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0	Infeasible
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0.71	Infeasible
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	Infeasible
8	Ford Motor	243283	24547	346990		137137	4139		0	0.70463
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	0.76929
10	Exxon	91296	40436	82000		110009	6470		0	Infeasible
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	Infeasible
12	Wal-Mart	37871	14762	675000		93627	2740		0.29	1.76516
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	0.48484
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0	Infeasible
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0	Infeasible
16	AT&T	88884	17274	299300		79609	139		0	0.52449
17										
18		Reference		DMU under	15	Stability				
19	**Constraints**	set		Evaluation		0.524492				
20	Assets	46618.92	≤	46618.922						
21	Equity	9060.07	≤	9060.07		Input				
22	Employees	203091.7	≤	299300						
23	Revenue	145773.2	≥	79609						
24	Profit	955.1922	≥	139						
25	Σλ	1								
26	λo	0								

Fig. 11.11 Input sensitivity analysis spreadsheet model

The stability measure is actually the optimal value to a specific measure-specific super-efficiency DEA model. Thus, the sensitivity analysis can be performed based upon the spreadsheets for related measure-specific models discussed in Chap. 3.

Figure 11.11 shows an input-oriented VRS measure-specific super-efficiency model where I= {Assets, Equity}. i.e., we are interested in the sensitivity of VRS efficiency to the (proportional) data changes in Assets and Equity.

In Fig. 11.11, cell F19 represents θ_I^o. The formulas for this spreadsheet are

Cell B20 =SUMPRODUCT(B2:B16,$I$2:$I$16)
Cell B21 =SUMPRODUCT(C2:C16,$I$2:$I$16)
Cell B22 =SUMPRODUCT(D2:D16,$I$2:$I$16)
Cell B23 =SUMPRODUCT(F2:F16,$I$2:$I$16)
Cell B24 =SUMPRODUCT(G2:G16,$I$2:$I$16)
Cell B25 =SUM(I2:I16)
Cell B26 =INDEX(I2:I16,E18,1)

Cell D20 =$F$19*INDEX(B2:B16,E18,1)
Cell D21 =$F$19*INDEX(C2:C16,E18,1)
Cell D22 =INDEX(D2:D16,E18,1)
Cell D23 =INDEX(F2:F16,E18,1)
Cell D24 =INDEX(G2:G16,E18,1)

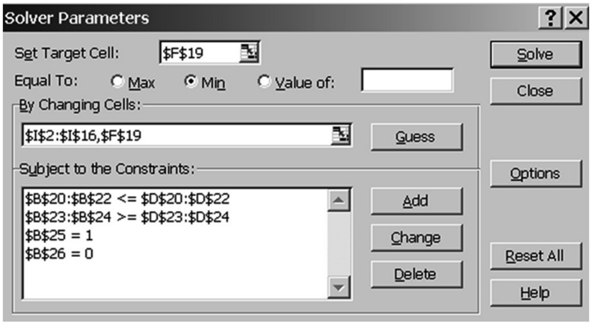


Fig. 11.12 Solver parameters for input sensitivity analysis

	A	B	C	D	E	F	G	H	I	J
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	Stability
2	Mitsubishi	91920.6	10950	36000		184365.2	1		0.8	0.94407
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0.19	0.93499
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	Infeasible
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0.01	Infeasible
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0	Infeasible
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	Infeasible
8	Ford Motor	243283	24547	346990		137137	4139		0	1.20986
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	1.46561
10	Exxon	91296	40436	82000		110009	6470		0	Infeasible
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	Infeasible
12	Wal-Mart	37871	14762	675000		93627	2740		0	Infeasible
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	2.11056
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0	Infeasible
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0	2.16259
16	AT&T	88884	17274	299300		79609	139		0	2.30681
17										
18		Reference		DMU under	15	Stability				
19	**Constraints**	set		Evaluation		2.306807				
20	Assets	88884	≤	88884						
21	Equity	10052.63	≤	17274			Output			
22	Employees	52057	≤	299300						
23	Revenue	183642.6	≥	183642.58						
24	Profit	139	≥	139						
25	Σλ	1								
26	λo	0								

Fig. 11.13 Output sensitivity analysis spreadsheet model

Figure 11.12 shows the Solver parameters for the spreadsheet shown in Fig. 11.11. If the optimal value in cell F19 is less than one, then this means that the associated company is VRS inefficient. The infeasibility in cells J2, J4:J7, J10:J11, and J14:J15 indicates that the corresponding companies remain VRS efficient to any simultaneous data changes in Assets and Equity across all DMUs. For DMU2 (Mitsui), we have the super-efficiency score of 1.75, indicating this DMU remains VRS efficient as long as the data variations satisfying $e\tilde{e} = 1.75$.

Next, we consider output changes. Figure 11.13 shows the spreadsheet for output-oriented VRS measure-specific super-efficiency model where $\mathbf{O} = \{\text{Revenue}\}$. In this spreadsheet, range names are used. They are, cells B2:D16—"InputUsed", cells F2:G16—“OutputProduced”, cells I2:I16—“Lambdas”, cells B20:B22—“ReferenceSetInput”, cells B23:B24—“ReferenceeSetOutput”, cell B25—“SumLambdas”, cell B26—“DMUo”, cells D20:D22—“DMUInput”, cells D23:D24—“DMUOutput”, cell E18—“DMU”, and cell F19—“SuperEfficiency”.

Based upon these range names, we have the following formulas for the spreadsheet shown in Fig. 11.13.

Cell B20 =SUMPRODUCT(INDEX(InputUsed,0,1),Lambdas)
Cell B21 =SUMPRODUCT(INDEX(InputUsed,0,2),Lambdas)
Cell B22 =SUMPRODUCT(INDEX(InputUsed,0,3),Lambdas)
Cell B23 =SUMPRODUCT(INDEX(OutputProduced,0,1),Lambdas)
Cell B24 =SUMPRODUCT(INDEX(OutputProduced, 0,2),Lambdas)
Cell B25 =SUM(Lambdas)
Cell B26 =INDEX(Lambdas,DMU,1)

Cell D20 =INDEX(InputUsed,DMU,1)
Cell D21 =INDEX(InputUsed,DMU,2)
Cell D22 =INDEX(InputUsed,DMU,3)
Cell D23 =SuperEfficiency*INDEX(OutputProduced,DMU,1)
Cell D24 =INDEX(OutputProduced,DMU,2)

Figure 11.14 shows the Solver parameters for the spreadsheet shown in Fig. 11.13. The calculation is performed by the following VBA procedure that can be applied to other data sets once the proper range names are defined.

```
Sub SensitivityGeneral()
Dim NDMUs As Integer, NInputs As Integer, NOutputs As Integer
NDMUs = 15
NInputs = 3
NOutputs = 2
Dim i As Integer
For i = 1 To NDMUs
Range("DMU") = i
SolverSolve UserFinish:=True
If SolverSolve(UserFinish:=True) = 5 Then
Range("A1").Offset(i, NInputs + NOutputs + 4) = "Infeasible"
Else
Range("A1").Offset(i,NInputs+NOutputs+4)=Range("SuperEfficienc
y")
End If
Next i
End Sub
```

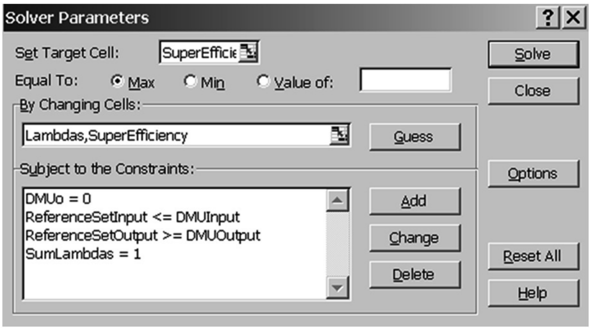


Fig. 11.14 Solver parameters for output sensitivity analysis

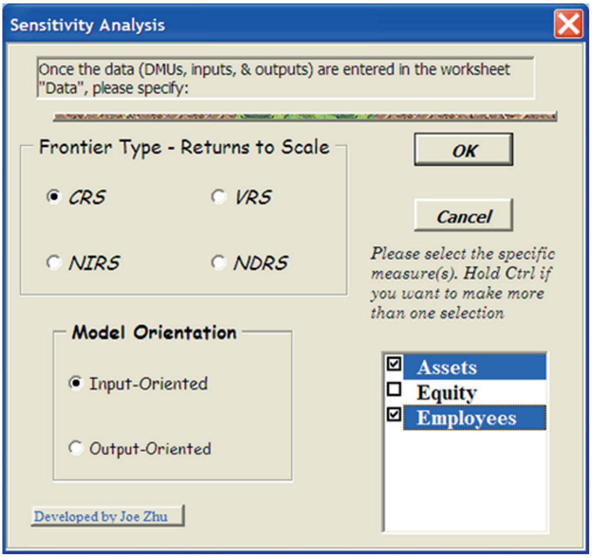


Fig. 11.15 Solver parameters for output sensitivity analysis

11.5 Sensitivity Analysis Using DEAFrontier

To perform the sensitivity analysis, select the "Perform Sensitivity Analysis" menu item. You will be prompted a form shown in Fig. 11.15. (You will select a model from Table 11.3.) The measures that are selected will be studied for sensitivity analysis. For example, in Fig. 11.5, Assets and Employees are selected (for input-oriented CRS model). The resulting super-efficiency score measures the efficiency stability with respect to changes in both Assets and Employees.

The results are reported in the "Sensitivity Report" sheet which records the optimal values to the related measure-specific super-efficiency model.

Based upon the discussion in this chapter, we can convert these super-efficiency scores into measures for efficiency stability.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_11) contains supplementary material, which is available to authorized users.

References

Ahn, T. S., & Seiford, L. M. (1993). Sensitivity of DEA to models and variable sets in a hypothesis test setting: The efficiency of University operations. In V. Ijiri (Ed.), *Creative and innovative approaches to the science of management* (pp. 191–208). New York: Quorum Books.

Charnes, A., & Neralic, L. (1990). Sensitivity analysis of the additive model in data envelopment analysis. *European Journal of Operational Research, 48,* 332–341.

Charnes, A., Cooper, W. W., Lewin, A. Y., Morey, R. C., & Rousseau, J. (1985a). Sensitivity and stability analysis in DEA. *Annals of Operations Research, 2,* 139–156.

Charnes, A., Cooper, W. W., Golany, B., Seiford, L. M., & Stutz, J. (1985b). Foundations of data envelopment analysis for Pareto-Koopman's efficient empirical production functions. *Journal of Econometrics, 30,* 1–17.

Charnes, A., Haag, S., Jaska, P., & Semple, J. (1992). Sensitivity of efficiency classifications in the additive model of data envelopment analysis. *International Journal of Systems Science, 23,* 789–798.

Charnes, A., Rousseau, J., & Semple, J. (1996). Sensitivity and stability of efficiency classifications in data envelopment analysis. *Journal of Productivity Analysis, 7,* 5–18.

Cooper, W. W., Li, S., Seiford, L. M., Thrall, R. M., & Zhu, J. (2001). Sensitivity and stability analysis in DEA: some recent developments. *Journal of Productivity Analysis, 15,* 217–246.

Neralic, L. (1994). Sensitivity analysis in data envelopment analysis: A review. In V. Rupnik, & M. Bogataj (Eds.), Proceedings of the symposium on operations research SOR '94 (pp. 29–42). Portoroz.

Rousseau, J. J., & Semple, J. H. (1995). Radii of classification preservation in data envelopment analysis: A case study of 'program follow-through'. *Journal of the Operational Research Society, 46,* 943–957.

Seiford, L. M., & Zhu, J. (1998a). Stability regions for maintaining efficiency in data envelopment analysis. *European Journal of Operational Research, 108*(1), 127–139.

Seiford, L. M., & Zhu, J. (1998b). Sensitivity analysis of DEA models for simultaneous changes in all the data. *Journal of the Operational Research Society, 49,* 1060–1071.

Seiford, L. M., & Zhu, J. (1999). Infeasibility of super efficiency data envelopment analysis models. *INFOR, 37*(2), 174–187.

Thompson, R. G., Dharmapala, P. S., & Thrall, R. M. (1994). Sensitivity analysis of efficiency measures with applications to Kansas farming and Illinois coal mining. In A. Charnes, W. W. Cooper, A. Y. Lewin, & L. M. Seiford (Eds.), *Data envelopment analysis: Theory, methodology, and applications*. Boston: Kluwer Academic Publishers.

Zhu, J. (1996). Robustness of the efficient DMUs in data envelopment analysis. *European Journal of Operational Research, 90*(3), 451–460.

Zhu, J. (2001). Super-efficiency and DEA sensitivity analysis. *European Journal of Operational Research, 129*(2), 443–455.

Chapter 12
Benchmarking Models

12.1 Introduction

Benchmarking is a process of defining valid measures of performance comparison among peer DMUs, using them to determine the relative positions of the peer DMUs and, ultimately, establishing a standard of excellence. In that sense, DEA can be regarded as a benchmarking tool, because the frontier identified can be regarded as an empirical standard of excellence.

Once the frontier is established, we may compare a set of new DMUs to the frontier. However, when a new DMU outperforms the identified frontier, a new frontier is generated by DEA. As a result, we do not have the same benchmark (frontier) for other (new) DMUs.

In the current chapter, we present a number of DEA-based benchmarking models where each (new) DMU is evaluated against a set of given benchmarks (standards).

12.2 Variable-benchmark Model

Cook et al. (2004) develop a set of variable-benchmark model. Let E^* represent the set of benchmarks or the best-practice identified by the DEA. Based upon the input-oriented CRS envelopment model, we have

$$
\begin{aligned}
&\min \delta^{CRS} \\
&\text{subject to} \\
&\sum_{j \in E^*} \lambda_j x_{ij} \le \delta^{CRS} x_i^{new} \\
&\sum_{j \in E^*} \lambda_j y_{rj} \ge y_r^{new} \\
&\lambda_j \ge 0, j \in E^*
\end{aligned}
\tag{12.1}
$$

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking,* International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_12,

where a new observation is represented by DMU^{new} with inputs x_i^{new} $(i = 1,\ldots,m)$ and outputs y_r^{new} $(r = 1,\ldots,s)$. The superscript of CRS indicates that the benchmark frontier composed by benchmark DMUs in set E^* exhibits CRS.

Model (12.1) measures the performance of DMU^{new} with respect to benchmark DMUs in set E^* when outputs are fixed at their current levels. Similarly, based upon the output-oriented CRS envelopment model, we can have a model that measures the performance of DMU^{new} in terms of outputs when inputs are fixed at their current levels.

$$\begin{aligned}
&\max \tau^{CRS}\\
&\text{subject to}\\
&\sum_{j\in E^*} \lambda_j x_{ij} \le x_i^{new}\\
&\sum_{j\in E^*} \lambda_j y_{rj} \ge \tau^{CRS} y_r^{new}\\
&\lambda_j \ge 0,\ j \in E^*
\end{aligned} \tag{12.2}$$

Theorem 12.1 $\delta^{CRS*} = 1/\tau^{CRS*}$, where δ^{CRS*} is the optimal value to model (12.1) and τ_o^{CRS*} is the optimal value to model (12.2).

[Proof]: Suppose $\lambda_j^*(j \in E^*)$ is an optimal solution associated with δ^{CRS*} in model (12.1). Now, let $\tau^{CRS*} = 1/\delta^{CRS*}$, and $\lambda_j' = \lambda_j^*\delta_o^{CRS*}$. Then τ^{CRS*} and λ_j' are optimal in model (12.2). Thus, $\delta^{CRS*} = 1/\tau^{CRS*}$. ■

Model (12.1) or (12.2) yields a benchmark for DMU^{new}. The ith input and the rth output for the benchmark can be expressed as

$$\begin{cases}
\sum_{j\in E^*} \lambda_j^* x_{ij} & (ith \quad input)\\
\sum_{j\in E^*} \lambda_j^* y_{ij} & (rth \quad output)
\end{cases} \tag{12.3}$$

Note also that although the DMUs associated with set E^* are given, the resulting benchmark may be different for each new DMU under evaluation. Because for each new DMU under evaluation, (12.3) may represent a different combination of DMUs associated with set E^*. Thus, models (12.1) and (12.2) represent a variable-benchmark scenario.

Theorem 12.2

1. $\delta^{CRS*} < 1$ or $\tau^{CRS*} > 1$ indicates that the performance of DMU_o^{new} is dominated by the benchmark in (12.3).
2. $\delta^{CRS*} = 1$ or $\tau^{CRS*} = 1$ indicates that DMU^{new} achieve the same performance level of the benchmark in (12.3).

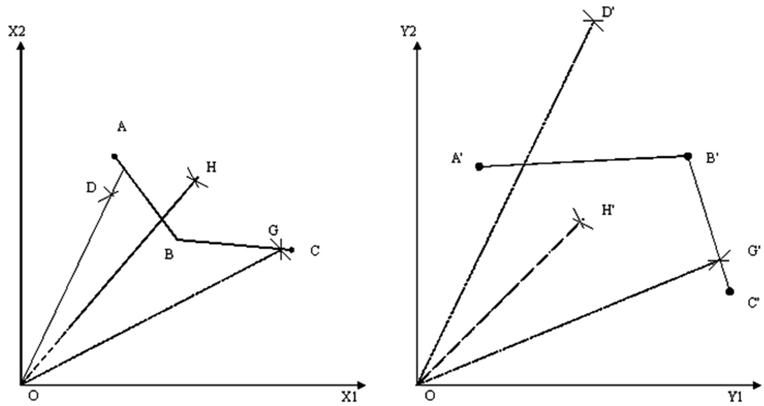


Fig. 12.1 Variable-benchmark model

3. $\delta^{CRS*} > 1$ or $\tau^{CRS*} < 1$ indicates that input savings or output surpluses exist in DMU_o^{new} when compared to the benchmark in (12.3).

[Proof]: (i) and (ii) are obvious results in terms of DEA efficiency concept.

Now, $\delta^{CRS*} > 1$ indicates that DMU^{new} can increase its inputs to reach the benchmark. This in turn indicates that δ^{CRS*}-1 measures the input saving achieved by DMU^{new}. Similarly, $\tau^{CRS*} < 1$ indicates that DMU^{new} can decrease its outputs to reach the benchmark. This in turn indicates that 1-τ^{CRS*} measures the output surplus achieved by DMU^{new}. ■

Figure 12.1 illustrates the three cases described in Theorem 12.2. ABC (A′B′C′) represents the input (output) benchmark frontier. D, H and G (or D′, H′, and G′) represent the new DMUs to be benchmarked against ABC (or A′B′C′). We have $\delta_D^{CRS*} > 1$ for DMU D ($\tau_{D'}^{CRS*} < 1$ for DMU D′) indicating that DMU D can increase its input values by δ_D^{CRS*} while producing the same amount of outputs generated by the benchmark (DMU D′ can decrease its output levels while using the same amount of input levels consumed by the benchmark). Thus, $\delta_D^{CRS*} > 1$ is a measure of input savings achieved by DMU D and $\tau_{D'}^{CRS*} < 1$ is a measure of output surpluses achieved by DMU D′.

For DMU G and DMU G′, we have $\delta_G^{CRS*} = 1$ and $\tau_{G'}^{CRS*} = 1$ indicating that they achieve the same performance level of the benchmark and no input savings or output surpluses exist. For DMU H and DMU H′, we have $\delta_H^{CRS*} < 1$ and $\tau_{H'}^{CRS*} > 1$ indicating that inefficiency exists in the performance of these two DMUs.

Note that for example, in Fig. 12.1, a convex combination of DMU A and DMU B is used as the benchmark for DMU D while a convex combination of DMU B and DMU C is used as the benchmark for DMU G. Thus, models (12.1) and (12.2) are called variable-benchmark models.

From Theorem 12.2, we can define δ^{CRS*}-1 or 1-τ^{CRS*} as the performance gap between DMU^{new} and the benchmark. Based upon δ^{CRS*} or τ^{CRS*}, a ranking of the benchmarking performance can be obtained.

It is likely that scale inefficiency may be allowed in the benchmarking. We therefore modify models (12.1) and (12.2) to incorporate scale inefficiency by assuming VRS.

$$
\begin{aligned}
&\min \delta^{VRS} \\
&\text{subject to} \\
&\sum_{j \in E^*} \lambda_j x_{ij} \le \delta^{VRS} x_i^{new} \\
&\sum_{j \in E^*} \lambda_j y_{rj} \ge y_r^{new} \\
&\sum_{j \in E^*} \lambda_j = 1 \\
&\lambda_j \ge 0, j \in E^*
\end{aligned}
\tag{12.4}
$$

$$
\begin{aligned}
&\max \tau^{VRS} \\
&\text{subject to} \\
&\sum_{j \in E^*} \lambda_j x_{ij} \le x_i^{new} \\
&\sum_{j \in E^*} \lambda_j y_{rj} \ge \tau^{VRS} y_r^{new} \\
&\sum_{j \in E^*} \lambda_j = 1 \\
&\lambda_j \ge 0, j \in E^*
\end{aligned}
\tag{12.5}
$$

Similar to Theorem 12.2, we have

Theorem 12.3

4. $\delta^{VRS*} < 1$ or $\tau^{VRS*} > 1$ indicates that the performance of DMU^{new} is dominated by the benchmark in (12.3).
5. $\delta^{VRS*} = 1$ or $\tau^{VRS*} = 1$ indicates that DMU^{new} achieve the same performance level of the benchmark in (12.3).
6. $\delta^{VRS*} > 1$ or $\tau^{VRS*} < 1$ indicates that input savings or output surpluses exist in DMU^{new} when compared to the benchmark in (12.3).

Note that model (12.2) is always feasible, and model (12.1) is infeasible only if certain patterns of zero data are present (Zhu 1996). Thus, if we assume that all the data are positive, (12.1) is always feasible. However, unlike models (12.1) and (12.2), models (12.4) and (12.5) may be infeasible.

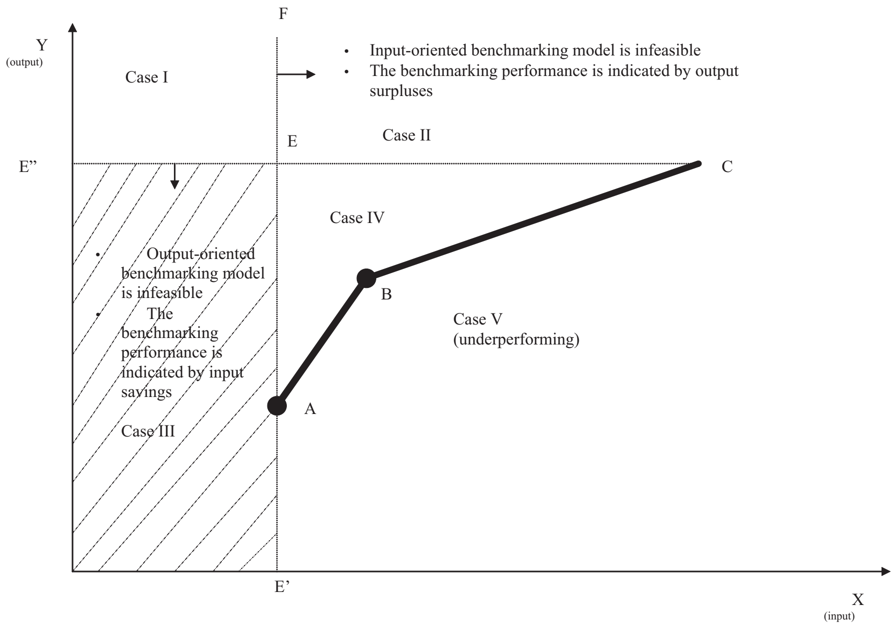


Fig. 12.2 Infeasibility of VRS variable-benchmark model

Theorem 12.4

1. If model (12.4) is infeasible, then the output vector of DMU^{new} dominates the output vector of the benchmark in (12.3).
2. If model (12.5) is infeasible, then the input vector of DMU^{new} dominates the input vector of the benchmark in (12.3).

[Proof]: The proof follows directly from the necessary and sufficient conditions for infeasibility in super-efficiency DEA model provided in Seiford and Zhu (1999). ■

The implication of the infeasibility associated with models (12.4) and (12.5) needs to be carefully examined. Consider Fig. 12.2 where ABC represents the benchmark frontier. Models (12.4) and (12.5) yield finite optimal values for any DMU^{new} located below EC and to the right of EA. Model (12.4) is infeasible for DMU^{new} located above ray E″C and model (12.5) is infeasible for DMU^{new} located to the left of ray E′E.

Both models (12.4) and (12.5) are infeasible for DMU^{new} located above E″E and to the left of ray EF. Note that if DMU^{new} is located above E″C, its output value is greater than the output value of any convex combinations of A, B and C.

Note also that if DMU^{new} is located to the left of E′F, its input value is less than the input value of any convex combinations of A, B and C.

Based upon Theorem 12.4 and Fig. 12.2, we have four cases:

Case I When both models (12.4) and (12.5) are infeasible, this indicates that DMU^{new} has the smallest input level and the largest output level compared to the benchmark. Thus, both input savings and output surpluses exist in DMU^{new}.

Case II When model (12.4) is infeasible and model (12.5) is feasible, the infeasibility of model (12.4) is caused by the fact that DMU^{new} has the largest output level compared to the benchmark. Thus, we use model (12.5) to characterize the output surpluses.

Case III When model (12.5) is infeasible and model (12.4) is feasible, the infeasibility of model (12.5) is caused by the fact that DMU^{new} has the smallest input level compared to the benchmark. Thus, we use model (12.4) to characterize the input savings.

Case IV When both models (12.4) and (12.5) are feasible, we use both of them to determine whether input savings and output surpluses exist.

If we change the constraint $\sum \lambda_j = 1$ to $\sum \lambda_j \le 1$ and $\sum \lambda_j \ge 1$, then we obtain the NIRS and NDRS variable-benchmark models, respectively. Infeasibility may be associated with these two types of RTS frontiers, and we should apply the four cases discussed above. Table 12.1 summarizes the variable-benchmark models.

We next use 22 internet companies to illustrate the variable-benchmark models. Table 12.2 presents the data. We have four inputs: (1) number of website visitors (thousand), (2) number of employees (person), (3) marketing expenditure ($ million), and (4) development expenditure ($ million), and two outputs: (1) number of customers, and (2) revenue ($ million).

Suppose we select the first seven companies (Barnes & Noble, Amazon.com, CDnow, eBay, 1–800-Flowers, Buy.com, and FTD.com) as the benchmarks. If we apply the output-oriented CRS envelopment model to the seven companies, the top three companies (Barnes & Noble, Amazon.com, and CDnow) are not on the best-practice frontier, and therefore can be excluded. However, if we include them in the benchmark set, the benchmarking results will not be affected. Because λ_j^* related to the three companies must be equal to zero.

The spreadsheet model of the variable-benchmark models is very similar to the context-dependent DEA spreadsheet model. In fact, the evaluation background now is the selected benchmarks. Figure 12.3 shows the spreadsheet model for the output-oriented CRS variable-benchmark model where the benchmarks (evaluation background) are entered in rows 2–8.

Cell F2 is reserved to indicate the DMU under benchmarking. Cell F4 is the target cell which represent the τ_o^{CRS} in model (12.2). Cells I2:I8 represent the λ_j for the benchmarks. Cell B9 contains the formula "=SUMPRODUCT (B2:B8,$I$2:$I$8)". This formula is then copied into cells C9:E9. Cell G9 contains the formula "=SUMPRODUCT(G2:G8,$I$2:$I$8)". This formula is then copied into cell H9.

Cells B11:E11, and Cells G11:H11 contain the formulas for the DMU under benchmarking—the right-hand-side of model (12.2). The formula for B11 is "=INDEX(B12:B26,$F$2,1)", and is copied into cells C11:E11. The formula for cell G11 is "=$F$4*INDEX(G12:G26,$F$2,1)", and is copied into cell H11.

Table 12.1 Variable-benchmark models

Frontier type	Input-oriented	Output-oriented
CRS	$\min \delta^{Frontier}$ subject to $\sum_{j\in E^*} \lambda_j x_{ij} \le \delta^{Frontier} x_i^{new}$ $\sum_{j\in E^*} \lambda_j y_{rj} \ge y_r^{new}$ $\lambda_j \ge 0, j \in E^*$	$\max \tau^{Frontier}$ subject to $\sum_{j\in E^*} \lambda_j x_{ij} \le x_i^{new}$ $\sum_{j\in E^*} \lambda_j y_{rj} \ge \tau^{Frontier} y_r^{new}$ $\lambda_j \ge 0, j \in E^*$
VRS	Add $\sum \lambda_j = 1$	
NIRS	Add $\sum \lambda_j \le 1$	
NDRS	Add $\sum \lambda_j \ge 1$	

Table 12.2 Data for the internet companies

Company	Visitors	Employee	Marketing	Develop-ment	Customers	Revenue
Barnes&Noble	64,812	1,237	111.55	21.01	4,700,000	202.57
Amazon.com	177,744	7,600	413.2	159.7	16,900,000	1640
CDnow	79,848	502	89.73	23.42	3,260,000	147.19
eBay	168,384	300	95.96	23.79	10,010,000	224.7
1–800-Flowers	11,940	2,100	92.15	8.07	7,800,000	52.89
Buy.com	27,372	255	71.3	7.84	1,950,000	596.9
FTD.com	11,856	75	29.93	5.29	1,800,000	62.6
Autobytel.com	12,000	225	44.18	14.26	2,065,000	40.3
Beyond.com	17,076	250	81.35	12.39	2,000,000	117.28
eToys	13,896	940	120.46	43.43	1,900,000	151.04
E*Trade	29,532	2,400	301.7	78.5	1,551,000	621.4
Garden.com	16,344	290	16	4.8	1,070,000	8.2
Drugstore.com	19,092	408	61.5	14.9	695,000	34.8
Outpost.com	7,716	164	41.67	7	627,000	188.6
iPrint	42,132	225	8.13	3.54	380,000	3.26
Furniture.com	10,668	213	33.949	6.685	260,000	12.904
PlanetRX.com	17,124	390	55.18	12.95	254,000	8.99
NextCard	46,836	365	24.65	22.05	220,000	26.56
PetsMart.com	18,564	72	33.47	2.43	180,000	12.45
Peapod	2,076	1,020	7.17	3.54	111,900	73.13
Webvan	1,680	1,000	11.75	15.24	47,000	13.31
CarsDirect.com	15,612	702	33.43	2.14	12,885	98.56

	A	B	C	D	E	F	G	H	I
1	Company	Visitors	Employee	Marketing	Development		Customers	Revenue	λ
2	Barnes&Noble	64812	1237	111.55	21.01	15	4700000	202.57	0
3	Amazon.com	177744	7600	413.2	159.7	Score	16900000	1640	0
4	Cdnow	79848	502	89.73	23.42	1.653097978	3260000	147.19	0
5	eBay	168384	300	95.96	23.79		10010000	224.7	0
6	1-800-Flowers	11940	2100	92.15	8.07		7800000	52.89	0
7	Buy.com	27372	255	71.3	7.84		1950000	596.9	0.272959184
8	FTD.com	11856	75	29.93	5.29		1800000	62.6	0
9	**Benchmarks**	7471.438776	69.6045918	19.46199	2.14		532270.41	162.929337	
10		≤	≤	≤	≤		≥	≥	Benchmarking
11	**DMU under evaluation**	15612	702	33.43	2.14		21300.167	162.929337	Score
12	Autobytel.com	12000	225	44.18	14.26		2065000	40.3	1.095779422
13	Beyond.com	17076	250	81.35	10.39		2000000	117.28	1.327240986
14	eToys	13896	940	120.46	43.43	Variable	1900000	151.04	1.600761668
15	E*Trade	29532	2400	301.7	78.5	Benchmark	1551000	621.4	1.036374356
16	Garden.com	16344	290	16	4.8		1070000	8.2	1.42713759
17	Drugstore.com	19092	408	61.5	14.9		695000	34.8	4.852307242
18	Outpost.com	7716	164	41.67	7		627000	188.6	0.890769639
19	iPrint	42132	225	8.13	3.54		380000	3.26	2.231776947
20	Furniture.com	10668	213	33.949	6.685		260000	10.904	7.369719683
21	PlanetRX.com	17124	390	55.18	12.95		254000	8.99	12.93451824
22	NextCard	46836	365	24.65	22.05		220000	26.56	5.917828607
23	PetsMart.com	18564	72	33.47	2.43		180000	10.45	5.549892175
24	Peapod	2076	1020	7.17	3.54		111900	73.13	0.619051561
25	Webvan	1680	1000	11.75	15.24		47000	13.31	2.732844609
26	CarsDirect.com	15612	702	33.43	2.14		12885	98.56	1.653097978

Fig. 12.3 Output-oriented CRS variable-benchmark spreadsheet model

Figure 12.4 shows the Solver parameters for the spreadsheet model shown in Fig. 12.3. A VBA procedure “VariableBenchmark” is used to record the benchmarking scores into cells I12:I26.

```
Sub VariableBenchmark()
Dim i As Integer
For i = 1 To 15
Range("F2") = i
SolverSolve UserFinish:=True
Range("I" & i + 11) = Range("F4")
Next
End Sub
```

Because the model in Fig. 12.3 is an output-oriented model, a smaller score (τ^{CRS*}) indicates a better performance. Thus, Peapod is the best company with respective to the specified benchmarks. The non-zero optimal λ_j^* indicates the actual benchmark for a company under benchmarking. For example, Buy.com is used as the actual benchmark for CarsDirect.com (see cell I7 in Fig. 12.3).

If we use the input-oriented CRS variable-benchmark model, we need change the formula for cell B11 in Fig. 12.3 to “=$F$4*INDEX (B12:B26,$F$2,1)”. This formula is then copied into cells C11:E11. The formula for cell G11 is changed to “=INDEX(G12:G26,$F$2,1)” and is copied into cell H11. All the other formulas in Fig. 12.3 remain unchanged.

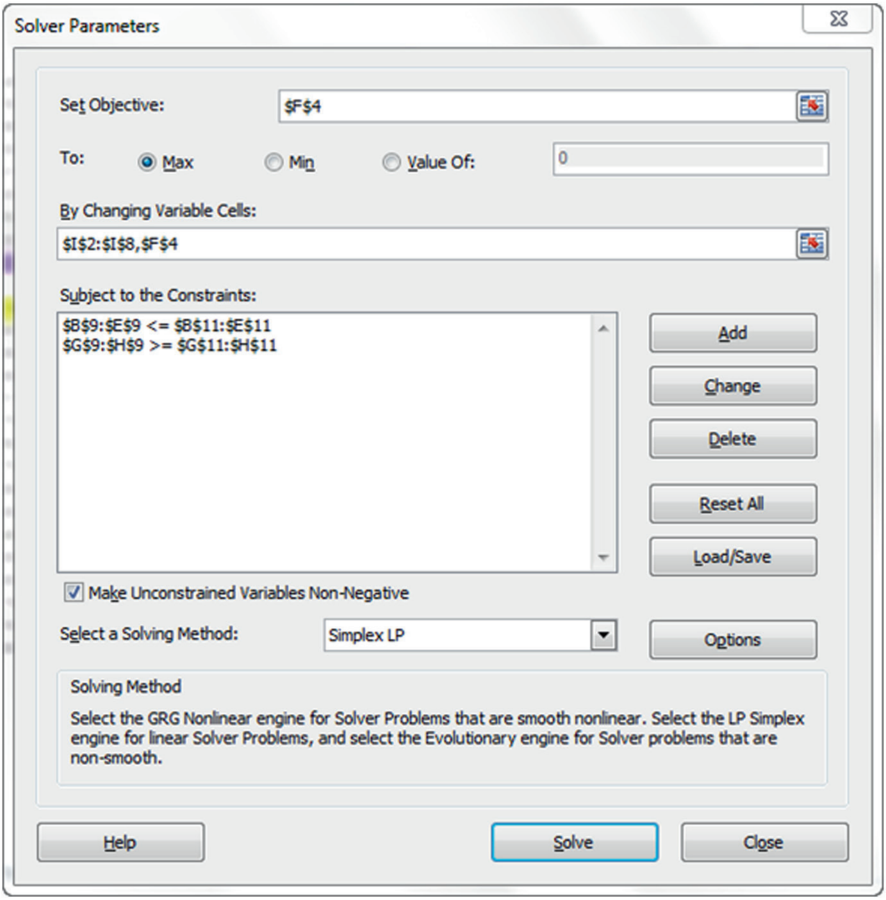


Fig. 12.4 Solver parameters for output-oriented CRS variable-benchmark model

We also need to change the Solver parameters shown in Fig. 12.4 by selecting "Min", as shown in Fig. 12.5. Figure 12.6 shows the spreadsheet model for the input-oriented CRS variable-benchmark model and the benchmarking scores. It can be seen that Theorem 12.1 is true.

We now consider the input-oriented VRS variable-benchmark model. We need to add a cell representing $\sum \lambda_j$ in the spreadsheet shown in Fig. 12.6. We select cell I9, and enter the formula "=SUM(I2:I8)". We also need to add an additional constraint on $\sum \lambda_j = 1$ in the Solver parameters shown in Fig. 12.5. This constraint is "$I$9 = 1", as shown in Fig. 12.7.

Figure 12.8 shows the spreadsheet for the input-oriented VRS variable-benchmark model and the benchmarking scores in cells I12:I26. The button "VRS Variable Benchmark" is linked to the VBA procedure "VRSVariableBenchmark".

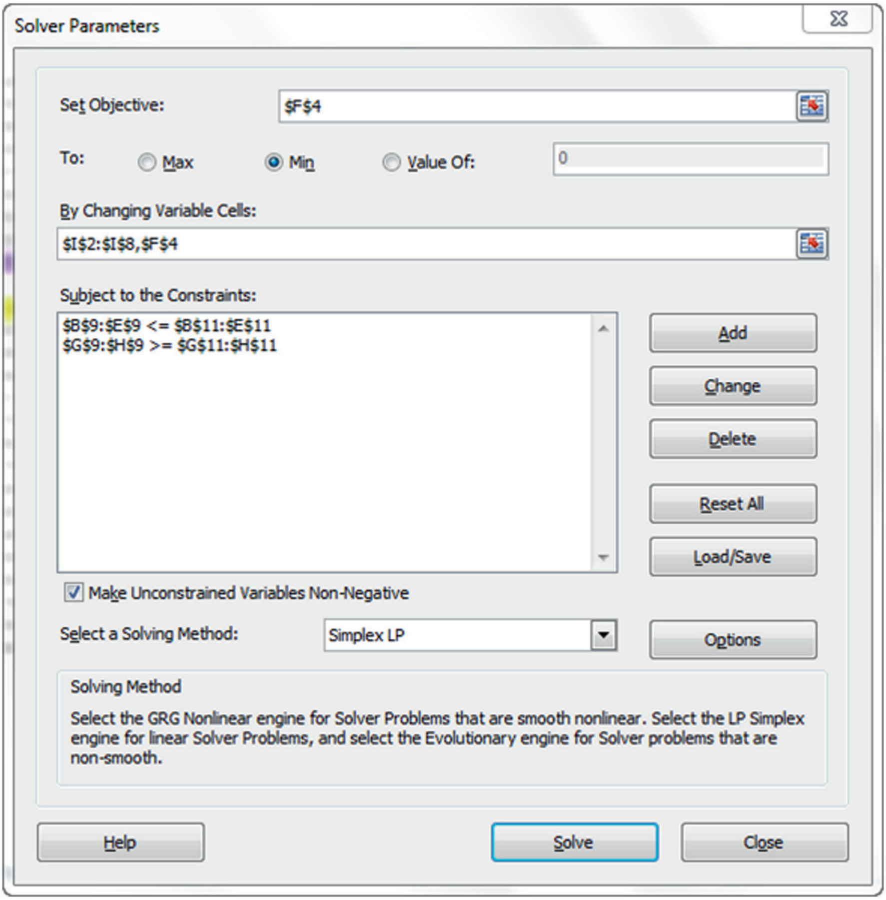


Fig. 12.5 Solver parameters for input-oriented CRS variable-benchmark model

```
Sub VRSVariableBenchmark()
Dim i As Integer
For i = 1 To 15
Range("F2") = i
SolverSolve UserFinish:=True
If SolverSolve(UserFinish:=True) = 5 Then
Range("I" & i + 11) = "Infeasible"
Else
Range("I" & i + 11) = Range("F4")
End If
Next
End Sub
```

	A	B	C	D	E	F	G	H	I
1	Company	Visitors	Employee	Marketing	Development		Customers	Revenue	λ
2	Barnes&Noble	64812	1237	111.55	21.01	15	4700000	202.57	0
3	Amazon.com	177744	7600	413.2	159.7	Score	16900000	1640	0
4	Cdnow	79848	502	89.73	23.42	[redacted]	3260000	147.19	0
5	eBay	168384	300	95.96	23.79		10010000	224.7	0
6	1-800-Flowers	11940	2100	92.15	8.07		7800000	52.89	0
7	Buy.com	27372	255	71.3	7.84		1950000	596.9	0.165119786
8	FTD.com	11856	75	29.93	5.29		1800000	62.6	0
9	**Benchmarks**	4519.65877	42.1055453	11.77304	1.294539119		321983.58	98.56	input-oriented
10		|∧	|∧	|∧	|∧		∨|	∨|	Benchmarking
11	**DMU under evaluation**	9444.086319	424.657225	20.22264	1.294539119		12885	98.56	Score
12	Autobytel.com	12000	225	44.18	14.26		2065000	40.3	0.912592425
13	Beyond.com	17076	250	81.35	10.39		2000000	117.28	0.753442676
14	eToys	13896	940	120.46	43.43	Variable	1900000	151.04	0.624702615
15	E*Trade	29532	2400	301.7	78.5	Benchmark	1551000	621.4	0.9649023
16	Garden.com	16344	290	16	4.8		1070000	8.2	0.700703287
17	Drugstore.com	19092	408	61.5	14.9		695000	34.8	0.206087527
18	Outpost.com	7716	164	41.67	7		627000	188.6	1.122624702
19	iPrint	42132	225	8.13	3.54		380000	3.26	0.448073452
20	Furniture.com	10668	213	33.949	6.685		260000	10.904	0.135690371
21	PlanetRX.com	17124	390	55.18	12.95		254000	8.99	0.077312505
22	NextCard	46836	365	24.65	22.05		220000	26.56	0.168980899
23	PetsMart.com	18564	72	33.47	2.43		180000	10.45	0.180183681
24	Peapod	2076	1020	7.17	3.54		111900	73.13	1.615374328
25	Webvan	1680	1000	11.75	15.24		47000	13.31	0.365919085
26	CarsDirect.com	15612	702	33.43	2.14		12885	98.56	0.604924822

Fig. 12.6 Input-oriented CRS variable-benchmark spreadsheet model

Because of the VRS frontier, the model may be infeasible. The SolverSolve function returns an integer value that indicates Solver's "success". If this value is 5, it means that there are no feasible solutions. This is represented by the statement "SolverSolve(UserFinish:=True) = 5". In the procedure, if the Solver returns a value of 5, then the procedure records "infeasible". Otherwise, the procedure records the optimal value in cell F4 of Fig. 12.8.

12.3 Fixed-benchmark Model

Although the benchmark frontier is given in the variable-benchmark models, a DMU^{new} under benchmarking has the freedom to choose a subset of benchmarks so that the performance of DMU^{new} can be characterized in the most favorable light. Situations when the same benchmark should be fixed are likely to occur. For example, the management may indicate that DMUs A and B in Fig. 12.1 should be used as the fixed benchmark. i.e., DMU C in Fig. 12.1 may not be used in constructing the benchmark.

To couple with this situation, Cook et al. (2004) turn to the multiplier models. For example, the input-oriented CRS multiplier model determines a set of referent best-practice DMUs represented by a set of binding constraints in optimality. Let set $\boldsymbol{B} = \{DMU_j : j \in \mathbf{I}_B\}$ be the selected subset of benchmark set E^*. i.e., $\mathbf{I}_B \subset E^*$. Based upon the input-oriented CRS multiplier model, we have

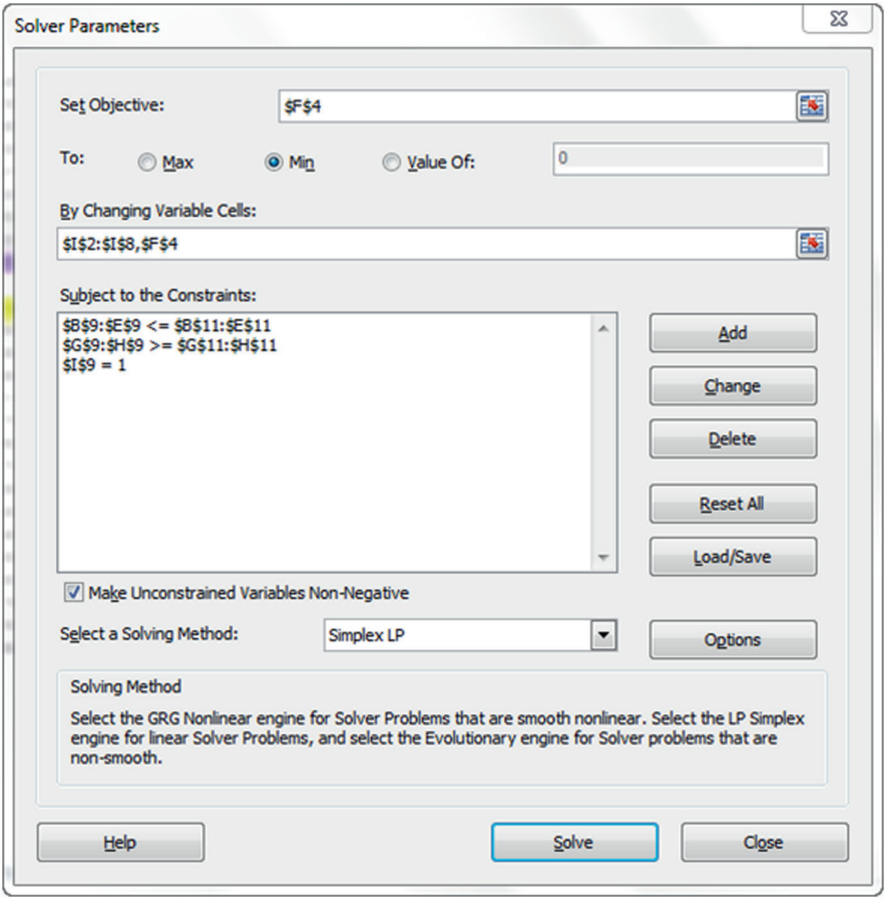


Fig. 12.7 Solver parameters for Input-oriented VRS variable-benchmark model

$$
\begin{aligned}
&\tilde{\sigma}^{\text{CRS}*} = \quad \max \sum_{r=1}^{s} \mu_r y_r^{new} \\
&\text{subject to} \\
&\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{s} v_i x_{ij} = 0 \qquad j \in \mathbf{I}_B \\
&\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{s} v_i x_{ij} \le 0 \qquad j \notin \mathbf{I}_B \\
&\sum_{i=1}^{m} v_i x_i^{new} = 1 \\
&\mu_r, v_i \ge 0.
\end{aligned}
\tag{12.6}
$$

By applying equalities in the constraints associated with benchmark DMUs, model (12.6) measures DMU^{new}'s performance against the benchmark constructed by

	A	B	C	D	E	F	G	H	I
1	Company	Visitors	Employee	Marketing	Development		Customers	Revenue	λ
2	Barnes&Noble	64812	1237	111.55	21.01	15	4700000	202.57	0
3	Amazon.com	177744	7600	413.2	159.7	Score	16900000	1640	0
4	Cdnow	79848	502	89.73	23.42	2.552160133	3260000	147.19	0
5	eBay	168384	300	95.96	23.79		10010000	224.7	0
6	1-800-Flowers	11940	2100	92.15	8.07		7800000	52.89	0
7	Buy.com	27372	255	71.3	7.84		1950000	596.9	0.067303013
8	FTD.com	11856	75	29.93	5.29		1800000	62.6	0.932696987
9	**Benchmarks**	12900.27355	87.1145424	32.71433	5.461622684		1810095.5	98.56	1
10		≤	≤	≤	≤		≥	≥	
11	**DMU under evaluation**	39844.32399	1791.61641	85.31871	5.461622684		12885	98.56	Score
12	Autobytel.com	12000	225	44.18	14.26		2065000	40.3	0.988309167
13	Beyond.com	17076	250	81.35	10.39		2000000	117.28	0.787957227
14	eToys	13896	940	120.46	43.43	VRS Variable	1900000	151.04	1.038346917
15	E*Trade	29532	2400	301.7	78.5	Benchmark	1551000	621.4	0.95011279
16	Garden.com	16344	290	16	4.8		1070000	8.2	Infeasible
17	Drugstore.com	19092	408	61.5	14.9		695000	34.8	Infeasible
18	Outpost.com	7716	164	41.67	7		627000	188.6	Infeasible
19	iPrint	42132	225	8.13	3.54		380000	3.26	Infeasible
20	Furniture.com	10668	213	33.949	6.685		260000	10.904	Infeasible
21	PlanetRX.com	17124	390	55.18	12.95		254000	8.99	Infeasible
22	NextCard	46836	365	24.65	22.05		220000	26.56	Infeasible
23	PetsMart.com	18564	72	33.47	2.43		180000	10.45	Infeasible
24	Peapod	2076	1020	7.17	3.54		111900	73.13	5.858280241
25	Webvan	1680	1000	11.75	15.24		47000	13.31	Infeasible
26	CarsDirect.com	15612	702	33.43	2.14		12885	98.56	2.552160133

Fig. 12.8 Input-oriented VRS variable-benchmark spreadsheet model

set ***B***. At optimality, some DMU$_j$ $j \notin \mathbf{I}_B$, may join the fixed-benchmark set if the associated constraints are binding.

Note that model (12.6) may be infeasible. For example, the DMUs in set ***B*** may not be fit into the same facet when they number greater than m+s−1, where *m* is the number of inputs and *s* is the number of outputs. In this case, we need to adjust the set ***B***.

Three possible cases are associated with model (12.6). $\tilde{\sigma}^{CRS*} > 1$ indicating that DMU^{new} outperforms the benchmark. $\tilde{\sigma}^{CRS*} = 1$ indicating that DMU^{new} achieves the same performance level of the benchmark. $\tilde{\sigma}^{CRS*} < 1$ indicating that the benchmark outperforms DMU^{new}.

By applying RTS frontier type and model orientation, we obtain the fixed benchmark models in Table 12.3

DMU^{new} is not included in the constraints of $\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} + \mu \le 0\left(j \notin \mathbf{I}_B \right)$ $\left(\sum_{i=1}^{m} v_i x_{ij} - \sum_{r=1}^{s} \mu_r y_{rj} + v \ge 0\left(j \notin \mathbf{I}_B \right) \right)$. However, other peer DMUs ($j \notin \mathbf{I}_B$) are included.

Figure 12.9 shows the output-oriented CRS fixed-benchmark spreadsheet model where 1–800-Flowers and Buy.com are two fixed benchmarks. Cells B5:E5 and G5:H5 are reserved for input and output multipliers, respectively. They are the changing cells in the Solver parameters.

Cell C7 is the target cell and contains the formula "=SUMPRODUCT (B5:E5,INDEX(B10:E24,C6,0))", where cell C6 indicates the DMU under evaluation—Autobytel.com.

Cell C8 contains the formula representing $\sum_{r=1}^{s} \mu_r y_r^{new}$

Cell C8=SUMPRODUCT(G5:H5,INDEX (G10:H24,C6,0))

The formula for cell I2 is "=SUMPRODUCT(B2:E2,\$B\$5:\$E\$5)-SUMPRODUCT(G2:H2,\$G\$5:\$H\$5)", and is copied into cells I3 and I10:I24.

Table 12.3 Fixed-benchmark models

Frontier type	Input-oriented	Output-oriented
	$\max \sum_{r=1}^{s} \mu_r y_r^{new} + \mu$	$\min \sum_{i=1}^{m} \nu_i x_i^{new} + \nu$
	subject to	subject to
	$\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{s} \nu_i x_{ij} + \mu = 0 \quad j \in \mathbf{I}_B$	$\sum_{i=1}^{s} \nu_i x_{ij} - \sum_{r=1}^{s} \mu_r y_{rj} + \nu = 0 \quad j \in \mathbf{I}_B$
	$\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{s} \nu_i x_{ij} + \mu \le 0 \quad j \notin \mathbf{I}_B$	$\sum_{i=1}^{s} \nu_i x_{ij} - \sum_{r=1}^{s} \mu_r y_{rj} + \nu \ge 0 \quad j \notin \mathbf{I}_B$
	$\sum_{i=1}^{m} \nu_i x_i^{new} = 1$	$\sum_{r=1}^{s} \mu_r y_r^{new} = 1$
	$\mu_r, \nu_i \ge 0$	$\mu_r, \nu_i \ge 0$
CRS	where $\mu = 0$	where $\nu = 0$
VRS	where μ free	where ν free
NIRS	where $\mu \le 0$	where $\nu \ge 0$
NDRS	where $\mu \ge 0$	where $\nu \le 0$

	A	B	C	D	E	F	G	H	I
1	Company	Visitors	Employee	Marketing	Development		Customers	Revenue	**Constraints**
2	1-800-Flowers	11940	2100	92.15	8.07		7800000	52.89	4.885E-15
3	Buy.com	27372	255	71.3	7.84		1950000	596.9	4.5519E-15
4									
5	Multipliers	3.4E-06	0.001489	0.006619	0		4.843E-07	0	
6	DMU under evaluation		**1**						
7	Score		0.668082						
8	Weighted output		1						
9									
10	Autobytel.com	12000	225	44.18	14.26		2065000	40.3	-0.3319178
11	Beyond.com	17076	250	81.35	10.39		2000000	[illegible]	[illegible].5479E-15
12	eToys	13896	940	120.46	43.43		1900000	[illegible]	[illegible].32392744
13	E*Trade	29532	2400	301.7	78.5		1551000	[illegible]	[illegible].91942618
14	Garden.com	16344	290	16	4.8		1070000	8.2	0.07489194
15	Drugstore.com	19092	408	61.5	14.9		695000	34.8	0.74265995
16	Outpost.com	7716	164	41.67	7		627000	188.6	0.24250252
17	iPrint	42132	225	8.13	3.54		380000	3.26	0.34747683
18	Furniture.com	10668	213	33.949	6.685		260000	10.904	0.45207709
19	PlanetRX.com	17124	390	55.18	12.95		254000	8.99	0.88092163
20	NextCard	46836	365	24.65	22.05		220000	26.56	0.75869065
21	PetsMart.com	18564	72	33.47	2.43		180000	10.45	0.30443706
22	Peapod	2076	1020	7.17	3.54		111900	73.13	1.51906364
23	Webvan	1680	1000	11.75	15.24		47000	13.31	1.54968667
24	CarsDirect.com	15612	702	33.43	2.14		12885	98.56	1.31316454

Weighted Input Target cell Min

Weighted Output = 1

Equality constraint on benchmark

Constraint not included

Constraints for other DMUs

Fig. 12.9 Output-oriented CRS fixed-benchmark spreadsheet model

Figure 12.10 shows the Solver parameters for Autobytel.com. Note that we have "\$I\$2:\$I\$3 = 0" for the two benchmarks. Note also that "\$I\$11:\$I\$24 >=0" does not include the DMU under evaluation, Autobytel.com.

To solve the remaining DMUs, we need to set up different Solver parameters. Because constraints change for each DMU under evaluation. For example, if we change the value of cell C6 to 15, i.e., we benchmark CarsDirect.com, we obtain a

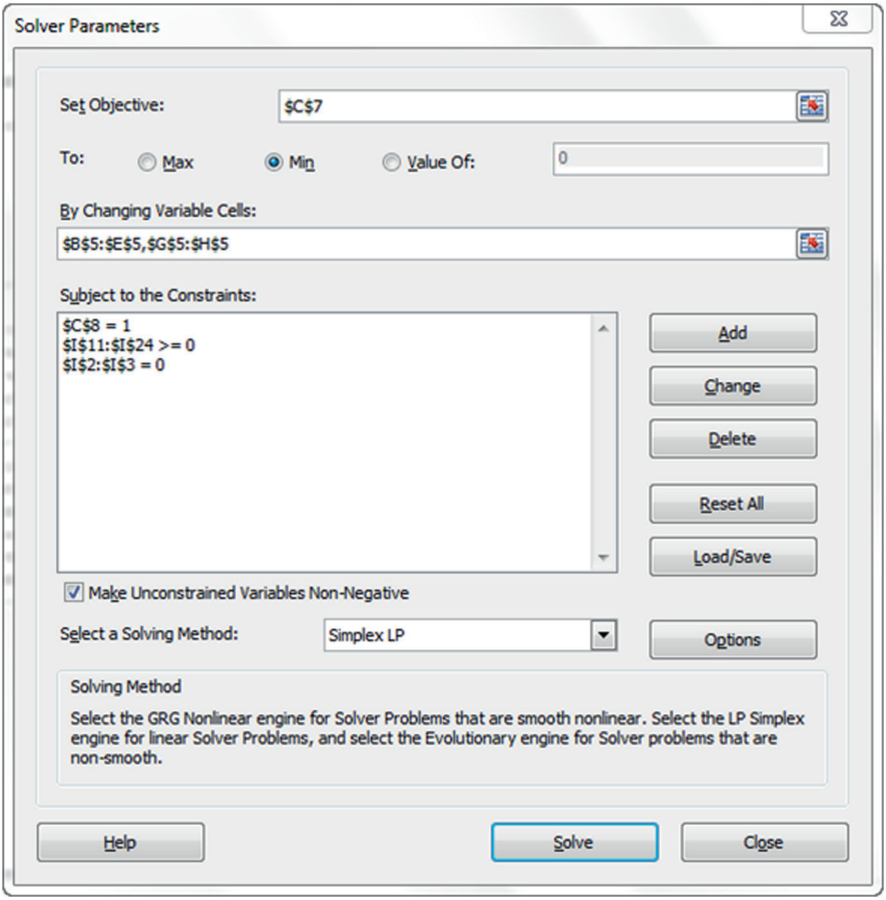


Fig. 12.10 Solver parameters for output-oriented CRS fixed-benchmark model

set of new Solver parameters by removing "$I$24>=0" from the Solver parameters shown in Fig. 12.10 and then adding "$I$10>=0", as shown in Fig. 12.11.

Because different Solver parameters are used for different DMUs under benchmarking, a set of sophisticated VBA codes is required to automate the calculation. We here do not discuss it, and suggest using the "DEAFrontier" software described in the current chapter to obtain the scores (see cells J10:J24 in Fig. 12.11).

12.4 Fixed-benchmark Model and Efficiency Ratio

A commonly used measure of efficiency is the ratio of output to input. For example, profit per employee measures the labor productivity. When multiple inputs and outputs are present, we may define the following efficiency ratio

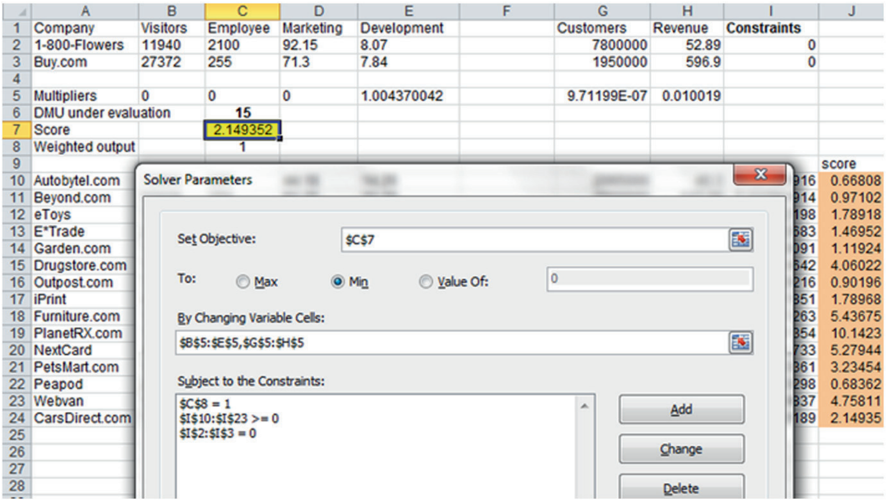


Fig. 12.11 Output-oriented CRS fixed-benchmark scores for internet companies

$$\frac{\sum_{r=1}^{s} u_r y_{ro}}{\sum_{i=1}^{m} v_i x_{io}}$$

where v_i and u_r represent the input and output weights, respectively.

DEA calculate the ratio efficiency without the information on the weights. In fact, the multiplier DEA models can be transformed into linear fractional programming problems. For example, if we define $\nu_i = tv_i$ and $\mu_r = tu_r$, where $t = 1/\sum v_i x_{io}$, the input-oriented CRS multiplier model can be transformed into

$$\begin{aligned}
&\max \frac{\sum_{r=1}^{s} u_r y_{ro}}{\sum_{i=1}^{m} v_i x_{io}} \\
&\text{subject to} \\
&\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \le 1 \quad j = 1, 2, \ldots, n \\
&u_r, v_i \ge 0 \qquad \forall r, i
\end{aligned} \tag{12.7}$$

The objective function in (12.7) represents the efficiency ratio of a DMU under evaluation. Because of the constraints in (12.7), the (maximum) efficiency cannot exceed one. Consequently, a DMU with an efficiency score of one is on the frontier. It can be seen that no additional information on the weights or tradeoffs are incorporated into the model (12.7).

If we apply the input-oriented CRS fixed-benchmark model to (12.7), we obtain

$$\begin{aligned}
&\max \frac{\sum_{r=1}^{s} u_r y_r^{new}}{\sum_{i=1}^{m} v_i x_i^{new}} \\
&\text{subject to} \\
&\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} = 1 \quad j \in \mathbf{I}_{\mathrm{B}} \\
&\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \le 1 \quad j \notin \mathbf{I}_{\mathrm{B}} \\
&u_r, v_i \ge 0 \qquad \forall r, i
\end{aligned} \tag{12.8}$$

It can be seen from (12.8) that the fixed benchmarks incorporate implicit tradeoff information into the efficiency evaluation. i.e., the constraints associated with $\mathbf{I}_{\mathrm{B}}$ can be viewed as incorporation of tradeoffs or weight restrictions in DEA. Model (12.8) yields the (maximum) efficiency under the implicit tradeoff information represented by the benchmarks.

As more DMUs are selected as fixed benchmarks, more complete information on the weights becomes available. For example, if we add FTD.com to the fixed-benchmark set, the benchmarking score for Autobytel.com becomes 1.1395, as shown in Fig. 12.12. As expected, the performance of those internet companies becomes worse when the set of fixed benchmarks expands.

Similarly, the output-oriented CRS fixed-benchmark model is equivalent to

$$\begin{aligned}
&\min \frac{\sum_{i=1}^{m} v_i x_i^{new}}{\sum_{r=1}^{s} u_r y_r^{new}} \\
&\text{subject to} \\
&\frac{\sum_{i=1}^{m} v_i x_{ij}}{\sum_{r=1}^{s} u_r y_{rj}} = 1 \quad j \in \mathbf{I}_{\mathrm{B}}
\end{aligned}$$

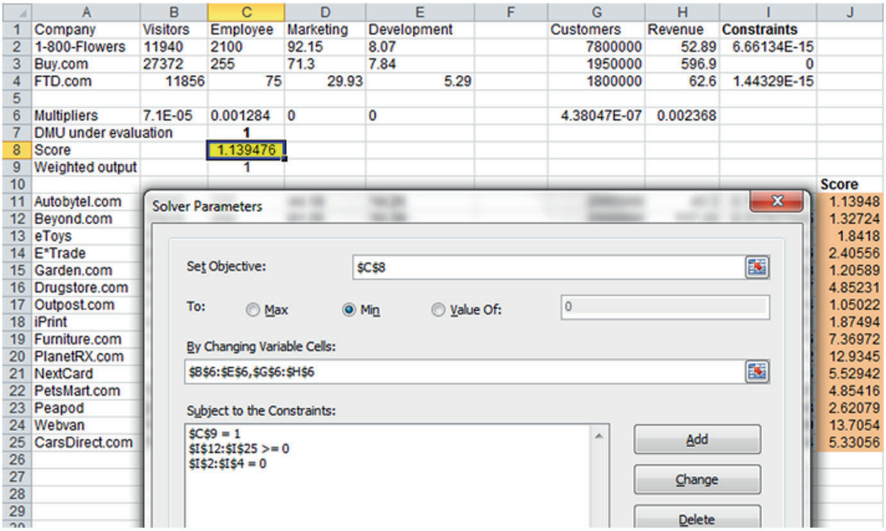


Fig. 12.12 Spreadsheet model and solver parameters for fixed-benchmark model

$$\frac{\sum_{i=1}^{m} v_i x_{ij}}{\sum_{r=1}^{s} u_r y_{rj}} \geq 1 \quad j \notin \mathbf{I}_{\mathrm{B}}$$

$$u_r, v_i \geq 0 \qquad \forall r, i$$

Note that we may define an ideal benchmark whose rth output y_r^{ideal} is the maximum output value across all DMUs, and ith input x_i^{ideal} the minimum input value across all DMUs. If we replace the fixed-benchmark set by the ideal benchmark, we have

$$\begin{aligned}
&\max \frac{\sum_{r=1}^{s} u_r y_r^{new}}{\sum_{i=1}^{m} v_i x_i^{new}} \\
&\text{subject to} \\
&\frac{\sum_{r=1}^{s} u_r y_r^{ideal}}{\sum_{i=1}^{m} v_i x_i^{ideal}} = 1 \\
&\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \leq 1 \\
&u_r, v_i \geq 0 \qquad \forall r, i
\end{aligned} \tag{12.9}$$

Because the ideal benchmark dominates all DMUs (unless DMU$_j$ is one of the ideal benchmark), the optimal value to (12.9) must not be greater than one. Further, $\sum u_r y_{rj} / \sum v_i x_{ij} \le 1$ are redundant, and model (12.9) can be simplified as

$$\begin{aligned}
&\max \frac{\sum_{r=1}^{s} u_r y_r^{new}}{\sum_{i=1}^{m} v_i x_i^{new}} \\
&\text{subject to} \\
&\frac{\sum_{r=1}^{s} u_r y_r^{ideal}}{\sum_{i=1}^{m} v_i x_i^{ideal}} = 1 \\
&u_r, v_i \ge 0 \qquad \forall r, i
\end{aligned} \tag{12.10}$$

Model (12.10) is equivalent to the following linear programming problem

$$\begin{aligned}
&\max \sum_{r=1}^{s} \mu_r y_r^{new} \\
&\text{subject to} \\
&\sum_{r=1}^{s} \mu_r y_r^{ideal} - \sum_{i=1}^{s} v_i x_i^{ideal} = 0 \\
&\sum_{i=1}^{m} v_i x_i^{new} = 1 \\
&\mu_r, v_i \ge 0.
\end{aligned} \tag{12.11}$$

Model (12.10) or (12.11) calculate the maximum efficiency of a specific DMU under evaluation given that the efficiency of the ideal benchmark is set equal to one. If we introduce RTS frontier type and model orientation into (12.10), we obtain other ideal-benchmark models, as shown in Table 12.4.

12.5 Minimum Efficiency Model

Note that the fixed-benchmark models yield the maximum efficiency scores when the tradeoffs are implicitly defined by the benchmarks. If we change the objective function of model (12.8) into minimization, we have

Table 12.4 Ideal-benchmark models

Frontier type	Input-oriented	Output-oriented
	$\max \sum_{r=1}^{s} \mu_r y_r^{new} + \mu$	$\min \sum_{i=1}^{m} v_i x_i^{new} + v$
	subject to	subject to
	$\sum_{r=1}^{s} \mu_r y_r^{ideal} - \sum_{i=1}^{s} v_i x_i^{ideal} + \mu = 0$	$\sum_{i=1}^{s} v_i x_i^{ideal} - \sum_{r=1}^{s} \mu_r y_r^{ideal} + v = 0$
	$\sum_{i=1}^{m} v_i x_i^{new} = 1$	$\sum_{r=1}^{s} \mu_r y_r^{new} = 1$
	$\mu_r, v_i \geq 0.$	$\mu_r, v_i \geq 0$
CRS	where $\mu = 0$	where $v = 0$
VRS	where μ free	where v free
NIRS	where $\mu \leq 0$	where $v \geq 0$
NDRS	where $\mu \geq 0$	where $v \leq 0$

$$\begin{aligned}
&\min \frac{\sum_{r=1}^{s} u_r y_r^{new}}{\sum_{i=1}^{m} v_i x_i^{new}} \\
&\text{subject to} \\
&\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} = 1 \quad j \in \mathbf{I}_{\mathrm{B}} \\
&\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \leq 1 \quad j \notin \mathbf{I}_{\mathrm{B}} \\
&u_r, v_i \geq 0 \qquad \forall r, i
\end{aligned} \tag{12.12}$$

We refer to (12.12) as the input-oriented CRS minimum efficiency model. Although the benchmarks implicitly define the tradeoffs amongst inputs and outputs, the exact tradeoffs are still unavailable to us. Thus, the optimal value to (12.12) gives the lower efficiency bound for DMU^{new}. The optimal value to (12.8) yields the upper efficiency bound. The true efficiency of DMU^{new} lies in-between the bounds.

In fact, model (12.12) describes the worst efficiency scenario whereas model (12.8) describe the best efficiency scenario. The minimum efficiency for the original input-oriented DEA models (e.g., model (12.7)) is zero, and for the original output-oriented DEA models is infinite.

Table 12.5 Minimum efficiency models

Frontier type	Input-oriented	Output-oriented
	$\min \sum_{r=1}^{s} \mu_r y_r^{new} + \mu$ subject to $\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{s} v_i x_{ij} + \mu = 0 \quad j \in \mathbf{I}_B$ $\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{s} v_i x_{ij} + \mu \le 0 \quad j \notin \mathbf{I}_B$ $\sum_{i=1}^{m} v_i x_i^{new} = 1$ $\mu_r, v_i \ge 0$	$\max \sum_{i=1}^{m} v_i x_i^{new} + v$ subject to $\sum_{i=1}^{s} v_i x_{ij} - \sum_{r=1}^{s} \mu_r y_{rj} + v = 0 \quad j \in \mathbf{I}_B$ $\sum_{i=1}^{s} v_i x_{ij} - \sum_{r=1}^{s} \mu_r y_{rj} + v \ge 0 \quad j \notin \mathbf{I}_B$ $\sum_{r=1}^{s} \mu_r y_r^{new} = 1$ $\mu_r, v_i \ge 0$
CRS	where $\mu = 0$	where $v = 0$
VRS	where μ free	where v free
NIRS	where $\mu \le 0$	where $v \ge 0$
NDRS	where $\mu \ge 0$	where $v \le 0$

Similarly, we can obtain the output-oriented CRS minimum efficiency model,

$$
\begin{aligned}
&\max \frac{\sum_{i=1}^{m} v_i x_i^{new}}{\sum_{r=1}^{s} u_r y_r^{new}} \\
&\text{subject to} \\
&\frac{\sum_{i=1}^{m} v_i x_{ij}}{\sum_{r=1}^{s} u_r y_{rj}} = 1 \quad j \in \mathbf{I}_B \\
&u_r, v_i \ge 0 \qquad \forall r, i
\end{aligned}
\tag{12.13}
$$

Recall that a smaller score indicates a better performance in the output-oriented DEA models. Therefore, the output-oriented CRS minimum efficiency score (optimal value to model (12.13) is greater than or equal to the efficiency score obtained from the output-oriented CRS fixed-benchmark model.

The linear program equivalents to (12.12) and (12.13) are presented in Table 12.5 which summarizes the minimum efficiency models.

The spreadsheet models for the minimum efficiency models are similar to the fixed-benchmark spreadsheet models. We only need to change the "Max" to "Min" in the Solver parameters for the input-oriented models, and change the "Min" to "Max" for the output-oriented models. For example, consider the output-oriented

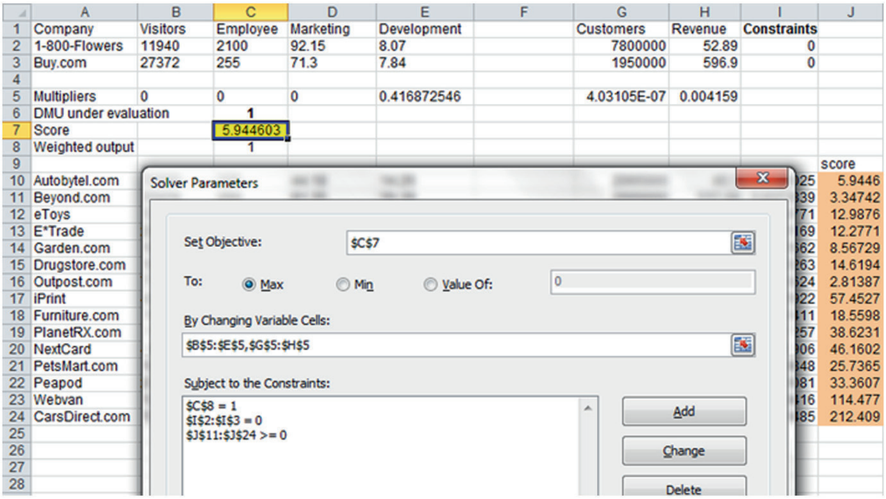


Fig. 12.13 Output-oriented CRS minimum efficiency spreadsheet model

CRS fixed-benchmark model shown in Fig. 12.9. Figure 12.13 shows the corresponding minimum efficiency spreadsheet model.

Under the tradeoffs characterized by the two benchmarks, the true efficiency of Autobytel.com lies in [0.6681, 5.9446]. Cells J10:J24 report the "minimum efficiency" for the 15 internet companies. The scores are calculated by the DEAFrontier software.

If we introduce the ideal benchmark into the minimum efficiency models, we obtain, for example, the input-oriented VRS ideal-benchmark minimum efficiency model

$$
\begin{aligned}
&\min \sum_{r=1}^{s} \mu_r y_r^{new} + \mu \\
&\text{subject to} \\
&\sum_{r=1}^{s} \mu_r y_r^{ideal} - \sum_{i=1}^{s} v_i x_i^{ideal} + \mu = 0 \\
&\sum_{i=1}^{m} v_i x_i^{new} = 1 \\
&\mu_r, v_i \geq 0 \text{ and } \mu \text{ free in sign}
\end{aligned}
\tag{12.14}
$$

Table 12.6 presents the ideal-benchmark minimum efficiency models.

Table 12.6 Ideal-benchmark minimum efficiency models

Frontier type	Input-oriented	Output-oriented
	$\min \sum_{r=1}^{s} \mu_r y_r^{new} + \mu$	$\max \sum_{i=1}^{m} \nu_i x_i^{new} + \nu$
	subject to	subject to
	$\sum_{r=1}^{s} \mu_r y_r^{ideal} - \sum_{i=1}^{s} \nu_i x_i^{ideal} + \mu = 0$	$\sum_{i=1}^{s} \nu_i x_i^{ideal} - \sum_{r=1}^{s} \mu_r y_r^{ideal} + \nu = 0$
	$\sum_{i=1}^{m} \nu_i x_i^{new} = 1$	$\sum_{r=1}^{s} \mu_r y_r^{new} = 1$
	$\mu_r, \nu_i \geq 0.$	$\mu_r, \nu_i \geq 0$
CRS	where $\mu = 0$	where $\nu = 0$
VRS	where μ free	where ν free
NIRS	where $\mu \leq 0$	where $\nu \geq 0$
NDRS	where $\mu \geq 0$	where $\nu \leq 0$

12.6 Buyer-seller Efficiency Model

As pointed out by Wise and Morrison (2000), one of the major flaws in the current business-to-business (B2B) model is that it focuses on price-driven transactions between buyers and sellers, and fails to recognize other important vendor attributes such as response time, quality and customization. In fact, a number of efficiency-based negotiation models have been developed to deal with multiple attributes—inputs and outputs. For example, DEA is used by Weber and Desai (1996) to develop models for vendor evaluation and negotiation. The fixed-benchmark models and the minimum efficiency models can better help the vendor in evaluating and selecting the vendors.

Talluri (2002) proposes a buyer-seller game model that evaluates the efficiency of alternative bids with respect to the ideal target set by the buyer. Zhu (2004) shows that this buyer-seller game model is closely related to DEA and can be simplified as the models presented in Tables 12.4 and 12.6.

We next use the data in Table 5.1 to demonstrate the use of DEA benchmarking models. A Fortune 500 pharmaceutical company was involved in the implementation of a Just-in-Time manufacturing system. Therefore, price, delivery performance, and quality were considered to be the three most important criteria in evaluating and selecting vendors. In Weber and Desai (1996), the price criterion is measured by the total purchase price based on a per unit contract delivered price, the delivery criterion is measured by the percentage of late deliveries, and the quality criterion is measured by the percentage of units rejected. Obviously, the measures for delivery and quality are bad outputs. Therefore, we re-define the delivery and quality by percentage on-time deliveries and percentage of accepted units, respectively. (Otherwise, we should use the method described in Chap. 5.)

Table 12.7 Data for the six vendors

Vendor	Price ($/unit)	% accepted units	% on-time deliveries
1	0.1958	98.8	95
2	0.1881	99.2	93
3	0.2204	100	100
4	0.2081	97.9	100
5	0.2118	97.7	97
6	0.2096	98.8	96

Table 12.8 Input-oriented CRS efficiency and efficient target for vendors

Vendor	Efficiency	Price ($/units)	% acceptance	% on-time deliveries
1	0.981	0.192145	*101.3333*	95
2	1	0.1881	99.2	93
3	0.918	0.202258	*106.6667*	100
4	0.972	0.202258	*106.6667*	100
5	0.926	0.19619	*103.4667*	97
6	0.926	0.194168	*102.4*	96

The results are based upon the input-oriented CRS envelopment model

Table 12.7 presents the data for six vendors that are obtained from the data presented in Table 5.1. The second column reports the input, and the third and forth columns report the two outputs. We next need to determine the frontier type. Because the outputs are measured in percentages, we assume the vendors form a VRS frontier. Otherwise, unreasonable results may be obtained if we assume CRS frontier. For example, Table 12.8 reports the input-oriented CRS efficiency scores (second column) with the efficient targets. It can be seen that the efficient targets on percentage of accepted units are impossible to achieve.

If we use the input-oriented VRS envelopment model, vendors 2, 3, and 4 are efficient, and can be selected. However, if we specify an ideal benchmark by the minimum input value and the maximum output values, as shown in Fig. 12.14, we can further characterize the six vendors.

Figure 12.14 shows the spreadsheet for the input-oriented VRS ideal-benchmark model. Cell B4 and cells D4:E4 are reserved for the input and output multipliers. The free variable is represented by cell G3 which contains the formula "=F4-G4". Cells F4:G4 are specified as changing cells in the Solver parameters (see Fig. 12.15).

Cell F2 contains the formula for the ideal benchmark, that is
Cell F2=SUMPRODUCT(D2:E2,D4:E4)-B2*B4+G3
Cell C5 is reserved to indicate the vendor under evaluation. The (maximum) efficiency is presented in cell C6 which contains the formula
Cell C6=SUMPRODUCT(D4:E4,INDEX(D9:E14,C5,0))+G3
Cell C7 is the weighted input and contains the formula
Cell C7=B4*INDEX(B9:B14,C5,1)

	A	B	C	D	E	F	G
1		Price ($/units)		% accepted units	% on-time deliveries	constraint	free variable= F4-G4
2	Ideal target	0.1881		100	100	0	
3						free variable	0.897424
4	multipliers	4.770992		0	0	0.8974237	0
5	DMU under evaluation		**6**				
6	Score		0.8974				
7	Weighted input		1			Maximum	
8						Efficiency	
9	Vendor 1	0.1958		98.8	95	0.9606742	
10	Vendor 2	0.1881		99.2	93	1	
11	Vendor 3	0.2204		100	100	0.8534483	Max
12	Vendor 4	0.2081		97.9	100	0.9038924	
13	Vendor 5	0.2118		97.7	97	0.888102	
14	Vendor 6	0.2096		98.8	96	0.8974237	

Fig. 12.14 Input-oriented VRS Ideal-benchmark spreadsheet model

The Solver parameters shown in Fig. 12.15 remain the same for all the vendors, and the calculation is performed by the VBA procedure “IdealBenchmark”.

```
Sub IdealBenchmark()
Dim i As Integer
For i = 1 To 6
Range("C5") = i
SolverSolve UserFinish:=True
Range("F" & i + 8) = Range("C6")
Next
End Sub
```

Based upon the scores in cells F9:F14 in Fig. 12.14, vendor 2 has the best performance.

Next, we turn to the ideal-benchmark minimum efficiency model (12.14). The spreadsheet is the same as the one shown in Fig. 12.14. However, we need to change “Max” to “Min” in the Solver parameters shown in Fig. 12.15. Figure 12.16 shows the result. Figure 12.17 shows the minimum efficiency scores in cells F9:F14. The minimum efficiency model also indicates that vendor 2 is the best one.

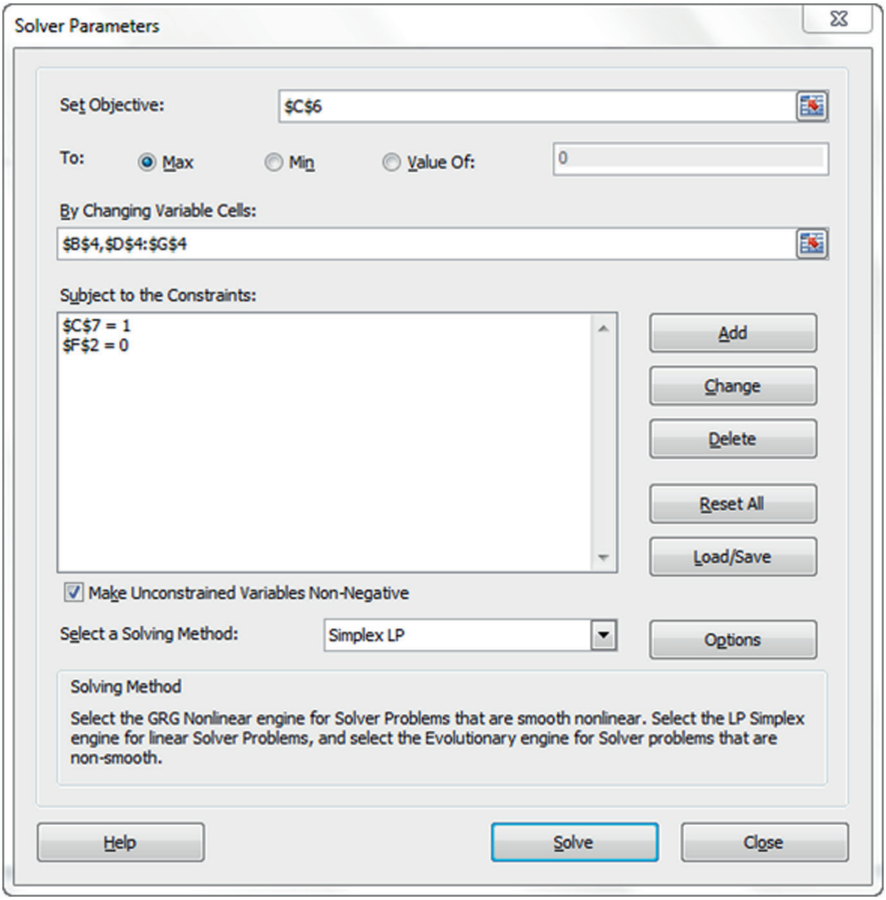


Fig. 12.15 Solver parameters for input-oriented VRS ideal-benchmark model

12.7 Acceptance System Decision

Troutt et al. (1996) propose the use of DEA in case or example based decision systems. Their discussion is based upon three assumptions: (1) conditional monotonicity of all variables; (2) convexity of the acceptable set; and (3) the cases contain no Type II errors. Their use of a DEA/acceptability frontier is as follows. First, some example acceptable cases are decided by one or more experts. Second, the DEA frontier of these sample cases is determined. Finally, a DEA fractional programming problem is established for a new case (with this new case included in the fractional restrictions). Some rules are proposed for determining whether a new case is acceptable. However, the decision of acceptance/rejection is actually determined by a linear programming model with an arbitrary objective function. If the linear programming problem is feasible, the case is accepted, and otherwise it is rejected.

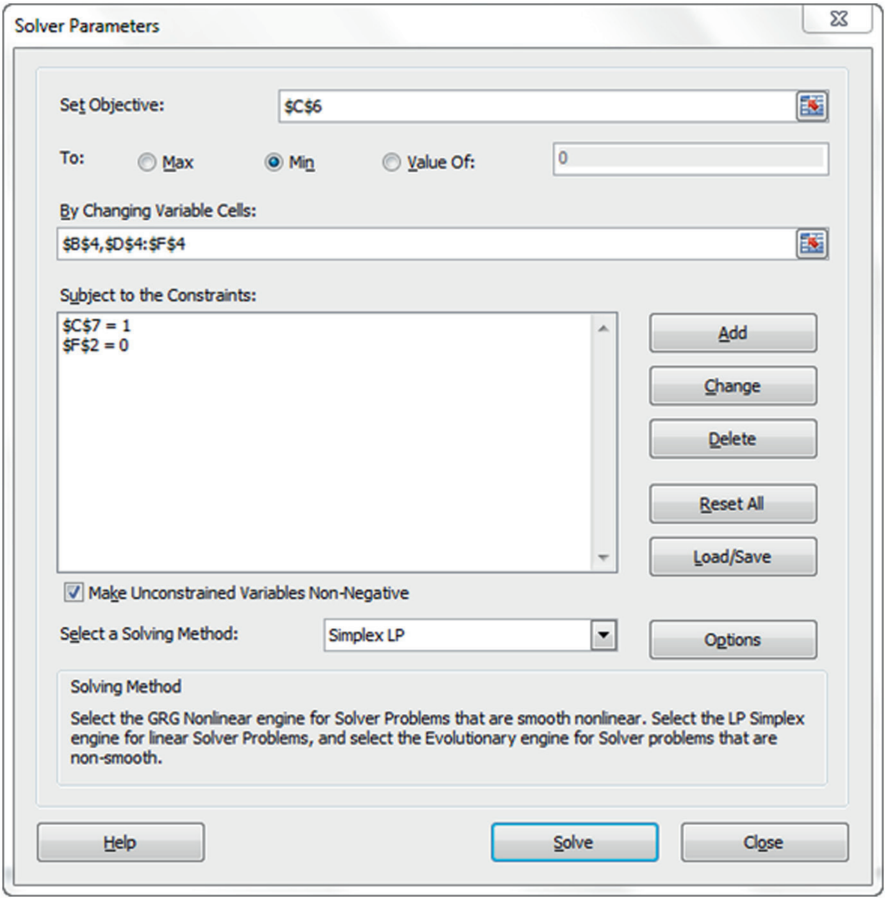


Fig. 12.16 Solver parameters for VRS ideal-benchmark minimum efficiency model

	A	B	C	D	E	F	G
1		Price ($/units)		% accepted units	% on-time deliveries	constraint	
2	Ideal target	0.1881		100	100	-1.11E-16	
3						free variable	0
4	multipliers	4.770992		0	0.008974	0	0
5	DMU under evaluation		**6**				
6	Score		0.8615				
7	Weighted input		1			Minimum	
8						Efficiency	
9	Vendor 1	0.1958		98.8	95	0.9126404	
10	Vendor 2	0.1881		99.2	93	0.93	Min
11	Vendor 3	0.2204		100	100	0.8534483	
12	Vendor 4	0.2081		97.9	100	0.8849106	
13	Vendor 5	0.2118		97.7	97	0.8614589	
14	Vendor 6	0.2096		98.8	96	0.8615267	

Fig. 12.17 Minimum efficiency scores for the six vendors

Seiford and Zhu (1998) show that their linear programming problem, in fact, can be obtained directly from their assumptions. Seiford and Zhu (1998) further provide a DEA type linear programming problem so that a new case can be identified whether it is located within, above, or below the sample frontier. Seiford and Zhu's (1998) model determines not only whether a new case is accepted, but also the location of the new case with respect to the samples previously determined. In fact, this type of model is closely related to the benchmarking models discussed in the current chapter. A reader will see that the acceptance decision model in Troutt et al. (1996) and Seiford and Zhu (1998) can be regarded as the basic model of our benchmarking models. To further illustrate this, we suppose $\mathbf{x}_j \in E^*$ are the data vectors of acceptable cases which have previously been decided. (This can be regarded as the benchmarks.) On the basis of the convexity assumption, the following case must be acceptable:

$$\overline{\mathbf{x}} = \sum_j \lambda_j \mathbf{x}_j = \begin{pmatrix} \sum_j \lambda_j \mathbf{x}_{1j} \\ \vdots \\ \sum_j \lambda_j \mathbf{x}_{ij} \\ \vdots \\ \sum_j \lambda_j \mathbf{x}_{mj} \end{pmatrix}$$

where $\sum_j \lambda_j = 1$ and $\lambda_j \geq 0$.

Let $\mathbf{x}^{new}$ be a new case. Then by the conditional monotonicity assumption we have that $\mathbf{x}^{new}$ is also acceptable if

$$\overline{\mathbf{x}}_i = \sum_j \lambda_j \mathbf{x}_{ij} \leq x_i^{new}, \text{ for all i, j, } x_{ij} \in E^*, \text{ and } \sum_j \lambda_j = 1, \lambda_j \geq 0. \tag{12.15}$$

Obviously, whether (12.15) holds depends on whether we can find a solution of λ_j. One possible way is that, as developed in Troutt et al. (1996), we check for a feasible solution to the following linear programming problem with an arbitrary objective $L(\lambda)$:

$$\begin{aligned} & \max L(\lambda) \\ & s.t. \sum_j \lambda_j \mathbf{x}_{ij} \leq x_i^{new}, \text{ for all i, j, } \mathbf{x}_{ij} \in E^* \\ & \sum_j \lambda_j = 1 \\ & \lambda_j \geq 0 \text{ for all j.} \end{aligned} \tag{12.16}$$

Table 12.9 Hypothetical cases

Case	1	2	3	4	5	6	7	8	9
x1	2	2	3	6	3	2	5	2	4
x2	6	4	2	1	2.5	5	2	2	0

Thus, if (12.16) is feasible, then (12.15) has a solution λ_j. Furthermore, if (12.16) is feasible then the new case is accepted, and otherwise it is rejected.

In order to fully incorporate DEA into a decision rule for the acceptance/rejection of credit risks, Seiford and Zhu (1998) provide the following DEA type linear programming model on the basis of $x_j \in E^*$:

$$
\begin{aligned}
& L^* = \min L \\
& s.t. \sum_j \lambda_j \mathbf{x}_{ij} \le L x_i^{new}, \text{ for all i, j, } \mathbf{x}_{ij} \in E^* \\
& \sum_j \lambda_j = 1 \\
& \lambda_j \ge 0 \text{ for all j, and } L \text{ is free}
\end{aligned}
\tag{12.17}
$$

The major difference between (12.16) and (12.17) lies in the fact that the objective function is no longer arbitrary in (12.17).

Obviously, model (12.17) is a (VRS) variable-benchmark model with an output value of unity. Seiford and Zhu (1998) provide the following theorem for (12.17):

Theorem 12.5 For a new case$\mathbf{x}^{new}$, we have:

1. L* > 1 or (12.17) is infeasible if and only if the new case alters E^*;
2. L* = 1 if and only if the new case lies within E^*;
3. L* < 1 if and only if the new case lies above the frontier determined by the cases in E^*.

The above theorem identifies the location of the new case, $\mathbf{x}^{new}$, with respect to the acceptable sample cases in E^*.

Furthermore, Seiford and Zhu (1998) show:

Theorem 12.6 A new case$\mathbf{x}^{new}$ is acceptable *if and only if* $L^* \le 1$, where L^* is the optimal value to (12.17).

After the acceptable cases of E^* have been determined by expert opinion, one solves (12.17) for a new case $\mathbf{x}^{new}$. If $L^* \le 1$, then the case is accepted, and otherwise it is rejected.

Consider the following nine-case example with two variables $\mathbf{x}_1$ and $\mathbf{x}_2$ (Table 12.9). By the CRS DEA model (with an output value of one), we find that cases 1, 2, 3, and 4 are on DEA frontier. i.e., $E^* = \{\text{case 1, case 2, case 3, case 4}\}$. The frontier determined by the cases in E^* is thus ABCD (Fig. 12.18).

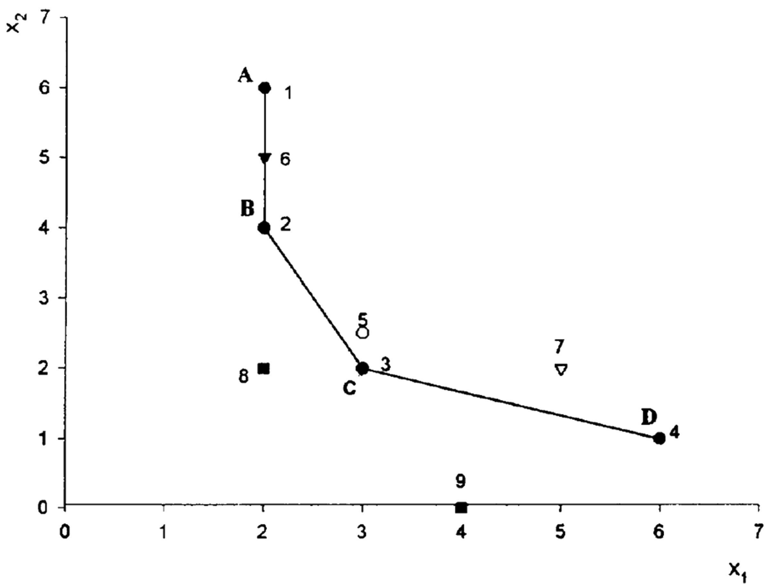


Fig. 12.18 Acceptance system

Next, four new cases, namely, cases 6, 7, 8, and 9, are presented. Solving (12.17) for each of the four new cases yields:

case 6: $L^*=1$, $\lambda_2^* = 1$ (with slack);
case 7: $L^*=9/11$, $\lambda_3^* = 7/11$, $\lambda_4^* = 4/11$;
case 8: $L^*=4/3$, $\lambda_2^* = 1/3$, $\lambda_3^* = 2/3$;
case 9: infeasible.

Therefore, cases 6 and 7 are acceptable and cases 8 and 9 are not. In fact, from Fig. 12.18, we see that case 6 lies on the frontier ABCD and case 7 lies above the frontier. Cases 8 and 9 lies below the frontier, and will alter E^* if they are included as acceptable cases.

12.8 Solving Benchmarking Models Using DEAFrontier

12.8.1 Variable-Benchmark Models

To run the variable-benchmark models presented in Table 12.1, we need set up the data sheets. *Store the benchmarks in a sheet named "Benchmarks" and the DMUs under evaluation in a sheet named "DMUs"*. The format for these two sheets is the same as that shown in Fig. 12.3. Then select the Variable Benchmark Model menu

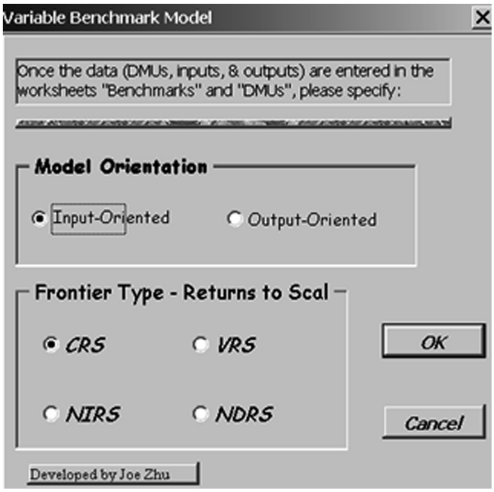


Fig. 12.19 Variable benchmark models

item. You will be prompted a form for selecting the model orientation and the frontier type as shown in Fig. 12.19. Note that if you select a frontier type other than CRS, the results may be infeasible. The benchmarking results are reported in the sheet "Benchmarking Results".

12.8.2 Fixed-Benchmark Models

To run the fixed-benchmark models presented in Table 12.3, we *store the benchmarks in a sheet named "Benchmarks" and the DMUs under evaluation in a sheet named "DMUs"*. Then select the Fixed-Benchmark Model menu item. You will be prompted a form for selecting the model orientation and the frontier type. The results are reported in the "Efficiency Report" sheet. If the benchmarks are not properly selected, you will have infeasible results and need to adjust the benchmarks.

The Ideal-benchmark Models in Table 12.4 should be calculated using the Fixed-Benchmark Model menu item. The data for the ideal benchmark is stored in the "Benchmarks" sheet.

12.8.3 Minimum Efficiency Models

To run the minimum efficiency models presented in Table 12.5, we *store the benchmarks in a sheet named "Benchmarks" and the DMUs under evaluation in a sheet named "DMUs"*. Then select the Minimum Efficiency Model menu item. You will be prompted a form for selecting the model orientation and the frontier type. The results are reported in the "Minimum Efficiency" sheet.

The Ideal-benchmark Minimum Efficiency Models in Table 12.6 should be calculated using the Minimum Efficiency menu item. The data for the ideal benchmark is stored in the "Benchmarks" sheet.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_12) contains supplementary material, which is available to authorized users.

References

Cook, W. D., Seiford, L. M., & Zhu, J. (2004). Models for performance benchmarking: Measuring the effect of e-commerce activities on banking performance. *OMEGA, 32*(4), 313–322.

Seiford, L. M., & Zhu, J. (1998). An acceptance system decision rule with data envelopment analysis. *Computers and Operations Research, 25*(4), 329–332.

Seiford, L. M., & Zhu, J. (1999). Infeasibility of super efficiency data envelopment analysis models. *INFOR, 37*(2), 174–187.

Talluri, S. (2002). A buyer–seller game model for selection and negotiation of purchasing bids. *European Journal of Operational Research, 143*(1), 171–180.

Troutt, M. D., Rai, A., & Zhang, A. (1996). The potential use of DEA for credit applicant acceptance systems. *Computers & Operations Research, 23,* 405–408.

Weber, C. A., & Desai, A. (1996). Determination of paths to vendor market efficiency using parallel coordinates representation: A negotiation tool for buyers. *European Journal of Operational Research, 90,* 142–155.

Wise, R., & Morrison, D. (2000). Beyond the exchange: the future of B2B. *Harvard Business Review, 78*(6), 86–96.

Zhu, J. (1996). Robustness of the efficient DMUs in data envelopment analysis. *European Journal of Operational Research, 90*(3), 451–460.

Zhu, J. (2004). A buyer-seller game model for selection and negotiation of purchasing bids: Extensions and new models. *European Journal of Operational Research, 154*(1), 150–156.

Chapter 13
Returns-to-Scale

13.1 Introduction

As demonstrated in Fig. 2.3, the VRS envelopment model identifies the VRS frontier with DMUs exhibiting IRS (increasing returns to scale), CRS (constant returns to scale), and DRS (decreasing returns to scale). In fact, the economic concept of RTS (returns to scale) has been widely studied within the framework of DEA. RTS have typically been defined only for single output situations. DEA generalizes the notion of RTS to the multiple-output case. This, in turn, further extended the applicability of DEA.

Seiford and Zhu (1999a) demonstrate that there are at least three equivalent *basic* methods of testing a DMU's RTS nature which have appeared in the DEA literature. Based upon the VRS multiplier models, the sign of the optimal free variable (μ^* or v^*) indicates the RTS (Banker et al. 1984). Based upon the CRS envelopment models, the magnitude of optimal $\sum_j^n \lambda_j^*$ indicates the RTS (Banker 1984). These two methods may fail when DEA models have alternate optimal solutions. The third method is based upon the scale efficiency index (Färe et al. 1994). The scale efficiency index method does not require information on μ^* or v^* or $\sum_j^n \lambda_j^*$, and is robust even when there exist multiple optima. However, the scale efficiency index method requires the calculation of three DEA models.

Seiford and Zhu (1999b) and Seiford and Zhu (2005) study the sensitivity of RTS classification. Seiford and Zhu (1999c) provide a use of RTS sensitivity analysis in improving performance of a two-stage process.

13.2 RTS Regions

It is meaningful to discuss RTS for DMUs located on the VRS frontier. We discuss the RTS for non-frontier DMUs by their VRS efficient targets as indicated in Table 2.2. Because a VRS envelopment model can be either input-oriented or output-oriented, we may obtain different efficient targets and RTS classifications for a specific non-frontier DMU.

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_13,

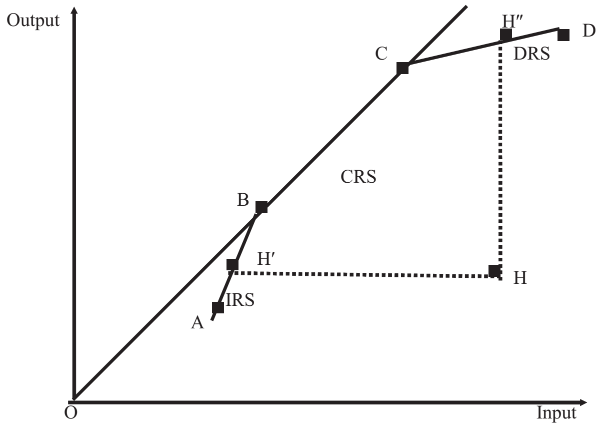


Fig. 13.1 RTS and VRS efficient target

Suppose we have five DMUs, A, B, C, D, and H as shown in Fig. 13.1. Ray OBC is the CRS frontier. AB, BC and CD constitute the VRS frontier, and exhibit IRS, CRS and DRS, respectively. B and C exhibit CRS. On the line segment AB, IRS prevail to the left of B. On the line segment CD, DRS prevail to the right of C.

Consider non-frontier DMU H. If the input-oriented VRS envelopment model is used, then H′ is the efficient target, and the RTS classification for H is IRS. If the output-oriented VRS envelopment model is used, then H″ is the efficient target, and the RTS classification for H is DRS.

However some IRS, CRS and DRS regions are uniquely determined no matter which VRS model is employed. They are region 'I'—IRS, region 'II'—CRS, and region 'III'—DRS. In fact, we have six RTS regions as shown in Fig. 13.2. Two RTS classifications will be assigned into the remaining regions IV, V and VI. Region 'IV' is of IRS (input-oriented) and of CRS (output-oriented). Region 'V' is of CRS (input-oriented) and of DRS (output-oriented). Region 'VI' is of IRS (input-oriented) and of DRS (output-oriented).

The RTS regions can provide a DMU classification. See also Gregoriou and Zhu (2005).

13.3 RTS Estimation

13.3.1 VRS and CRS RTS Methods

Let μ^* represent the optimal value of μ in the input-oriented VRS multiplier model, and ν^* the optimal value of ν in the output-oriented VRS multiplier model, then we have the VRS RTS method.

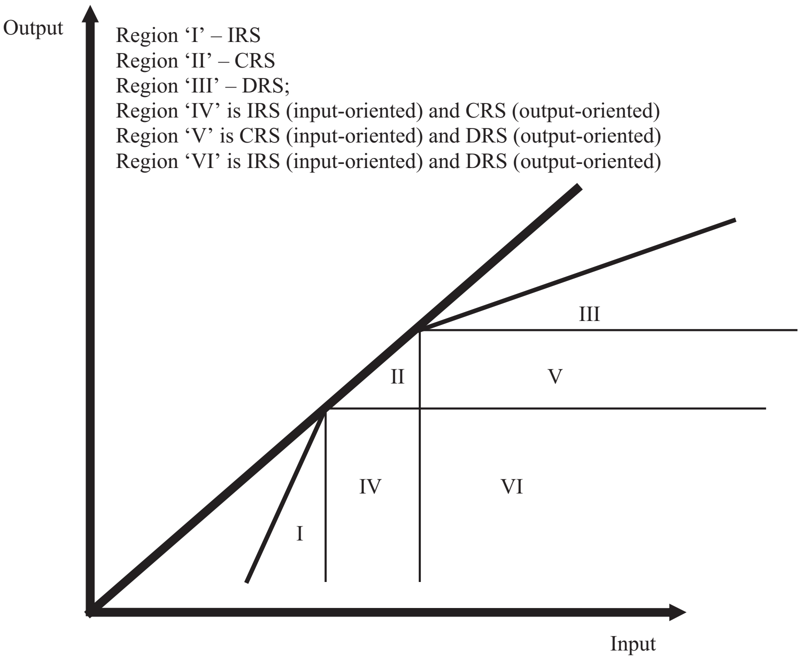


Fig. 13.2 RTS region

Theorem 13.1:

1. If $\mu^* = 0$ (or $v^* = 0$) in *any* alternate optima, then CRS prevail on DMU_o.
2. If $\mu^* > 0$ (or $v^* < 0$)in *all* alternate optima, then IRS prevail on DMU_o.
3. If $\mu^* < 0$ (or $v^* > 0$)in *all* alternate optima, then DRS prevail on DMU_o.

Note that the VRS frontier can be expressed as $\sum_{r=1}^{s} \mu_r y_{rj} = \sum_{i=1}^{m} v_i x_{ij} - \mu \left(\text{or} \sum_{r=1}^{s} \mu_r y_{rj} = \sum_{i=1}^{m} v_i x_{ij} + v \right)$. Thus, geometrically, in the case of single output, $-\mu^*$ (or v^*) represents the y-intercept on the output axis. Consider Fig. 13.1. The intercept is positive for line segment CD so $\mu^* < 0$ (or $v^* > 0$) and RTS is decreasing for any DMU on CD (excluding C), whereas the intercept is negative for line segment AB so $\mu^* > 0$ (or $v^* < 0$) and RTS is increasing for any DMU on AB (excluding B). The intercept for line OBC is zero so $\mu^* = 0$ (or $v^* = 0$) and RTS is constant. However, in computation, we may not obtain the unique optimal solution (the frontier), and we may obtain supporting hyperplanes at VRS frontier DMUs. Consequently, we have to check all optimal solutions as indicated in Theorem 13.1.

Table 13.1 presents five VRS frontier DMUs with two inputs and one output. The last column indicates the RTS classification

Table 13.1 DMUs for RTS estimation

DMU	Input 1 (x1)	Input 2 (x2)	Output (y)	RTS
1	2	5	2	CRS
2	2	2	1	CRS
3	4	1	1	CRS
4	2	1	1/2	IRS
5	6	5	5/2	DRS

Table 13.2 Optimal values for RTS estimation

DMU	$\mu^* \in \left[\mu^-, \mu^+\right]$	λ_j^*
1	[−7, 1]	$\lambda_1^* = 1; \sum_{j=1}^{6} \lambda_j^* = 1$
2	[0, 1]	solution 1: $\lambda_2^* = 1; \sum_{j=1}^{6} \lambda_j^* = 1$
		solution 2: $\lambda_1^* = 1/3, \lambda_3^* = 1/3; \sum_{j=1}^{6} \lambda_j^* = 2/3$
3	[−5/3, 1]	$\lambda_3^* = 1; \sum_{j=1}^{6} \lambda_j^* = 1$
4	[1/2, 1]	$0 \le \lambda_1^* \le 1/12, \lambda_2^* = 1/4 - 3\lambda_1^*, \lambda_3^* = 1/4 - \lambda_1^* 5/12 \sum_{j=1}^{6} \lambda_j^* \le 1/2$
5	[−∞, −3/37]	$\lambda_1^* = 35/48 - \lambda_2^*/3, 0 \le \lambda_2^* \le 35/16, \lambda_3^* = 25/24 - \lambda_2^*/385/48 \sum_{j=1}^{6} \lambda_j^* \le 15/6$

The second column of Table 13.2 reports the optimal μ^*. μ^* can take all the optimal values in the interval $\left[\mu^-, \mu^+\right] \cdot \mu^* = 0$ is found in DMUs 1, 2, and 3, therefore the three DMUs exhibit CRS. All μ^* are positive and negative in DMU5 and DMU6, respectively, therefore IRS and DRS prevail on DMU5 and DMU6, respectively.

The above RTS method uses the VRS multiplier models. In fact, we can use CRS envelopment models to estimate the RTS classification (Zhu 2000a). Let λ_j^* be the optimal values in CRS envelopment models. We have

Theorem 13.2:

4. If $\sum_j^n \lambda_j^* = 1$ in *any* alternate optima, then CRS prevail on DMU_o.
5. If $\sum_j^n \lambda_j^* < 1$ for *all* alternate optima, then IRS prevail on DMU_o.
6. If $\sum_j^n \lambda_j^* > 1$ for *all* alternate optima, then DRS prevail on DMU_o.

From Table 13.2, we see that DMU2 has alternate optimal λ_j^*. Nevertheless, there exists an optimal solution such that $\sum_j^n \lambda_j^* = 1$ indicating CRS. DMU4 exhibits IRS because $\sum_j^n \lambda_j^* < 1$ in all optima, and DMU5 exhibits DRS because $\sum_j^n \lambda_j^* > 1$ in all optima.

13.3.2 Improved RTS Method

In real world applications, the examination of alternative optima is a laborious task, and one may attempt to use a single set of resulting optimal solutions in the application of the RTS methods. However, this may yield erroneous results. For instance, if we obtain $\lambda_1^* = \lambda_3^* = 1/3$, or $\mu^* = 1$ for DMU2, then DMU2 may erroneously be classified as having IRS because $\sum \lambda_j^* < 1$ or $\mu^* > 0$ in one particular alternate solution.

A number of methods have been developed to deal with multiple optimal solutions in the VRS multiplier models and the CRS envelopment models. Seiford and Zhu (1999a) show the following results with respect to the relationship amongst envelopment and multiplier models, respectively.

Theorem 13.3:

1. The CRS efficiency score is equal to the VRS efficiency score *if and only if* there exists an optimal solution such that $\sum_j^n \lambda_j^* = 1$. If The CRS efficiency score is not equal to the VRS efficiency score, then
2. The VRS efficiency score is greater than the NIRS efficiency score *if and only if* $\sum_j^n \lambda_j^* < 1$ in all optimal solutions of the CRS envelopment model.
3. The VRS efficiency score is equal to the NIRS efficiency score *if and only if* $\sum_j^n \lambda_j^* > 1$ in all optimal solutions of the CRS envelopment model.

Theorem 13.4:

1. The CRS efficiency score is equal to the VRS efficiency score *if and only if* there exists an optimal solution $\mu^* = 0$ (or $v^* = 0$). If The CRS efficiency score is not equal to the VRS efficiency score, then
2. The VRS efficiency score is greater than the NIRS efficiency score *if and only if* $\mu^* > 0$ (or $v^* < 0$) in all optimal solutions.
3. The VRS efficiency score is equal to the NIRS efficiency score *if and only if* $\mu^* < 0$ (or $v^* > 0$) in all optimal solutions.

Based upon Theorems 13.3 and 13.4, we have

Theorem 13.5:

1. If DMU_o exhibits IRS, then $\sum_j^n \lambda_j^* < 1$ for *all* alternate optima.
2. If DMU_o exhibits DRS, then $\sum_j^n \lambda_j^* > 1$ for *all* alternate optima.

The significance of Theorem 13.5 lies in the fact that the possible alternate optimal λ_j^* obtained from the CRS envelopment models only affect the estimation of RTS for those DMUs that truly exhibit CRS, and have nothing to do with the RTS estimation on those DMUs that truly exhibit IRS or DRS. That is, if a DMU exhibits IRS (or DRS), then $\sum_j^n \lambda_j^*$ must be less (or greater) than one, no matter whether there exist alternate optima of λ_j.

Further, we can have a very simple approach to eliminate the need for examining all alternate optima.

Theorem 13.6:

1. The CRS efficiency score is equal to the VRS efficiency score *if and only if* CRS prevail on DMU_o. Otherwise,

	A	B	C	D	E	F	G	H	I	J	K
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	CRS Efficiency	Σλ
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		0	0.662831738	1.101942
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0	1	1
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	1	1
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0	1	1
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0.47	1	1
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	0.971966637	0.956252
8	Ford Motor	243283	24547	346990		137137	4139		0	0.737166307	0.99751
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	0.524557613	0.819264
10	Exxon	91296	40436	82000		110009	6470		0	1	1
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	0.841423731	1.067172
12	Wal-Mart	37871	14762	675000		93627	2740		0.01	1	1
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	0.386057261	0.62692
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0	1	1
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0	0.348577853	0.619252
16	AT&T	88884	17274	299300		79609	139		0	0.270381772	0.481746
17											
18		Reference		DMU under	15	Efficiency					
19	**Constraints**	set		Evaluation		0.270382					
20	Assets	24032.61	≤	24032.613							
21	Equity	3338.645	≤	4670.5747		Input-oriented					
22	Employees	12922.98	≤	80925.264		CRS (RTS)					
23	Revenue	79609	≥	79609							
24	Profit	139	≥	139							
25	Σλ	0.481746									

Fig. 13.3 Input-oriented RTS classification spreadsheet model

2. $\sum_j^n \lambda_j^* < 1$ *if and only if* IRS prevail on DMU_o.
3. $\sum_j^n \lambda_j^* > 1$ *if and only if* DRS prevail on DMU_o.

Thus, in empirical applications, we can explore RTS in two steps. First, select all the DMUs that have the same CRS and VRS efficiency scores regardless of the value of $\sum_j^n \lambda_j^*$. These DMUs are in the CRS region. Next, use the value of $\sum_j^n \lambda_j^*$ (in any CRS envelopment model outcome) to determine the RTS for the remaining DMUs. We observe that in this process we can safely ignore possible multiple optimal solutions of λ_j.

Similarly, based upon VRS multiplier models, we have

Theorem 13.7:

1. The CRS efficiency score is equal to the VRS efficiency score *if and only if* CRS prevail on DMU_o. Otherwise,
2. $\mu^* > 0\,(\text{or}\, v^* < 0)$ if and only if IRS prevail on DMU_o.
3. $\mu^* < 0\,(\text{or}\, v^* > 0)$ if and only if DRS prevail on DMU_o.

13.3.3 Spreadsheets for RTS Estimation

We here develop spreadsheet models for RTS estimation based upon Theorem 13.6. The RTS spreadsheet model uses VRS and CRS envelopment spreadsheets. Figure 13.3 shows a spreadsheet for the input-oriented CRS envelopment model where CRS efficiency scores and the optimal $\sum_j^n \lambda_j^*$ are recorded in columns J and

M2 =IF(J2=L2,"CRS",IF(AND(J2<>L2,K2<1),"IRS",IF(AND(J2<>L2,K2>1),"DRS")))

	J	K	L	M	N	O	P
1	CRS Efficiency	Σλ	VRS Efficiency	RTS	Company		
2	0.662831738	1.101942	1	DRS	Mitsubishi		
3	1	1	1	CRS	Mitsui		
4	1	1	1	CRS	Itochu		
5	1	1	1	CRS	General Motors		
6	1	1	1	CRS	Sumitomo		
7	0.971966637	0.956252	1	IRS	Marubeni		
8	0.737166307	0.99751	0.737555958	IRS	Ford Motor		
9	0.524557613	0.819264	0.603245345	IRS	Toyota Motor		
10	1	1	1	CRS	Exxon		
11	0.841423731	1.067172	1	DRS	Royal Dutch/Shell Group		
12	1	1	1	CRS	Wal-Mart		
13	0.386057261	0.62692	0.557595838	IRS	Hitachi		
14	1	1	1	CRS	Nippon Life Insurance		
15	0.348577853	0.619252	0.470610997	IRS	Nippon Telegraph & Telephone		
16	0.270381772	0.481746	0.533543522	IRS	AT&T		

Fig. 13.4 Input-oriented RTS classification

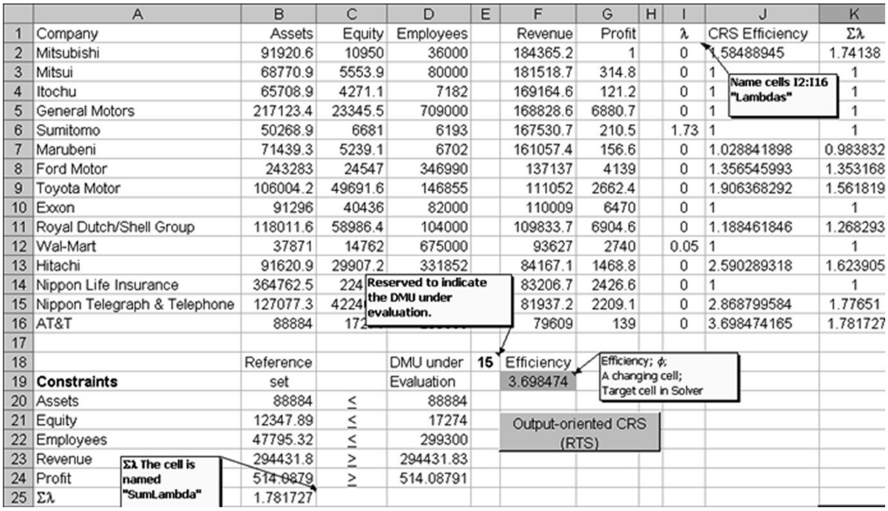
K, respectively. The button "Input-oriented CRS (RTS)" is linked to a VBA procedure "RTS".

```
Sub RTS()
    Dim i As Integer
    For i = 1 To 15
'set the value of cell E18 equal to i (1, 2,..., 15)
    Range("E18") = i
'Run the Solver model. The UserFinish is set to True so that
'the Solver Results dialog box will not be shown
    SolverSolve UserFinish:=True
'Place the efficiency into column J
    Range("J" & i + 1) = Range("F19")
'Place the sum of lambdas into column K
     Range("K" & i + 1) = Range("B25")
    Next i
End Sub
```

In order to obtain the RTS classification, we need also to calculate the input-oriented VRS envelopment model. This can be achieved by using the spreadsheet model shown in Fig. 1.8 (Chap. 1). We then copy the VRS efficiency scores into column L, as shown in Fig. 13.4. Cells M2:M16 contain formulas based upon Theorem 13.6. The formula for cell M2 which is copied into cells M3:M16 is

=IF(J2=L2,"CRS",IF(AND(J2<>L2,K2<1),"IRS",IF(AND(J2<>L2,K2>1),"DRS")))

To obtain the output-oriented RTS classification, we use the spreadsheet for output-oriented CRS envelopment model. Figure 13.5 shows the spreadsheet, and Fig. 13.6



	A	B	C	D	E	F	G	H	I	J	K
1	Company	Assets	Equity	Employees		Revenue	Profit		λ	CRS Efficiency	Σλ
2	Mitsubishi	91920.6	10950	36000		184365.2	1		0	1.58488945	1.74138
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0	1	1
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0	1	1
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0	1	1
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		1.73	1	1
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0	1.028841898	0.983832
8	Ford Motor	243283	24547	346990		137137	4139		0	1.356545993	1.353168
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0	1.906368292	1.561819
10	Exxon	91296	40436	82000		110009	6470		0	1	1
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0	1.188461846	1.268293
12	Wal-Mart	37871	14762	675000		93627	2740		0.05	1	1
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0	2.590289318	1.623905
14	Nippon Life Insurance	364762.5	224			83206.7	2426.6		0	1	1
15	Nippon Telegraph & Telephone	127077.3	4224			81937.2	2209.1		0	2.868799584	1.77651
16	AT&T	88884	172			79609	139		0	3.698474165	1.781727
17											
18		Reference		DMU under	15	Efficiency					
19	Constraints	set		Evaluation		3.698474					
20	Assets	88884	≤	88884							
21	Equity	12347.89	≤	17274							
22	Employees	47795.32	≤	299300							
23	Revenue	294431.8	≥	294431.83							
24	Profit	514.0879	≥	514.08791							
25	Σλ	1.781727									

Fig. 13.5 Output-oriented RTS classification spreadsheet model

shows the Solver parameters. Note that range names are used in the spreadsheet shown in Fig. 13.5 as in the spreadsheet for output-oriented VRS envelopment model shown in Fig. 1.27. For example, cell E18 is named as “DMU”, cell F19 is named as “Efficiency”, and cell B25 is named as “SumLambda”. The button “Output-oriented CRS” is linked to a VBA procedure “GeneralRTS” which automates the calculation, and records the efficiency score and $\sum_j^n \lambda_j^*$ into columns J and K, respectively.

```
Sub GeneralRTS()
Dim NDMUs As Integer, NInputs As Integer, NOutputs As Integer
    NDMUs = 15
    NInputs = 3
    NOutputs = 2
    Dim i As Integer
    For i = 1 To NDMUs
    Range("DMU") = i
    SolverSolve UserFinish:=True
Range("A1").Offset(i,NInputs+NOutputs+4) = Range("Efficiency")
Range("A1").Offset(i, NInputs+NOutputs+5) = Range("SumLambda")
    Next
End Sub
```

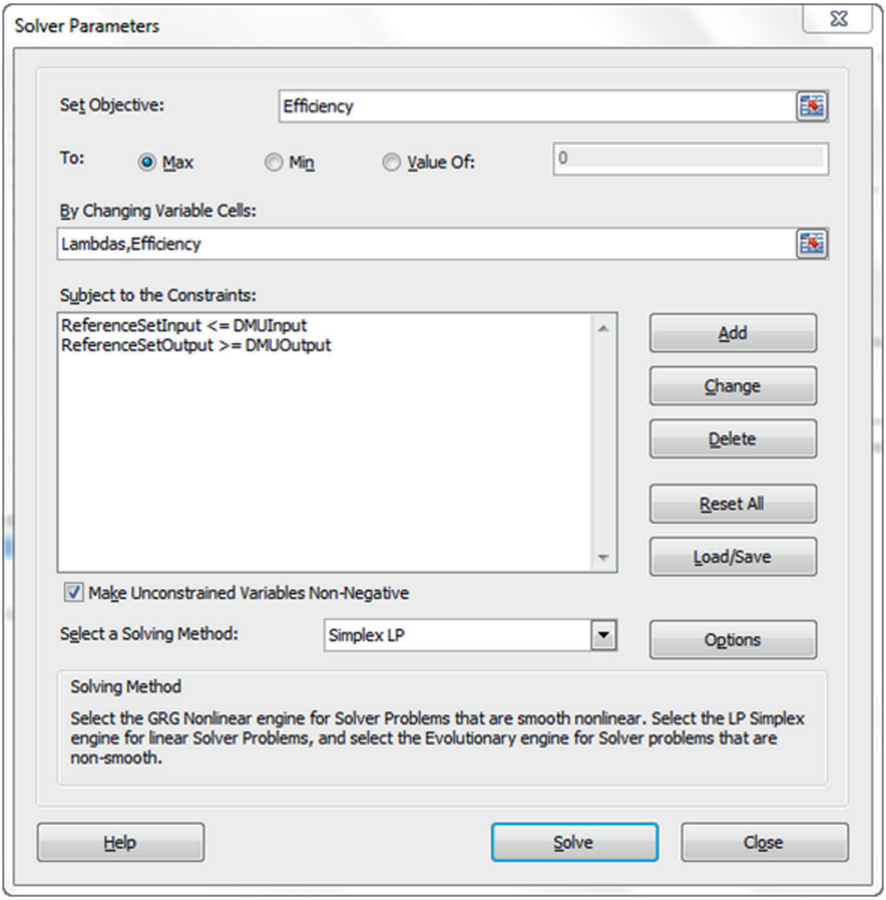


Fig. 13.6 Solver parameters for output-oriented CRS envelopment model

Note that we can assign "RTS" to the button "Output-oriented CRS (RTS)". In fact, when the range names are used, Range("DMU"), Range("Efficiency"), and Range("SumLambda") are equivalent to Range("E18"), Range("F19"), and Range("B25"), respectively. The procedure "GeneralRTS" can be applied to other data sets with the range names.

With the output-oriented VRS efficiency scores and Theorem 13.6, we can obtain the output-oriented RTS classification shown in Fig. 13.7.

Based upon Figs. 13.4 and 13.7, we obtain the RTS regions (see column O in Fig. 13.7).

M2		=IF(J2=L2,"CRS",IF(AND(J2<>L2,K2<1),"IRS",IF(AND(J2<>L2,K2>1),"DRS")))					
	J	K	L	M	N	O	P
1	CRS Efficiency	Σλ	VRS Efficiency	RTS	Company	RTS Region	
2	1.58488945	1.74138	1	DRS	Mitsubishi	Region III	
3	1	1	1	CRS	Mitsui	Region II	
4	1	1	1	CRS	Itochu	Region II	
5	1	1	1	CRS	General Motors	Region II	
6	1	1	1	CRS	Sumitomo	Region II	
7	1.028841898	0.983832	1	IRS	Marubeni	Region I	
8	1.356545993	1.353168	1.158414974	DRS	Ford Motor	Region VI	
9	1.906368292	1.561819	1.371588284	DRS	Toyota Motor	Region VI	
10	1	1	1	CRS	Exxon	Region II	
11	1.188461846	1.268293	1	DRS	Royal Dutch/Shell Group	Region III	
12	1	1	1	CRS	Wal-Mart	Region II	
13	2.590289318	1.623905	1.898938938	DRS	Hitachi	Region VI	
14	1	1	1	CRS	Nippon Life Insurance	Region II	
15	2.868799584	1.77651	1.892916538	DRS	Nippon Telegraph & Telephone	Region VI	
16	3.698474165	1.781727	2.311193684	DRS	AT&T	Region VI	

Fig. 13.7 Output-oriented RTS classification

13.4 Scale Efficient Targets

By using the most productive scale size (MPSS) concept (Banker 1984), we can develop linear programming problems to set unique scale efficient target. Consider the following linear program when the input-oriented CRS envelopment model is solved (Zhu 2000b).

$$
\begin{aligned}
&\min \sum_{j=1}^{n} \lambda_j \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le \theta^* x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} \qquad r = 1,2,\ldots,s; \\
&\lambda_j \ge 0. \qquad\qquad j = 1,2,...,n.
\end{aligned}
\tag{13.1}
$$

where θ^* is the input-oriented CRS efficiency score.

Based upon the optimal values from (13.1) (i.e., $\sum \lambda_j^*$), the MPSS concept yields the following scale-efficient target for DMU_o corresponding to the largest MPSS

$$
\text{MPSS}_{\max} : \begin{cases} \tilde{x}_{io} = \theta^* x_{io} \big/ \sum \lambda_j^* \\ \tilde{y}_{ro} = y_{ro} \big/ \sum \lambda_j^* \end{cases}
\tag{13.2}
$$

where (~) represents the target value.

If we change the objective of (13.1) to maximization,

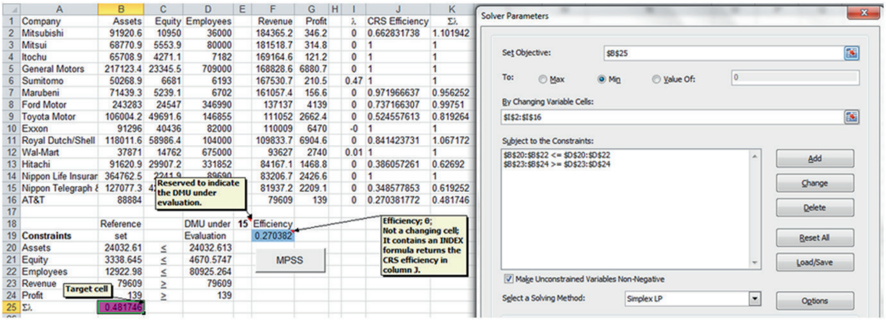


Fig. 13.8 Largest MPSS spreadsheet model

$$
\begin{aligned}
&\min \sum_{j=1}^{n} \hat{\lambda}_j \\
&\text{subject to} \\
&\sum_{j=1}^{n} \hat{\lambda}_j x_{ij} \le \theta^* x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \hat{\lambda}_j y_{rj} \ge y_{ro} \qquad r = 1,2,\ldots,s; \\
&\hat{\lambda}_j \ge 0. \qquad j = 1,2,\ldots,n.
\end{aligned}
\tag{13.3}
$$

then we have the scale efficient target corresponding to the smallest MPSS.

$$
\text{MPSS}_{\min}: \begin{cases} \tilde{x}_{io} = \theta^* x_{io} / \sum \hat{\lambda}_j^* \\ \tilde{y}_{ro} = y_{ro} / \sum \hat{\lambda}_j^* \end{cases}
\tag{13.4}
$$

Note that models (13.1) and (13.3) are based upon the input-oriented CRS envelopment model. However, by using the relationship between the input-oriented and output-oriented CRS envelopment models (see Lemma 13.2), it is trivial to show that $\text{MPSS}_{\max}$ ($\text{MPSS}_{\min}$) remains the same under both orientations. Consequently, $\text{MPSS}_{\max}$ and $\text{MPSS}_{\min}$ are uniquely determined by θ^* and $\sum \lambda_j^* \left(\sum \hat{\lambda}_j^* \right)$.

We can select the largest or the smallest MPSS target for a particular DMU under consideration based upon the RTS preference over performance improvement. For example, one may select the smallest MPSS for an IRS DMU and the largest MPSS for a DRS DMU. Further, if the CRS envelopment models yield the unique optimal solutions, then the $\text{MPSS}_{\max}$ and $\text{MPSS}_{\min}$ are the same.

The spreadsheet model for calculating the scale efficient target involves (i) calculating CRS envelopment model, and (ii) calculating model (13.1). We demonstrate (ii) using the input-oriented CRS envelopment model shown in Fig. 13.3.

In Fig. 13.8, the target cell is B25, and contains the formula "=SUM(I2:I16)", representing the $\sum \lambda_j^*$. Cell F19 is no longer a changing cell, and contains the

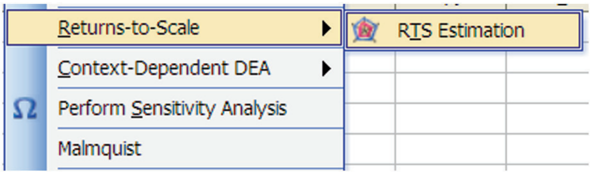


Fig. 13.9 Returns-to-scale menu

formula “=INDEX(J2:J16,E18,1)”. This formula returns the CRS efficiency score of a DMU under evaluation from column J.

The changing cells are I2:I16. The constraints in the Solver parameters for the input-oriented CRS envelopment model shown in Fig. 1.24 remain the same. Figure 13.8 also shows the Solver parameters for calculating the model (13.1). Select “Max” if model (13.3) is used.

To automate the computation, we remove the statement Range(“J”& i+1)=Range(“F19”) from the procedure “RTS”, and name the new procedure “MPSS”.

```
Sub MPSS()
  Dim i As Integer
    For i = 1 To 15
'set the value of cell E18  equal to i (1, 2,..., 15)
    Range("E18") = i
'Run the Solver model. The UserFinish is set to True so that
'the Solver Results dialog box will not be shown
    SolverSolve UserFinish:=True
'Place the sum of lambdas into column K
     Range("K" &  i + 1) = Range("B25")
    Next i
End Sub
```

It can be seen that the maximum $\sum \lambda_j^*$ is the same as that obtained from the input-oriented CRS envelopment model shown in Fig. 13.3. This is due to the fact that we have unique optimal solutions on λ_j^*. As a result, minimum $\sum \hat{\lambda}_j^* =$ maximum $\sum \lambda_j^*$. We can apply (13.2) or (13.4) to obtain the scale efficient targets for the 15 DMUs.

13.5 Solving DEA Using DEAFrontier Software

RTS Estimation can be found at the Returns-to-Scale menu item, as shown in Fig. 13.9.

The RTS Estimation menu will provide (i) the RTS classifications, and (ii) RTS regions as shown in Fig. 13.2 (see Fig. 13.10).

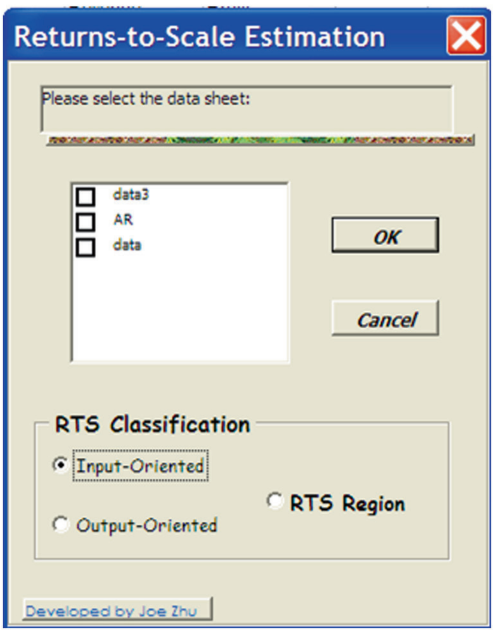


Fig. 13.10 RTS estimation

If RTS Region is selected, the software will run both the input-oriented and output-oriented envelopment models. The results are reported in the "RTS Region" sheet.

If Input-Oriented is selected, then the software will generate the RTS classification based upon the input-oriented envelopment models and report the results in the sheet "RTS Report". If Output-Oriented is selected, then the software will generate the RTS classification based upon the output-oriented envelopment models.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_13) contains supplementary material, which is available to authorized users.

References

Banker, R. D. (1984). Estimating most productive scale size using data envelopment analysis. *European Journal of Operational Research, 17,* 35–44.

Banker, R. D., Charnes, A., Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Science, 30,* 1078–1092.

Färe, R., Grosskopf, S., & Lovell, C. A. K. (1994). *Production frontiers*. Cambridge: Cambridge University Press.

Gregoriou, G. N., & Zhu, J. (2005). *Evaluating hedge fund and CTA performance: Data envelopment analysis approach*. New York: Wiley.

Seiford, L. M., & Zhu, J. (1999a). An investigation of returns to scale under data envelopment analysis, *OMEGA, 27*(1), 1–11.

Seiford, L. M., & Zhu, J. (1999b). Sensitivity and stability of the classification of returns to scale in data envelopment analysis. *Journal of Productivity Analysis, 12*(1), 55–75.

Seiford, L. M., & Zhu, J. (1999c). Profitability and marketability of the top 55 US commercial banks. *Management Science, 45*(9), 1270–1288.

Seiford, L. M., & Zhu, J. (2005). Notes on sensitivity and stability of the classifications of returns to scale in data envelopment analysis: A comment. *Journal of Productivity Analysis, 23*(3), 315–316.

Zhu, J. (2000a). Further discussion on DEA and linear production function. *European Journal of Operational Research, 127*(3), 611–618.

Zhu, J. (2000b). Setting scale efficient targets in DEA via returns to scale estimation method. *Journal of Operational Research Society, 51*(3), 376–378.

Chapter 14
DEA Models for Two-Stage Network Processes

14.1 Introduction

While the definition of a DMU is generic and DMUs can be in various forms such as hospitals, products, universities, cities, courts, business firms, and others, DMUs can have a two-stage structure in many cases. For example, banks use labor and assets to generate deposits which are in turn used to generate load incomes. Seiford and Zhu (1999) use a two-stage process to measure the profitability and marketability of US commercial banks. In their study, profitability is measured using labor and assets as inputs, and the outputs are profits and revenue. In the second stage for marketability, the profits and revenue are then used as inputs, while market value, returns and earnings per share are used as outputs. Chilingerian and Sherman (2004) describe another two-stage process in measuring physician care. Their first stage is a manager-controlled process with inputs including registered nurses, medical supplies, and capital and fixed costs. These inputs generate the outputs or intermediate measures (inputs to the second stage), including patient days, quality of treatment, drug dispensed, among others. The outputs of the second (physician controlled) stage include research grants, quality of patients, and quantity of individuals trained, by specialty.

In these settings, a DMU represents a two-stage process and intermediate measures exist in-between the two stages. The first stage uses inputs to generate outputs which become the inputs to the second stage. The first stage outputs are therefore called intermediate measures. The second stage then uses these intermediate measures to produce outputs. A key feature here is that the first stage's outputs are the only inputs to the second stage. i.e., in addition to the intermediate measures, the first stage does not have its own outputs and the second stage does not have its own inputs.

These two-stage processes are different from the supply chains discussed in Chap. 15 where the second stage also has its own independent inputs. An usual attempt to deal with such two-stage processes is to apply the standard DEA model to each stage (see, e.g., Seiford and Zhu 1999). However, as noted in Chap. 8 and

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_14,

Chen and Zhu (2004), such an approach may conclude that two inefficient stages lead to an overall efficient DMU with the inputs of the first stage and outputs of the second stage. Consequently, improvement to the DEA frontier can be distorted. i.e., the performance improvement of one stage affects the efficiency status of the other, because of the presence of intermediate measures.

Based upon the variable returns to scale (VRS) envelopment model, Chen and Zhu (2004) develop a linear DEA type model where each stage's efficiency is defined on its own production possibility set. The two production possibility sets are linked with the intermediate measures which are set as decision variables for each DMU under evaluation. Chen and Zhu's (2004) model guarantees an overall efficient two-stage process when each stage is efficient. For inefficient DMUs, Chen and Zhu's (2004) model provides a DEA projection with a set of optimal intermediate measures.

Kao and Hwang (2008), on the other hand, modify the standard DEA model by taking into account the series relationship of the two stages within the whole process. Under their framework, the efficiency of the whole process can be decomposed into the product of the efficiencies of the two sub-processes. Yet, their approach cannot be directly applied to the VRS assumption.

Chen et al. (2009) develop an equivalence between Chen and Zhu (2004) and Kao and Hwang (2008) under the condition of constant returns to scale (CRS).

The current chapter presents the models of Chen and Zhu (2004) and Kao and Hwang (2008) and their relations.

14.2 VRS Two-Stage Model

Chen and Zhu (2004) consider the indirect impact of information technology (IT) on firm performance where IT directly impacts certain intermediate measures which in turn are transformed to realize firm performance. Figure 14.1 describes the indirect impact of IT on firm performance where the first stage uses inputs $x_i (i = 1,\ldots,m)$ to produce outputs $z_d (d = 1,\ldots,D)$, and then these z_d are used as inputs in the second stage to produce outputs $y_r (r = 1,\ldots,s)$. It can be seen that z_d (intermediate measures) are outputs in stage 1 and inputs in stage 2. The first stage is viewed as an IT-related value-added activity where deposit is generated and then used as the input to the second stage where revenue is generated.

Based upon the VRS envelopment model, Chen and Zhu (2004) develop the following model

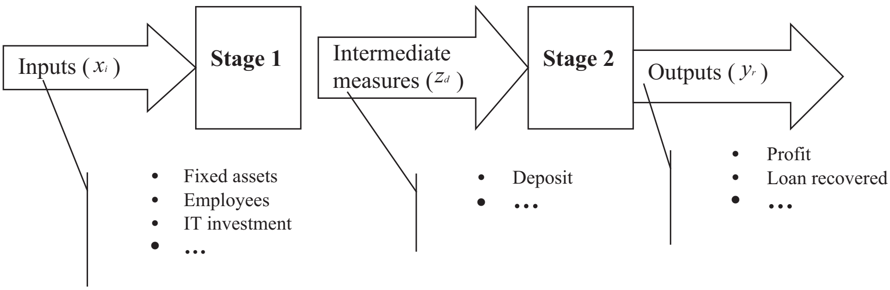


Fig. 14.1 IT Impact on firm performance

$$
\begin{aligned}
&\min_{\alpha,\beta,\lambda_j,\mu_j,\tilde{z}} w_1\alpha - w_2\beta \\
&\textit{subject to} \\
&\text{(stage 1)} \\
&\sum_{j=1}^{n}\lambda_j x_{ij} \le \alpha x_{ij_o} \qquad i = 1,\ldots,\mathrm{m} \\
&\sum_{j=1}^{n}\lambda_j z_{dj} \ge \tilde{z}_{dj_o} \qquad d = 1,\ldots,\mathrm{D} \\
&\sum_{j=1}^{n}\lambda_j = 1 \\
&\lambda_j \ge 0, \quad j = 1,\ldots,n \\
&\alpha \le 1 \\
&\text{(Stage 2)} \\
&\sum_{j=1}^{n}\mu_j z_{dj} \le \tilde{z}_{dj_o} \qquad d = 1,\ldots,\mathrm{D} \\
&\sum_{j=1}^{n}\mu_j y_{rj} \ge \beta y_{rj_o} \qquad r = 1,\ldots,\mathrm{s} \\
&\sum_{j=1}^{n}\mu_j = 1 \\
&\mu_j \ge 0, \quad j = 1,\ldots,n \\
&\beta \ge 1
\end{aligned}
\tag{14.1}
$$

where w_1 and w_2 are user-specified weights reflecting the preference over the two stages' performance, and symbol "~" represents unknown decision variables.

The rationale of model (14.1) is as follows: (i) when we evaluate the impact of IT investment on the intermediate measures, we want to minimize the input usage given the intermediate measures. For example, given the deposits generated, our

	A	B	C	D	E	F	G	H	I	J	K	L
1	Supply Chain	Shipping costs	Labor		Revenue		Profit	λ	μ	α	β	Optimal Revenue
2	A	7	9		4		16	0.14925	0	1	1.4375	4
3	B	9	4		6		14	0.85075	0	1	1.6429	6
4	C	11	6		3		23	0	1	0.79104	1	3
5												
6					3							
7				3	α	β						
8	Constraints				0.79104	1	-0.20896					
9	Shipping costs	8.701492537	≤	8.70149								
10	Labor	4.746268657	≤	4.74627								
11	Revenue	5.701492537	≥	3								
12	Revenue	3	≤	3		Run						
13	Profit	23	≥	23								
14	Σλ	1	=	1								
15	Σμ	1	=	1								

Fig. 14.2 Spreadsheet model for model (14.1)

objective is to examine whether a bank can reduce its input consumption (including IT investment) compared to the best practice, and (ii) when we evaluate the firm performance as a result of the intermediate measures, we want to maximize the performance given the intermediate measures. For example, given the deposits it generated, our objective is to examine whether a bank can increase its profit. Model (14.1) characterizes the indirect impact of IT on firm performance in a single linear programming problem.

If $\alpha^* = \beta^* = 1$, the two-stage achieves efficient performance when the two-stage process is viewed as a whole.

If $\alpha^* = 1$ and $\beta^* > 1\,(or\, \alpha^* < 1\, and\ \beta^* = 1)$, then model (14.1) indicates that one of the stages can achieve 100 % efficiency given a set of optimal intermediate measures.

A DMU must be a frontier point in both stages with respect to $\alpha^* x_{ij_o}$ $(i = 1, \ldots, m)$, $\tilde{z}^*_{dj_o}$ $(d = 1, \ldots, D)$, and $\beta^* y_{rj_o}$ $(r = 1, \ldots, s)$, where (*) represents optimal value in model (14.1).

In model (14.1), the intermediate measures for a specific DMU_o under evaluation are set as unknown decision variables, $\tilde{z}_{dj_o}$. As a result, additional constraints can be imposed on the intermediate measures. This can further help in correctly characterizing the indirect impact of IT on firm performance.

To illustrate model (14.1), Fig. 14.2 shows the spreadsheet model of (14.1) with the data in Table 8.1 (Chap. 8). Since the intermediate measures are set as decision variables, cell E6 is reserved to represent the Revenue variable. Cell D7 indicates the DMU under evaluation. Cells E8 and F8 represent α and β, respectively. Cell G8 is the objective function and contains the formula "=E8-F8".

The changing cells are cells H2:H4, cells I2:I4, cells E8:F8,,and cell E6. The formulas for cells B9:B15 are

Cell B9 =SUMPRODUCT(B2:B4,H2:H4)
Cell B10 =SUMPRODUCT(C2:C4,H2:H4)
Cell B11 =SUMPRODUCT(E2:E4,H2:H4)
Cell B12 =SUMPRODUCT(E2:E4,I2:I4)
Cell B13 =SUMPRODUCT(G2:G4,I2:I4)

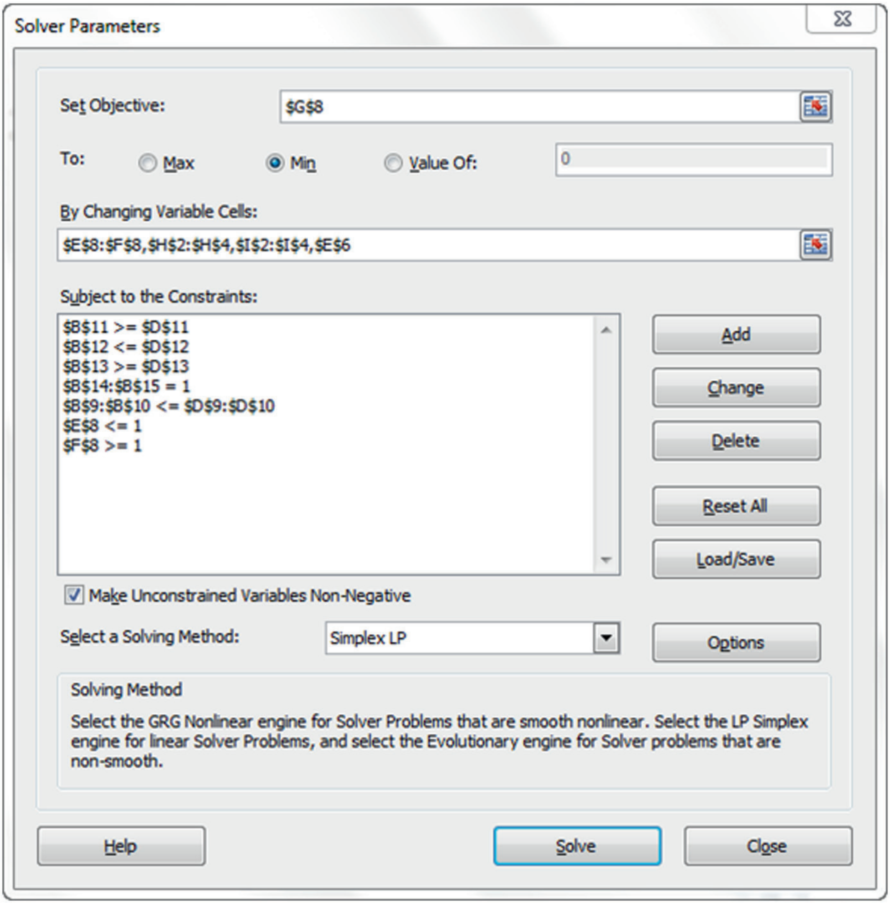


Fig. 14.3 Solver parameters for model (14.1)

Cell B14 =SUM(H2:H4)
Cell B15 =SUM(I2:I4)

The formulas for cells D9:D13 are

Cell D9 =E8*INDEX(B2:B4,D7,1)
Cell D10 =E8*INDEX(C2:C4,D7,1)
Cell D11 =E6
Cell D12 =E6
Cell D13 =F8*INDEX(G2:G4,D7,1)

Figure 14.3 shows the Solver parameters for the spreadsheet model shown in Fig. 14.2. We have "$E$8 <=1" and "$F$8 >=1" in the Constraints, representing $\alpha \leq 1$ and $\beta \geq 1$, respectively.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	DMU	Fixed assets	IT budget	# of employees		Deposits			Profit	load recovered	λ	μ	α	β
2	1	0.713	0.15	13.3		14.478			0.232	0.986	0.2322	0	1	1
3	2	1.071	0.17	16.9		19.502			0.34	0.986	0	0	0.94516	1
4	3	1.224	0.235	24		20.952			0.363	0.986	0	0.71632	0.71639	1
5	4	0.363	0.211	15.6		13.902			0.211	0.982	0	0	1	1.00388
13	12	0.874	0.05	13.1		14.342			0.177	0.985	0	0	1	1.00097
14	13	6.918	0.37	12.5		32.491			0.648	0.945	0.3898	0.03809	1	1
15	14	4.432	0.44	41.9		47.653			0.639	0.979	0	0	0.60513	1
16	15	4.504	0.431	41.1		52.63			0.741	0.981	0	0	0.71259	1
17														
18						46.7158								
19					15	α	β							
20	Constraints					0.71259	1		-0.2874					
21	Fixed assets	3.2095005	≤	3.2095005										
22	IT budget	0.20174686	≤	0.3071258										
23	# of employees	29.2874047	≤	29.287405			IT							
24		0												
25	Deposits	46.7158495	≥	46.71585										
26	Deposits	46.7158495	≤	46.71585										
27														
28	Profit	0.741	≥	0.741										
29	load recovered	0.981	≥	0.981										
30	Σλ	1	=	1										
31	Σμ	1	=	1										

Fig. 14.4 IT Spreadsheet model

In Fig. 14.2, cells J2:K4 record the efficiency scores. Cells L2:L4 report the optimal values on Revenue. Since optimal Revenue values are equal to the original values, α^* and β^*must be equal to the θ^* and ϕ^* of the VRS envelopment models, respectively. In this case, multiple optimal solutions on Revenue exist. (Click the "Run" button several time, you may get a set of different optimal values on Revenue.)

We next apply model (14.1) to 15 DMUs in a data set used by Wang, Gopal and Zionts (1997) which consists observations on 22 firms in the banking industry in the years 1987–1989[1]. The data can be found in the file "two stage spreadsheet.xls". The inputs, intermediate measure and outputs are given in the Fig. 14.1.

Figure 14.4 shows the spreadsheet. Range names are used in the spreadsheet shown in Fig. 14.4. Cells B2:D16 are named as "Stage1Inputs". Cells F2:F16 are named as "Intermediate". Cells I2:J16 are named as "Stage2Outputs". These cells represent the performance measures for the 15 banks.

Cells K2:K16 are named as "Lambdas" and cells L2:L16 are named as "Mus". These cells are changing cells in the Solver parameters. Other changing cells include cell F18—"Deposits", representing the decision variables for the intermediate measures, cell F20—"Efficiency1", and cell G20—"Efficiency2", representing α and β in model (14.1).

Cell I20 is the objective function of model (14.1). It contains the formula "=Efficiency1-Efficiency2", i.e., "=F20-G20". Cell I20 is named as "Efficiency".

Based upon these range names and the related cells, we have the formulas for the constraints

[1] There are 36 observations in Wang et al. (1997). The data on IT budgets are obtained from the annual survey by Computer World on top 100 effective users of information systems. The data on the percentage of loans recovered and the dollar value of deposits are generated from Standard and Poor's Industry Surveys. The remaining data are obtained from the Compustat database.

Cell B21 =SUMPRODUCT(Lambdas,INDEX(Stage1Inputs,0,1))
Cell B22 =SUMPRODUCT(Lambdas,INDEX(Stage1Inputs,0,2))
Cell B23 =SUMPRODUCT(Lambdas,INDEX(Stage1Inputs,0,3))

Cell B25 =SUMPRODUCT(Lambdas,INDEX(Intermediate,0,1))
Cell B26 =SUMPRODUCT(Mus,INDEX(Intermediate,0,1))

Cell B28 =SUMPRODUCT(Mus,INDEX(Stage2Outputs,0,1))
Cell B29 =SUMPRODUCT(Mus,INDEX(Stage2Outputs,0,2))
Cell B30 =SUM(Lambdas)
Cell B31 =SUM(Mus)

Cell D21 =Efficiency1*INDEX(Stage1Inputs,E19,1)
Cell D22 =Efficiency1*INDEX(Stage1Inputs,E19,2)
Cell D23 =Efficiency1*INDEX(Stage1Inputs,E19,3)

Cell D25 =Deposits
Cell D26 =Deposits

Cell D28 =Efficiency2*INDEX(Stage2Outputs,E19,1)
Cell D29 =Efficiency2*INDEX(Stage2Outputs,E19,2)

We then apply the following range names to the constraints

Cells B21:B23 – ReferenceSetInput
Cells B25 – ReferenceSetInter1
Cells B26 – ReferenceSetInter2
Cells B28:B29 – ReferenceSetOutput
Cell B30 – SumLambda
Cell B31 – SumMu
Cells D21:D23 – DMUInput
Cells D25 – DMUInter1
Cells D26 – DMUInter2
Cells D28:D29 – DMUOutput

Figure 14.5 shows the Solver Parameters. The calculation is performed by the following VBA procedure

```
Sub IT()
Dim i As Integer
For i = 1 To 15
Range("E19") = i
SolverSolve UserFinish:=True
'Place the efficiency into column J and column K
Range("M" & i + 1) = Range("Efficiency1")
Range("N" & i + 1) = Range("Efficiency2")
Next i
End Sub
```

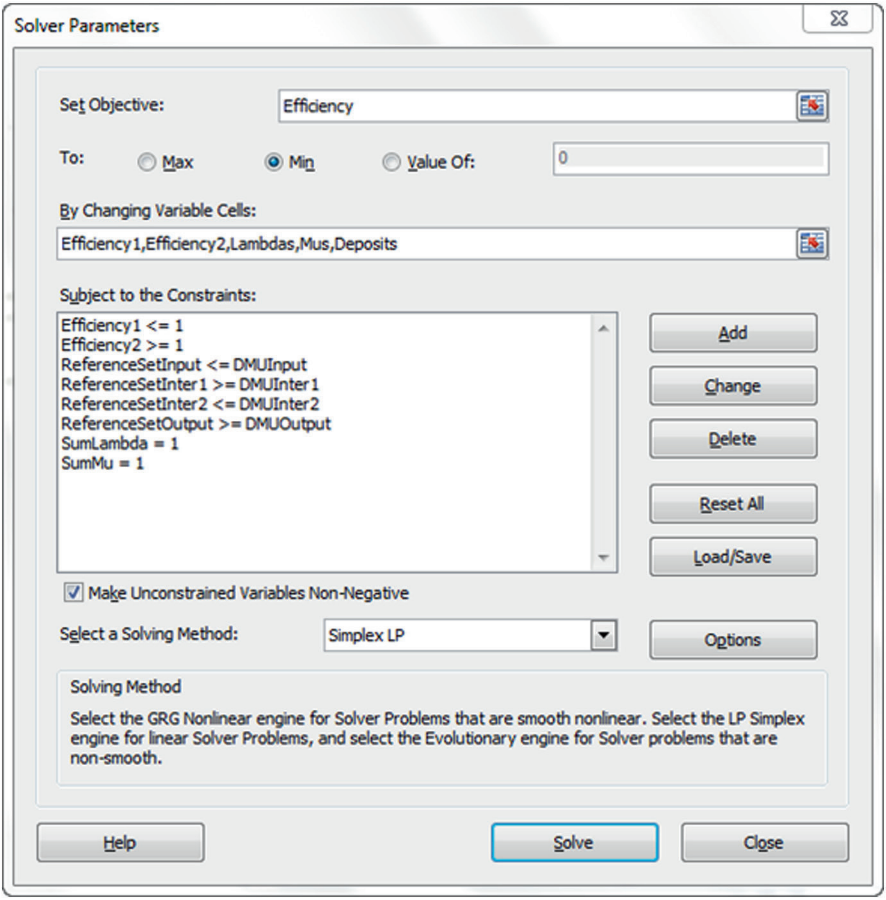


Fig. 14.5 Solver parameters for IT spreadsheet

14.3 CRS Two-Stage Model: Centralized Model

Based upon the CRS DEA model, the efficiency scores of the two-stage process and the two individual stages can be expressed as

$$\theta_j = \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}}, \ \theta_j^1 = \frac{\sum_{d=1}^{D} w_d z_{dj}}{\sum_{i=1}^{m} v_i x_{ij}} \ and \ \theta_j^2 = \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{d=1}^{D} \tilde{w}_d z_{dj}} \tag{14.2}$$

where v_i, w_d, $\tilde{w}_d$, and u_r are unknown non-negative weights. Note that w_d can be equal to $\tilde{w}_d$.

Note that the intermediate measures of z_{dj} do not appear in θ_j. Kao and Hwang (2008) and Liang et al. (2008) assume that $w_d = \tilde{w}_d$. As a result, for a specific $DMU_{j_0}, \theta^1_{j_0} \bullet \theta^2_{j_0}$ becomes $\dfrac{\sum_{r=1}^{s} u_r y_{rj_0}}{\sum_{i=1}^{m} v_i x_{ij_0}}$ which is the overall efficiency defined in the

Kao and Hwang (2008) model:

$$
\begin{aligned}
& Max\ \theta^1_o \bullet \theta^2_o = \frac{\sum_{r=1}^{s} u_r y_{rj_0}}{\sum_{i=1}^{m} v_i x_{ij_0}} \\
& s.t.\ \ \theta^1_j \le 1\ and\ \theta^2_j \le 1\ for\ all\ j \\
& w_d = \tilde{w}_d \text{ for all } d
\end{aligned}
\tag{14.3}
$$

Model (14.3) indicates an efficiency decomposition. That is, efficiency of the whole process can be decomposed into the product of the efficiencies of the two sub-processes

Note that model (14.3) is an input-oriented model. It can be converted into the following linear program

$$
\begin{aligned}
& e_o^{centralized} = Max \sum_{r=1}^{s} u_r y_{ro} \\
& s.t. \sum_{r=1}^{s} u_r y_{rj} - \sum_{d=1}^{D} w_d z_{dj} \le 0\ \ j = 1, 2, \ldots, n \\
& \sum_{d=1}^{D} w_d z_{dj} - \sum_{i=1}^{m} v_i x_{ij} \le 0\ \ j = 1, 2, \ldots, n \\
& \sum_{i=1}^{m} v_i x_{io} = 1 \\
& w_d \ge 0, d = 1, 2, \ldots, D; v_i \ge, i = 1, 2, \ldots, m; u_r \ge, r = 1, 2, \ldots, s
\end{aligned}
\tag{14.4}
$$

In Liang et al. (2008), the above model is called centralized model. Model (14.4) gives the overall efficiency of the two-stage process. Assume the above model (14.4) yields a unique solution. We then obtain the efficiencies for the first and second stages, namely

$$e_o^{1,Centralized} = \frac{\sum_{d=1}^{D} w_d^* z_{do}}{\sum_{i=1}^{m} v_i^* x_{io}} = \sum_{d=1}^{D} w_d^* z_{do} \ and \ e_o^{2,Centralized} = \frac{\sum_{r=1}^{s} u_r^* y_{ro}}{\sum_{d=1}^{D} w_d^* z_{do}} \tag{14.5}$$

If we denote the optimal value to model (14.4) as $e_o^{centralized}$, then we have $e_o^{centralized} = e_o^{1,Centralized} \bullet e_o^{2,Centralized}$. Note that optimal multipliers from model (14.4) may not be unique, meaning that $e_o^{1,Centralized}$ and $e_o^{2,Centralized}$ may not be unique. Liang et al. (2008) develop a procedure for testing for the uniqueness. They first determine the maximum achievable value of $e_o^{1,Centralized}$ via

$$\begin{aligned}
& e_o^{1+} = Max \sum_{d=1}^{D} w_d z_{do} \\
& s.t. \sum_{r=1}^{s} u_r y_{ro} = e_o^{centralized} \\
& \sum_{d=1}^{D} w_d z_{dj} - \sum_{i=1}^{m} v_i x_{ij} \le 0 \ j = 1,2,....,n \\
& \sum_{r=1}^{s} u_r y_{rj} - \sum_{d=1}^{D} w_d z_{dj} \le 0 \ j = 1,2,....,n \\
& \sum_{i=1}^{m} v_i x_{io} = 1 \\
& w_d \ge 0, d = 1,2,...,D; v_i \ge 0, i = 1,2,...,m; u_r \ge 0, r = 1,2,...,s
\end{aligned} \tag{14.6}$$

It then follows that the minimum of $e_o^{2,Centralized}$ is given by $e_o^{2-} = \frac{e_o^{centralized}}{e_o^{1+}}$.

The maximum of $e_o^{2,Centralized}$, which we denote by e_o^{2+}, can be calculated in a manner similar to the above, and the minimum of $e_o^{1,Centralized}$ is then calculated as $e_k^{1-} = e_o^{centralized} / e_o^{2+}$. Note that $e_o^{1-} = e_o^{1+}$ if and only of $e_o^{2-} = e_o^{2+}$. Note also that if $e_o^{1-} = e_o^{1+}$ or $e_o^{2-} = e_o^{2+}$, then $e_o^{1,Centralized}$ and $e_o^{2,Centralized}$ are uniquely determined via model (14.4). If $e_o^{1-} \ne e_o^{1+}$ or $e_o^{2-} \ne e_o^{2+}$, then presumably some flexibility exists in setting values for $e_o^{1,Centralized}$ and $e_o^{2,Centralized}$. A legitimate reason for taking advantage of such flexibility is one of cooperation and fairness. That is, in the spirit of cooperative games, once the optimal value for the centralized score is determined, it is reasonable to search for a decomposition that is as fair as possible to both parties.

The equivalent output-oriented model can be expressed as

$$\begin{aligned}
&Min\frac{\sum_{i=1}^{m} v_i x_{ij_0}}{\sum_{r=1}^{s} u_r y_{rj_0}} \\
&s.t.\ \theta_j^1 \le 1\ and\ \theta_j^2 \le 1\ for\ all\ j \\
&w_d = \tilde{w}_d \text{ for all } d
\end{aligned} \tag{14.7}$$

Model (14.7) is equivalent to the following linear program

$$\begin{aligned}
&Min\sum_{i=1}^{m} v_i x_{ij_0} \\
&s.t.\sum_{r=1}^{s} u_r y_{rj} - \sum_{d=1}^{D} w_d z_{dj} \le 0\ j = 1,2,\ldots,n \\
&\sum_{d=1}^{D} w_d z_{dj} - \sum_{i=1}^{m} v_i x_{ij} \le 0\ j = 1,2,\ldots,n \\
&\sum_{r=1}^{s} u_r y_{rj_0} = 1 \\
&w_d, d = 1,2,\ldots,D;\ v_i, i = 1,2,\ldots m;\ u_r, r = 1,2,\ldots,s \ge 0
\end{aligned} \tag{14.8}$$

The above model is an output-oriented version of Kao and Hwang's (2008) and Liang et al.'s (2008) model.

14.4 CRS Two-Stage Model: Equivalence

We next present the Chen and Zhu (2004) model under the CRS assumption by removing the convexity constraints of $\sum \lambda_j = \sum \mu_j = 1$ in (14.1). We have

$$
\begin{aligned}
&\min_{\alpha,\beta,\lambda_j,\mu_j,\tilde{z}} \alpha-\beta \\
&\textit{subject to} \\
&\text{(stage 1)} \\
&\sum_{j=1}^{n}\lambda_j x_{ij} \le \alpha x_{ij_o} \qquad i=1,\ldots,\mathrm{m} \\
&\sum_{j=1}^{n}\lambda_j z_{dj} \ge \tilde{z}_{dj_o} \qquad d=1,\ldots,\mathrm{D} \\
&\lambda_j \ge 0, \qquad j=1,\ldots,n \\
&\alpha \le 1 \\
&\text{(Stage 2)} \\
&\sum_{j=1}^{n}\mu_j z_{dj} \le \tilde{z}_{dj_o} \qquad d=1,\ldots,\mathrm{D} \\
&\sum_{j=1}^{n}\mu_j y_{rj} \ge \beta y_{rj_o} \qquad r=1,\ldots,\mathrm{s} \\
&\mu_j \ge 0, \qquad j=1,\ldots,n \\
&\beta \ge 1
\end{aligned}
\tag{14.9}
$$

Chen et al. (2009) show (i) that model (14.9) is equivalent to the model (14.8), and (ii) at optimality $\alpha^*=1$ and β^* is the optimal value to model (14.8), representing the overall efficiency.

To establish the equivalence between models (14.8) and (14.9), Chen et al. (2009) first consider the following linear program

$$
\begin{aligned}
&\min_{\alpha,\beta,\lambda_j,\mu_j} \alpha-\beta \\
&\textit{subject to} \\
&\sum_{j=1}^{n}\lambda_j x_{ij} \le \alpha x_{ij_o} \qquad i=1,\ldots,\mathrm{m} \\
&\sum_{j=1}^{n}\mu_j y_{rj} \ge \beta y_{rj_o} \qquad \mathrm{r}=1,2,\ldots,\mathrm{s} \\
&\sum_{j=1}^{n}(\lambda_j-\mu_j) z_{dj} \ge 0 \qquad d=1,2,\ldots,D \\
&\lambda_j,\mu_j \ge 0, \qquad j=1,2,\ldots,n \\
&\alpha \le 1, \beta \ge 1
\end{aligned}
\tag{14.10}
$$

Chen et al. (2009) then show that model (14.10)'s optimal solutions are optimal in model (14.9).

Let $\lambda_j^{'} = \frac{\lambda_j}{\alpha}$, $\mu_j^{'} = \frac{\mu_j}{\alpha}$, and $\sigma = \beta/\alpha$, then model (14.10) becomes

$$
\begin{aligned}
& Max \quad \alpha(\sigma - 1) \\
& s.t. \sum \lambda_j^{'} x_{ij} \le x_{ij_0} \\
& \sum \mu_j^{'} y_{rj} \ge \sigma y_{rj_0} \\
& \sum (\lambda_j^{'} - \mu_j^{'}) z_{dj} \ge 0 \\
& \lambda_j^{'}, \mu_j^{'} \ge 0, \ \alpha \le 1, \alpha\sigma \ge 1
\end{aligned}
\tag{14.11}
$$

Next, consider the following model

$$
\begin{aligned}
& Max \quad \alpha(\sigma - 1) \\
& s.t. \sum \lambda_j^{'} x_{ij} \le x_{ij_0} \\
& \sum \mu_j^{'} y_{rj} \ge \sigma y_{rj_0} \\
& \sum (\lambda_j^{'} - \mu_j^{'}) z_{dj} \ge 0 \\
& \lambda_j^{'}, \mu_j^{'} \ge 0, \ \alpha \le 1, \sigma \ge 1
\end{aligned}
\tag{14.12}
$$

The only difference between models (14.11) and (14.12) is that model (14.12) sets $\alpha = 1$ in the constraint of $\alpha\sigma \ge 1$ in model (14.11). We have model (14.12) optimal solutions are optimal in model (14.11) (see Chen et al. 2009.)

Note that $\alpha \le 1$ does not appear in other constraints of model (14.12). Therefore, at optimality, $\alpha^* = 1$ and model (14.12) is equivalent to the following linear program

$$
\begin{aligned}
& Max \ \sigma \\
& s.t. \sum \lambda_j^{'} x_{ij} \le x_{ij_0} && i = 1, 2, \ldots, m \\
& \sum \mu_j^{'} y_{rj} \ge \sigma y_{rj_0} && r = 1, 2, \ldots, m \\
& \sum (\lambda_j^{'} - \mu_j^{'}) z_{dj} \ge 0 && d = 1, 2, \ldots, D \\
& \lambda_j^{'}, \mu_j^{'} \ge 0, \ \sigma \ge 1
\end{aligned}
\tag{14.13}
$$

Model (14.13) is actually the dual to the model (14.8). Therefore, we have that at optimality $\alpha^* = 1$ and $\beta^* = \sigma^*$, where α^* and β^* are optimal values of α and β in model (14.9) and σ^* is the optimal value of σ in model (14.13).

This further indicates that the optimal α^* and β^* in model (14.9) do not represent the efficiency scores of individual stages under the CRS condition. In fact, α^* is always equal to unity and β^* represents the overall efficiency of the two-stage process. i.e., model (14.7) can be used to measure the overall efficiency of the two-stage process under the CRS condition.

We finally note that under the VRS condition $\theta_{j_0}^1 \bullet \theta_{j_0}^2$ no longer equals to $\frac{\sum_{r=1}^{s} u_r y_{rj_0}}{\sum_{i=1}^{m} v_i x_{ij_0}}$, because of the free variable in the related DEA model. As a result, the VRS version of Kao and Hwang's and Liang et al.'s (2008) model cannot be modelled as in model (14.3). The proven equivalence between the two approaches sheds lights on possible ways to developing the VRS version of the centralized model.

14.5 Frontier Projection

As indicated in Chen et al. (2010), the centralized model does not provide information on DEA projection. Chen et al. (2010) develop models for determining the DEA projection for inefficient DMUs.

The input-oriented projection model is

$$
\begin{aligned}
&\min \tilde{\theta} \\
&s.t. \quad \sum_{j=1}^{n} \lambda_j x_{ij} \leq \tilde{\theta} x_{io} && i = 1,\ldots,\mathrm{m} \\
&\qquad \sum_{j=1}^{n} \mu_j y_{rj} \geq y_{ro} && r = 1,\ldots,\mathrm{s} \\
&\qquad \sum_{j=1}^{n} \lambda_j z_{dj} \geq \tilde{z}_{do} && d = 1,\ldots,\mathrm{D} \\
&\qquad \sum_{j=1}^{n} \mu_j z_{dj} \leq \tilde{z}_{do} && d = 1,\ldots,\mathrm{D} \\
&\qquad \tilde{z}_{do} \geq 0, && d = 1,\ldots,\mathrm{D} \\
&\qquad \ddot{e}_j \geq 0, && j = 1,\ldots,n \\
&\qquad \mu_j \geq 0, && j = 1,\ldots,n \\
&\qquad \tilde{\theta} \leq 1
\end{aligned}
\tag{14.14}
$$

The output-oriented projection model is

$$
\begin{aligned}
& \max \tilde{\phi} \\
& s.t. \quad \sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io} \qquad i = 1,\dots,\mathrm{m} \\
& \qquad \sum_{j=1}^{n} \mu_j y_{rj} \ge \tilde{\phi} y_{ro} \qquad r = 1,\dots,\mathrm{s} \\
& \qquad \sum_{j=1}^{n} \lambda_j z_{dj} \ge \tilde{z}_{do} \qquad d = 1,\dots,\mathrm{D} \\
& \qquad \sum_{j=1}^{n} \mu_j z_{dj} \le \tilde{z}_{do} \qquad d = 1,\dots,\mathrm{D} \\
& \qquad \tilde{z}_{do} \ge 0, \qquad d = 1,\dots,\mathrm{D} \\
& \qquad \lambda_j \ge 0, \qquad j = 1,\dots,n \\
& \qquad \mu_j \ge 0, \qquad j = 1,\dots,n \\
& \qquad \phi \ge 1
\end{aligned}
\tag{14.15}
$$

The key to obtain the correct projections lie in the fact that the intermediate measures need to be optimized. The centralized models do not provide an optimized intermediate measures.

14.6 CRS Two-Stage Model: Leader-Follower Model

This approach is described in Liang et al. (2008). Under the input-orientation, when the first stage is selected as the leader, the first stage's DEA efficiency is determined first in the following CRS model

$$
\begin{aligned}
& e_o^{1*} = \mathrm{Max} \sum_{d=1}^{D} w_d z_{do} \\
& s.t. \sum_{d=1}^{D} w_d z_{dj} - \sum_{i=1}^{m} v_i x_{ij} \le 0 \qquad j = 1, 2, \dots, n \\
& \sum_{i=1}^{m} v_i x_{io} = 1 \\
& w_d \ge 0, d = 1, 2, \dots, D; v_i \ge 0, i = 1, 2, \dots, m.
\end{aligned}
$$

Once we obtain the efficiency for the first stage, the second stage will only consider those variables w_d that maintain $e_o^1 = e_o^{1*}$. The efficiency model for the second stage (now, the follower) is

$$
\begin{aligned}
e_o^{2*} &= Max\left(\sum_{r=1}^{s} u_r y_{ro}\right) / e_o^{1*} \\
s.t.\ & \sum_{r=1}^{s} u_r y_{rj} - \sum_{d=1}^{D} w_d z_{dj} \le 0 \quad j = 1,2,....,n \\
& \sum_{d=1}^{D} w_d z_{dj} - \sum_{i=1}^{m} v_i x_{ij} \le 0 \quad j = 1,2,....,n \\
& \sum_{i=1}^{m} v_i x_{io} = 1 \\
& \sum_{d=1}^{D} w_d z_{do} = e_o^{1*} \\
& w_d \ge 0, d = 1,2,...,D; v_i \ge 0, i = 1,2,...,m; u_r \ge 0, r = 1,2,...,s
\end{aligned}
$$

In a similar manner, if we assume the second stage as the leader, we then calculate the regular DEA efficiency $(e_o^{2^o})$ for the second stage first using the CRS model

$$
\begin{aligned}
e_o^{2^o} &= Max\sum_{r=1}^{s} u_r y_{ro} \\
s.t.\ & \sum_{r=1}^{s} u_r y_{rj} - \sum_{d=1}^{D} w_d z_{dj} \le 0 \ j = 1,2,\ldots,n \\
& \sum_{d=1}^{D} w_d z_{do} = 1 \\
& w_d \ge 0, = 1,2,\ldots,D;\ u_r \ge 0,\ r = 1,2,\ldots,s
\end{aligned}
$$

The efficiency for the first stage (the follower) is then calculated via

$$
\begin{aligned}
\frac{1}{e_o^{1^o}} &= \mathrm{Min} \sum_{i=1}^{m} v_i x_{io} \\
s.t.\ & \sum_{d=1}^{D} w_d z_{dj} - \sum_{i=1}^{m} v_i x_{ij} \le 0 \quad j = 1,2,....,n \\
& \sum_{r=1}^{s} u_r y_{rj} - \sum_{d=1}^{D} w_d z_{dj} \le 0 \quad j = 1,2,....,n \\
& \sum_{d=1}^{D} w_d z_{do} = 1 \\
& \sum_{r=1}^{s} u_r y_{ro} = e_o^{2^o} \\
& w_d \ge 0, d = 1,2,...,D; v_i \ge 0, i = 1,2,...,m; u_r \ge 0, r = 1,2,...,s
\end{aligned}
$$

The output-oriented results are the reversal of their corresponding input-oriented results.

	A	B	C	D	E	F	G	H	I
1	DMU	Fixed asse	IT budget	# of employees		Deposits		Profit	load recovered
2	1	0.713	0.15	13.3		14.478		0.232	0.986
3	2	1.071	0.17	16.9		19.502		0.34	0.986
4	3	1.224	0.235	24		20.952		0.363	0.986
5	4	0.363	0.211	15.6		13.902		0.211	0.982
6	5	0.409	0.133	18.485		15.206		0.237	0.984
7	6	5.846	0.497	56.42		81.186		1.103	0.955
8	7	0.918	0.06	56.42		81.186		1.103	0.986
9	8	1.235	0.071	12		11.441		0.199	0.985
10	9	18.12	1.5	89.51		124.072		1.858	0.972
11	10	1.821	0.12	19.8		17.425		0.274	0.983

Fig. 14.6 Data sheet format for two-stage

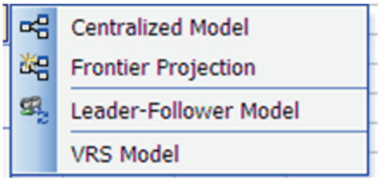


Fig. 14.7 Two-stage network process

14.7 Solving Two-Stage Network Process Using DEAFrontier

Since intermediate measures are present, the DMUs in the data sheet are set in a format shown in Fig. 14.6. The inputs are entered first and followed by a blank column, and then the intermediate measures are entered followed by a blank column and the outputs.

There are four approaches under this two-stage network DEA model, as shown in Fig. 14.7.

Select the "VRS Model" menu item to calculate the model (14.1). The results are reported in the "Efficiency1" (for stage 1), "Efficiency2" (for stage 2) and "Intermediate" (for optimal intermediate measures) sheets.

By selecting the "Centralized Model" option (namely, models 14.3 and 14.4), the user is required to select a sheet that stores the two-stage process data, and the model orientation (see Fig. 14.8). Note that the centralized model is developed under the assumption of CRS. Therefore, the frontier type is assumed to be CRS.

The results are reported in sheet "centralized report" where the overall efficiency scores (under heading "centralized efficiency") are reported along with the efficiency decomposition. This sheet also reports a set of optimal multipliers for the above model.

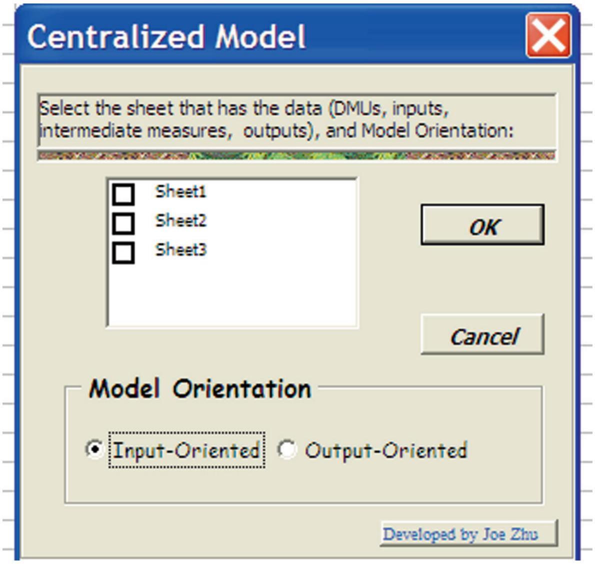


Fig. 14.8 Two-stage network process: centralized model

The results for frontier projection are reported in sheets "Overall_Efficiency" and "OptimalIntermediate". The "OptimalIntemediate" sheet reports a set of optimal intermediate measures. The "Overall_Efficiency" sheets reports the centralized efficiency scores.

If input-orientation is selected, then the frontier point consists of overall efficiency score * input levels, optimal intermediate measures, and current output levels (Fig. 14.9).

The results for leader-follower model are reported in sheet "LF Report". A set of optimal multipliers from the follower's model is also reported. If the user wants to get detailed information on the leader stage, please use the related CRS envelopment or multiplier model.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_14) contains supplementary material, which is available to authorized users.

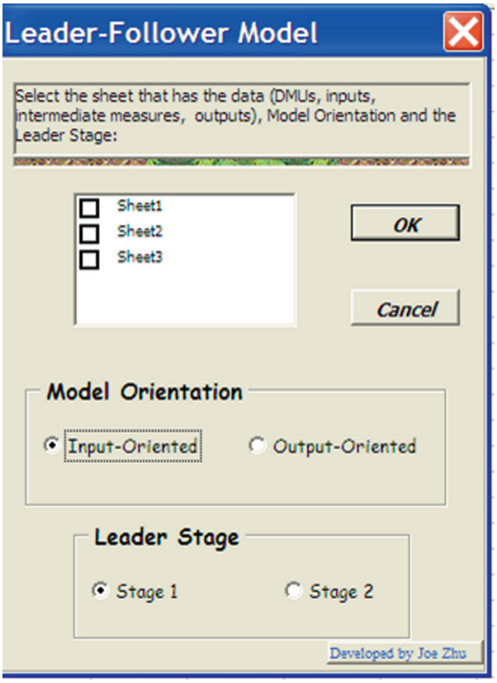


Fig. 14.9 Two-stage network process: leader-follower model

References

Chen, Y., Liang, L., & Zhu, J. (2009). Equivalence in two-stage DEA approaches. *European Journal of Operational Research, 193,* 600–604.

Chen, Y., & Zhu, J. (2004). Measuring information technology's indirect impact on firm performance. *Information Technology Management Journal, 5*(1–2), 9–22.

Chen, Y., Cook, W. D., & Zhu, J. (2010). Deriving the DEA frontier for two-stage processes. *European Journal of Operational Research, 202,* 138–142.

Chilingerian, J. A., & Sherman, H. D. (2004). Health care applications: From hospitals to physicians, from productive efficiency to quality frontiers. In W. W. Cooper, L. M. Seiford, & J. Zhu (Eds.), *Handbook on data envelopment analysis, Chapter 17*. Boston: Kluwer Academic Publishers.

Kao, C., & Hwang, S.-N. (2008). Efficiency decomposition in two-stage data envelopment analysis: An application to non-life insurance companies in Taiwan. *European Journal of Operational Research, 185*(1), 418–429.

Liang, L., Cook, W. D., & Zhu, J. (2008). DEA models for two-stage processes: Game approach and efficiency decomposition. *Naval Research Logistics, 55,* 643–653.

Seiford, L. M., & Zhu, J. (1999). Profitability and marketability of the top 55 US commercial banks. *Management Science, 45*(9), 1270–1288.

Wang, C. H., Gopal, R. D., & Zionts, S. (1997). Use of data envelopment analysis in assessing information technology impact on firm performance. *Annals of Operations Research, 73,* 191–213.

Chapter 15
Models for Evaluating Supply Chains and Network Structures

15.1 Supply Chain Efficiency

So far, the value-added processes or systems have been treated as a "black-box". We examine the resources available to the processes or systems and monitor the "conversions" of these resources (inputs) into the desired outputs. However, each process or system can include many subprocesses. For example, if the process is to make a car, then important subprocesses include assembling and painting. If we evaluate the efficiency of a supply chain system, then we need to measure the performance of each individual supply chain components, including suppliers, manufacturers, retailers, and customers.

While there are studies on supply chain performance using DEA, the research has been focused on a single member of the supply chain.

Within the context of DEA, there are a number of methods that have the potential to be used in supply chain efficiency evaluation. Seiford and Zhu (1999a) and Chen and Zhu (2004) provide two approaches in modeling efficiency as a two-stage process. Färe and Grosskopf (2000) develop the network DEA approach to model general multi-stage processes with intermediate inputs and outputs. Golany, Hackman and Passy (2006) provide an efficiency measurement framework for systems composed of two subsystems arranged in series that simultaneously compute the efficiency of the aggregate system and each subsystem.

Note that an effective management of the supply chain requires knowing the performance of the overall chain rather than simply the performance of the individual supply chain members. Each supply chain member has its own strategy to achieve efficiency; however, what is best for one member may not work in favor of another member. Sometimes, because of the possible conflicts between supply chain members, one member's inefficiency may be caused by another's efficient operations. For example, the supplier may increase its raw material price to enhance its revenue and to achieve an efficient performance. This increased revenue means increased cost to the manufacturer. Consequently, the manufacturer may become inefficient unless it adjusts its current operating policy. Measuring supply chain performance becomes a difficult and challenging task because of the need to deal with the mul-

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_15,

Table 15.1 Simple supplier-manufacturer example

	Manufacturer						
	Supplier						
Supply chain	Shipping costs	Labor	Revenue material costs	Profit	Supplier efficiency	Manufacturer efficiency	Overall efficiency
A	7	9	4	16	1	0.75	1
B	9	4	6	14	1	0.5	1
C	11	6	3	23	0.791	1	1

tiple performance measures related to the supply chain members, and to integrate and coordinate the performance of those members.

Two hurdles are present in measuring the performance of value chains. One is the existence of multiple measures that characterize the performance of each member in a supply chain. The other is the existence of conflicts between supply chain members with respect to specific measures.

Consider three supplier-manufacturer supply chains presented in Table 15.1 where the supplier has two inputs (shipping cost and labor) and one output (revenue from selling the raw materials to the manufacturer), and the manufacture has one input (raw material cost which is the supplier's revenue) and one output (profit).

Applying the input-oriented VRS envelopment model to the suppliers and the manufacturers indicate that the suppliers in supply chains A and B, and the manufacturer in supply chain C are efficient. Now, if we ignore the intermediate measure of revenue (raw material cost) and apply the input-oriented VRS envelopment model, the last column of Table 15.1 indicates that all supply chains are efficient.

This simple numerical example indicates that the conventional DEA fails to correctly characterize the performance of supply chain. Since an overall DEA efficient performance does not necessarily indicate efficient performance in individual components in the supply chain. Consequently, improvement to the best-practice can be distorted. i.e., the performance improvement of one member affects the efficiency status of the other, because of the presence of intermediate measures. Seiford and Zhu (1999a) develop a procedure for value chain performance improvement by using returns to scale (RTS) sensitivity analysis (Seiford and Zhu 1999b). In this chapter, we present models that can directly evaluate the performance of supply chains or value chains that have more than one members/components.

15.2 Supply Chain Efficiency

Supply chain management has been proven a very effective tool to provide prompt and reliable delivery of high-quality products and services at the least cost. To achieve this, performance evaluation of entire supply chain is extremely important, since it means utilizing the combined resources of the entire supply chain in the most efficient way possible to provide market-wining and cost-effective products and services. However, a lack of appropriate performance measurement systems has been a major obstacle to an effective supply chain management (Lee and Billington 1992).

This is due to the fact that the concept of supply chain management requires the performance of overall supply chain rather than only the performance of the individual supply chain members. Each supply chain member has its own strategy to achieve 100% efficiency. One supply chain member's 100% efficiency does not necessarily mean another's 100% efficiency. Sometimes, because of the possible conflicts between supply chain members, one member's inefficiency may be caused by another's efficient operations. For example, the supplier may increase its raw material price to increase its revenue and to achieve an efficient performance. This increased revenue means increased cost to the manufacturer. Consequently, the manufacturer may become inefficient unless the manufacturer adjusts its current operating policy.

As demonstrated in Table 15.1, some measures linked to related supply chain members cannot be simply classified as "outputs" or "inputs" of the supply chain. For example, the supplier's revenue is not only an output of the supplier (the supplier wishes to maximize it), but also an input to the manufacturer (the manufacturer wishes to minimize it). Simply minimizing the total supply chain cost or maximizing the total supply chain revenue (profit) does not model and solve the conflicts. Therefore, the meaning of supply chain efficiency needs to be carefully defined and studied, and we need models that can both define and measure the efficiency of supply chain as well as supply chain members.

Methods have been developed to estimate the exact performance of supply chain members based upon single performance measures (e.g., Cheung and Hansman 2000). However, no attempts have been made to identify the best practice of the supply chain. No solid mathematical models have been developed to simultaneously (i) define and measure the whole supply chain performance with possible conflicts on specific measures, (ii) evaluate the performance of supply chain members, and (iii) identify the best practice and provide directions to achieve the supply chain best practice.

The new DEA model measures the efficiency of supply chain system as a whole as well as each supply chain member, and provides directions for supply chain improvement to reach the best practice. This eliminates the needs for unrealistic assumptions in typical supply chain optimization models and probabilistic models, e.g., a typical EOQ model assumes constant and known demand rate and lead-time for delivery.

15.2.1 Supply Chain as an Input-Output System

A typical supply chain can be presented in Fig. 15.1 with four echelons—suppliers, manufacturers, distributors, and retailers. The traditional objective of supply chain management is to minimize the total supply chain cost to meet the customer needs

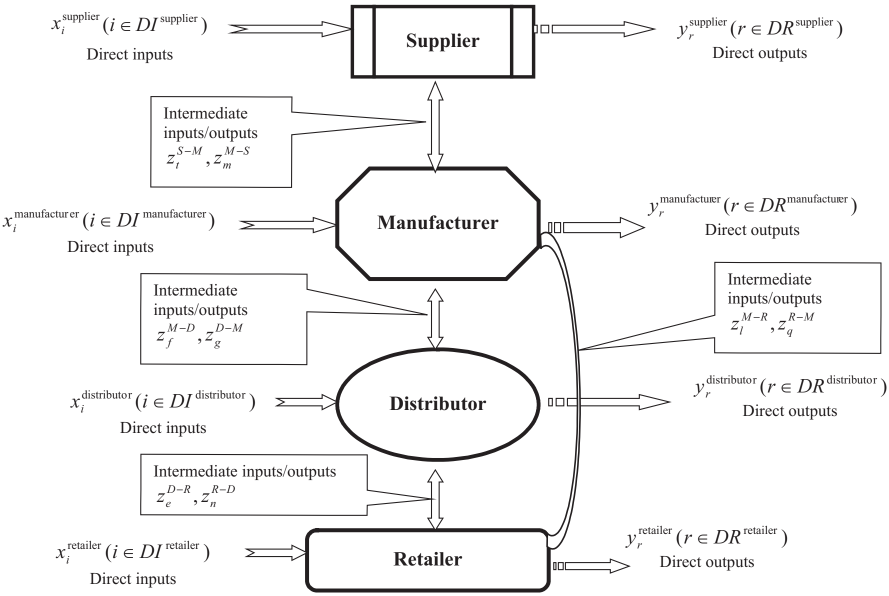


Fig. 15.1 Supply chain

through coordination efforts among supply chain members. To achieve this objective, timely and accurate assessment of the supply chain system and individual member performance is of extreme importance. Because an effective performance evaluation system (i) provides the basis to understand the supply chain operations, (ii) monitors and manages supply chain performance through identifying the best-practice supply chain operations, and (iii) provides directions for further supply chain improvement.

Supply chain systems can be viewed as an integrated input-output system where each supply chain member uses inputs to produce. Consequently, we may classify supply chain member's performance measures into inputs and outputs. Caution should be paid when we classify the performance measures into inputs and outputs based upon specific supply chain members, since incorrect classification may lead to false conclusion on the efficiency of supply chain members as well as supply chain. The classification can be based upon the material and information flows in a supply chain system.

Let I^Δ and R^Δ represent the input and output subscript sets for a supply chain member Δ, respectively. We denote $x_i^\Delta(i \in I^\Delta)$ and $y_r^\Delta(r \in R^\Delta)$ the inputs and outputs associated with each supply chain member, respectively. Now, let $\vec{x}_\Delta$ and $\vec{y}_\Delta$ be the vectors consisting of $x_i^\Delta(i \in I^\Delta)$ and $y_r^\Delta(r \in R^\Delta)$, respectively. The following Pareto-Koopmans efficiency is used to define an efficient supply chain member.

Definition 15.1 (Efficient Supply Chain Member): A supply chain member Δ is efficient *if* $(\vec{x}_\Delta, \vec{y}_\Delta)$ is not dominated.

Some measures are associated with a specific supply chain member only. We call these measures the "direct" inputs and outputs. For example, labor and manufacturing lead time are two direct inputs to the manufacturer. These direct inputs and outputs of supply chain members can be viewed as the inputs and outputs of the supply chain.

We also have "intermediate" inputs/outputs associated with two supply chain members. For example, the number of finished products shipped from the manufacturer to the retailer and the distributor represent outputs of the manufacturer. These outputs then become inputs to the distributor and the retailer (see Fig. 15.1). These intermediate measures cannot be simply treated as inputs or outputs of the supply chain, although they are inputs/outputs of specific supply chain members.

In supply chain management, it is believed that values of intermediate measures should be determined through coordination among related supply chain members (Parlar and Weng 1997; Thomas and Griffin 1996). Because such intermediate measures are usually cost to one supply chain member and benefit to the other. Simply minimizing the total supply chain cost or maximizing the supply chain revenue does not model situations with intermediate measures. This poses a challenge to defining and measuring the supply chain efficiency.

To facilitate our discussion, let DI^{Δ} and DR^{Δ} represent the direct input and direct output subscript sets for a supply chain member Δ, respectively. We then use the following notions to represent intermediate inputs and outputs, x_i^{Δ} $(i \notin DI^{\Delta} \subseteq I^{\Delta})$ and $y_r^{\Delta} (r \notin DR^{\Delta} \subseteq R^{\Delta})$,

z_t^{S-M} = tth intermediate output from the supplier to manufacturer, $t=1,\ldots,$ T;

z_m^{M-S} = mth intermediate output from the manufacturer to the supplier, $m=1,\ldots,$ M;

z_f^{M-D} = fth intermediate output from the manufacturer to the distributor, $f=1,\ldots,$ F;

z_g^{D-M} = gth intermediate output from the distributor to the manufacturer, $g=1,\ldots,$ G;

z_l^{M-R} = lth intermediate output from the manufacturer to the retailer, $l=1,\ldots,$ L;

z_q^{R-M} = qth intermediate output from the retailer to the manufacturer, $q=1,\ldots,$ Q;

z_e^{D-R} = eth intermediate output from the distributor to the retailer, $e=1,\ldots,$ E;

z_n^{R-D} = nth intermediate output from the retailer to the distributor, $n=1,\ldots,$ N.

Note that only intermediate outputs are defined, since each such output also represents an input to an associated supply chain member. For example, z_t^{S-M} (output of the supplier) also represents an input to the manufacturer.

15.2.2 Supply Chain Efficiency Model

Suppose we have J observations associated with each supply chain member. i.e., we have observed input and output values of $x_{ij}^{\Delta}(i \in I^{\Delta})$ and $y_{rj}^{\Delta}(r \in R^{\Delta})$, where $j = 1, \ldots, J$. The efficiency of supply chain member Δ can be measured by the following DEA model—input-oriented CRS envelopment model

$$\theta^{\Delta^*} = \min_{\phi_j^{\Delta}, \theta^{\Delta}} \theta^{\Delta} \tag{15.1}$$

subject to

$$\sum_{j=1}^{J} \phi_j^{\Delta} x_{ij}^{\Delta} \le \theta^{\Delta} x_{ij_o}^{\Delta} \; i \in I^{\Delta}$$

$$\sum_{j=1}^{J} \phi_j^{\Delta} y_{rj}^{\Delta} \ge y_{rj_o}^{\Delta} \; r \in R^{\Delta}$$

$$\phi_j^{\Delta} \ge 0, j = 1, \ldots, J$$

If $\theta^{\Delta^*} = 1$, then a supply chain member Δ is efficient (including weakly efficient). Also, for inefficient performance, model (15.1) provides projection paths onto the efficient frontier via the optimal values of $\sum_{j=1}^{J} \phi_j^{\Delta^*} x_{ij}^{\Delta}$ and $\sum_{j=1}^{J} \phi_j^{\Delta^*} y_{rj}^{\Delta}$.

Because of the possible conflicts represented by the intermediate measures between associated supply chain members, the supply chain's performance cannot be simply defined and characterized by non-dominancy through using model (15.1). Let w_i be the user-specified weights reflecting the preference over supply chain member's performance (operation). We establish the following liner programming problem for the supply chain

$$\Omega^* = \min_{\Omega_i, \lambda_j, \beta_j, \delta_j, \gamma_j, \tilde{z}} \frac{\sum_{i=1}^{4} w_i \Omega_i}{\sum_{i=1}^{4} w_i} \tag{15.2}$$

subject to

(supplier)

$$\sum_{j=1}^{J} \lambda_j x_{ij}^{\text{supplier}} \le \Omega_1 x_{ij_o}^{\text{supplier}} \qquad i \in DI^{\text{supplier}}$$

$$\sum_{j=1}^{J} \lambda_j y_{rj}^{\text{supplier}} \ge y_{rj_o}^{\text{supplier}} \qquad r \in DR^{\text{supplier}}$$

$$\sum_{j=1}^{J} \lambda_j z_{tj}^{\text{S-M}} \ge \tilde{z}_{tj_o}^{\text{S-M}} \qquad t = 1, \ldots, \text{T}$$

$$\sum_{j=1}^{J} \lambda_j z_{mj}^{\text{M-S}} \le \tilde{z}_{mj_o}^{\text{M-S}} \qquad m = 1, \ldots, \text{M}$$

$$\lambda_j \ge 0, \quad j = 1, \ldots, J$$

$$
\begin{aligned}
&(\textit{manufacturer})\\
&\sum_{j=1}^{J} \beta_j x_{ij}^{\text{manufacturer}} \le \Omega_2 x_{ij_o}^{\text{manufacturer}} && i \in \mathrm{DI}^{\text{manufacturer}}\\
&\sum_{j=1}^{J} \beta_j y_{rj}^{\text{manufacturer}} \ge y_{rj_o}^{\text{manufacturer}} && r \in \mathrm{DR}^{\text{manufacturer}}\\
&\sum_{j=1}^{J} \beta_j z_{tj}^{\text{S-M}} \le \tilde{z}_{tj_o}^{\text{S-M}} && t = 1,\ldots,\mathrm{T}\\
&\sum_{j=1}^{J} \beta_j z_{mj}^{\text{M-S}} \ge \tilde{z}_{mj_o}^{\text{M-S}} && m = 1,\ldots,\mathrm{M}\\
&\sum_{j=1}^{J} \beta_j z_{fj}^{\text{M-D}} \ge \tilde{z}_{fj_o}^{\text{M-D}} && f = 1,\ldots,\mathrm{F}\\
&\sum_{j=1}^{J} \beta_j z_{gj}^{\text{D-M}} \le \tilde{z}_{gj_o}^{\text{D-M}} && g = 1,\ldots,\mathrm{G}\\
&\sum_{j=1}^{J} \beta_j z_{lj}^{\text{M-R}} \ge \tilde{z}_{lj_o}^{\text{M-R}} && l = 1,\ldots,\mathrm{L}\\
&\sum_{j=1}^{J} \beta_j z_{qj}^{\text{R-M}} \le \tilde{z}_{qj_o}^{\text{R-M}} && q = 1,\ldots,\mathrm{Q}\\
&\beta_j \ge 0, \quad j = 1,\ldots,J\\
&\textit{(distributor)}\\
&\sum_{j=1}^{J} \delta_j x_{ij}^{\text{distributor}} \le \Omega_3 x_{ij_o}^{\text{distributor}} && i \in \mathrm{DI}^{\text{distributor}}\\
&\sum_{j=1}^{J} \delta_j y_{rj}^{\text{distributor}} \ge y_{rj_o}^{\text{distributor}} && r \in \mathrm{DR}^{\text{distributor}}\\
&\sum_{j=1}^{J} \delta_j z_{fj}^{\text{M-D}} \le \tilde{z}_{fj_o}^{\text{M-D}} && f = 1,\ldots,\mathrm{F}\\
&\sum_{j=1}^{J} \delta_j z_{gj}^{\text{D-M}} \ge \tilde{z}_{gj_o}^{\text{D-M}} && g = 1,\ldots,\mathrm{G}\\
&\sum_{j=1}^{J} \delta_j z_{ej}^{\text{D-R}} \ge \tilde{z}_{ej_o}^{\text{D-R}} && e = 1,\ldots,\mathrm{E}\\
&\sum_{j=1}^{J} \delta_j z_{nj}^{\text{D-R}} \le \tilde{z}_{nj_o}^{\text{D-R}} && n = 1,\ldots,\mathrm{N}\\
&\delta_j \ge 0, \quad j = 1,\ldots,J
\end{aligned}
$$

(retailer)

$$\sum_{j=1}^{J} \gamma_j x_{ij}^{\text{retailer}} \le \Omega_4 x_{ij_o}^{\text{retailer}} \qquad i \in \text{DI}^{\text{retailer}}$$

$$\sum_{j=1}^{J} \gamma_j y_{rj}^{\text{retailer}} \ge y_{rj_o}^{\text{retailer}} \qquad r \in \text{DR}^{\text{retailer}}$$

$$\sum_{j=1}^{J} \gamma_j z_{lj}^{\text{M-R}} \le \tilde{z}_{lj_o}^{\text{M-R}} \qquad l = 1,\ldots,\text{L}$$

$$\sum_{j=1}^{J} \gamma_j z_{qj}^{\text{R-M}} \ge \tilde{z}_{qj_o}^{\text{R-M}} \qquad q = 1,\ldots,\text{Q}$$

$$\sum_{j=1}^{J} \gamma_j z_{ej}^{\text{D-R}} \le \tilde{z}_{ej_o}^{\text{D-R}} \qquad e = 1,\ldots,\text{E}$$

$$\sum_{j=1}^{J} \gamma_j z_{nj}^{\text{R-D}} \ge \tilde{z}_{nj_o}^{\text{R-D}} \qquad n = 1,\ldots,\text{N}$$

$$\gamma_j \ge 0, \quad j = 1,\ldots,J$$

Additional constraints can be added into model (15.2). For example, if z_f^{M-D} represents the number of product *f* shipped from the manufacturer to the distributor, and if the capacity of this manufacturer in producing product *f* is C_f, then we may add $\tilde{z}_f^{M-D} \le C_f$.

Obviously, if $\Omega^* = 1$, then there must exists an optimal solution such that $\lambda_{j_o}^* = \beta_{j_o}^* = \delta_{j_o}^* = \gamma_{j_o}^* = 1$, where (*) represents optimal value in model (15.2). Further, if $\Omega^* = 1$, then $\theta^{\Delta *} = 1$, where $\theta^{\Delta *}$ is the optimal value to model (15.1). i.e., when $\Omega^* = 1$, all supply chain members are efficient.

If $\Omega^* \ne 1$, then we immediately have the following result

All supply chain members are efficient with respect to $\Omega_1^* x_{ij_o}^{\text{supplier}} (i \in DI^{\text{supplier}})$, $\Omega_2^* x_{ij_o}^{manufacturer} (i \in DI^{manufacturer})$, $\Omega_3^* x_{ij_o}^{distributor} (i \in DI^{distributor})$, $\Omega_4^* x_{ij_o}^{retailer} (i \in DI^{retailer})$, $y_{rj_o}^{\text{supplier}} (r \in DR^{\text{supplier}})$, $y_{rj_o}^{manufacturer} (r \in DR^{manufacturer})$, $y_{rj_o}^{distributor} (r \in DR^{distributor})$, $y_{rj_o}^{retailer} (r \in DR^{retailer})$, $\tilde{z}_{tj_o}^{S-M*} (t = 1, \ldots, T)$, $\tilde{z}_{mj_o}^{M-S*} (m = 1,\ldots, M)$, $\tilde{z}_{fj_o}^{M-D*} (f = 1, \ldots, F)$, $\tilde{z}_{gj_o}^{D-M*} (g = 1, \ldots, G)$, $\tilde{z}_{lj_o}^{M-R*} (l = 1, \ldots, L)$, $\tilde{z}_{qj_o}^{R-M*} (q = 1, \ldots, Q)$, $\tilde{z}_{ej_o}^{D-R*} (e = 1, \ldots, E)$, *where (*) represents optimal value in model (15.2).*

Definition 15.2 (Efficient Supply Chain): A supply chain is efficient if $\Omega^* = 1$, where Ω^* is the optimal value to model (15.2).

$\theta^{\Delta *}$ measures the efficiency of supply chain member Δ under the context of supply chain member best practice. Ω_i^* can actually be used as a new efficiency measure for a specific supply chain member under the context of supply chain best practice. We have

Definition 15.3: Ω_i^* is called supply-chain-best-practice-dependent efficiency score for a specific supply chain member.

Note that Ω^* can be viewed as an index for input or cost savings for (inefficient) supply chains. The smaller the Ω^*, more savings could be achieved to reach the best practice. The same observation can also be applied to $\theta^{\Delta*}$ in the context of supply chain member best practice. Let $(w_1\theta^{\text{supplier}*} + w_2\theta^{\text{manufacturer}*} + w_3\theta^{\text{distributor}*} + w_4\theta^{\text{retailer}*}) / \sum_{i=1}^{4} w_i$ represent the index for input savings achievable by all supply chain members combined. The following Theorem indicates that supply chain as a whole has potential to achieve more input savings and a better performance

Theorem 15.1: $\Omega^* \leq (w_1\theta^{\text{supplier}*} + w_2\theta^{\text{manufacturer}*} + w_3\theta^{\text{distributor}*} + w_4\theta^{\text{retailer}*}) / \sum_{i=1}^{4} w_i$.

15.2.3 An example

We establish a spreadsheet model for a numerical example constructed as follows. For the supplier, we use labor and operating cost as two direct inputs, and revenue as the intermediate output. This revenue becomes an intermediate input of the manufacturer.

For the manufacturer, we use manufacturing cost and manufacturing lead time as two direct inputs, in addition to the intermediate input—supplier's revenue. We also have three intermediate manufacturer outputs: number of products shipped to the distributor, number of products shipped to the retailer, and distributor's fill rate. These outputs then become inputs to the distributor and the retailer. Note that the distributor's fill rate is actually a cost measure to the distributor, since the fill rate is associated with inventory holding cost and the amount of products required from the manufacturer. The distributor's fill rate implies benefit to the manufacturer, since more products are needed from the manufacturer (meaning more revenue to the manufacturer) if the distributor wishes to maintain a higher fill rate. Thus, the distributor's fill rate is treated as an output from the manufacturer and an input to the distributor. From a distributor's point of view, the distributor always tries to meet the needs of its customer while maintaining a fill rate as low as possible, because unnecessary high fill rate incurs additional cost to the distributor.

For the distributor, we use customer response time and distribution cost as two direct inputs in addition to the above intermediate inputs linked with the manufacturer. Two intermediate outputs from the distributor are the number of products shipped from the distributor to the retailer, and the percentage of on-time delivery.

For the retailer, in addition to the intermediate inputs from the manufacturer and the distributor, we have one direct input of retailer cost, and one direct output of profit. Figure 15.2 presents the data with ten observations, i.e., J = 10.

In Fig. 15.2, cells D18:M21 represents λ_j, β_j, δ_j, and γ_j. Cell B25 indicates the observation under evaluation. Cells D23:D26 represents Ω_i (i = 1, 2, 3, 4). Cell C25 is the objective function of model (15.2), and contains the formula "=(D23+D24+D25+D26)/4".

	A	B	C	D	E	F	G	H	I	J	K	L	M
1			**Observation**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
2			Supplier-labor	150	140	130	165	170	145	155	175	160	125
3			Supplier-cost	130	150	165	170	200	185	135	190	185	190
4			Supplier-revenue	20	21	23	24	27	25	24	30	28	25
5			mfg cost	125	120	110	150	146	115	105	100	135	120
6			mfg time	3	2	3	4	2	3	2	2	4	3
7			DC cost	90	100	80	70	85	77	78	90	78	68
8			customer res time	3	3	2	4	2	2	1	3	2	1
9			fill rate	70%	90%	78%	88%	73%	95%	89%	87%	95%	90%
10			product DC-retailer	1800	2000	2400	2300	2500	2500	2000	2000	2500	2300
11			on-time	96%	95%	97%	89%	99%	89%	93%	88%	99%	83%
12			product mfg-retaile	2000	2100	2500	1900	2600	2300	2200	2300	2500	2500
13			product mfg-DC	500	300	450	200	300	250	350	450	300	400
14			Retailer cost	100	110	130	125	140	135	125	155	135	130
15			Retailer revenue	310	220	300	230	320	240	350	370	325	355
16													
17													
18			λ	0	0	0	0	0	0	0.884	0	0	0
19			β	0	1.011	0	0	0	0	0	0	0	0
20			δ	0	0	0.07784	0	0	0	0	0	0.893	0
21			γ	1	0	0	0	0	0	0	0	0	3E-12
22													
23				0.92			Supply Chain						
24				0.97									
25		1	0.932955109	0.84									
26				1		Efficienc	supplier		mfg		DC		Retailer
27	Supplier-labor	150	S-labor	138		137.733	137.05						
28	Supplier-cost	130	S-cost	119		119.368	119.37						
29	Supplier-revenue	20	S-revenue	21.2			21.221		21.2				
30	mfg cost	125	mfg cost	121		121.263			121				
31	mfg time	3	mfg time	2.91		2.91032			2.02				
32	DC cost	90	DC cost	75.9		75.9147					75.91		
33	customer res time	3	customer res time	2.53		2.53049					1.943		
34	fill rate	0.7	fill rate	0.91					0.91		0.909		
35	product DC-retailer	1800	product DC-retailer	2420							2420		1800
36	on-time	0.96	on-time	0.96					0.96		0.96		0.96
37	product mfg-retaile	2000	product mfg-retaile	2122					2122				2000
38	product mfg-DC	500	product mfg-DC	303					303		303.1		
39	Retailer cost	100	Retailer cost	100		100							100
40	Retailer revenue	310	Retailer revenue										310

Fig. 15.2 Supply chain efficiency spreadsheet model

Cells B27:B40 record the performance measures for a specific observation under evaluation. Cell B27 contains the formula "=INDEX (D2:M2,1,$B$25) which is copied into cell B28:B40.

Cells D27:D39 are used to represent the decision variables. The formulas used in the rest of the spreadsheet model shown in Fig. 15.2 are

Cell F27=$D$23*B27
Cell F28=$D$23*B28
Cell F30=$D$24*B30
Cell F31=$D$24*B31
Cell F32=$D$25*B32
Cell F33=$D$25*B33
Cell F39=$D$26*B39

Cell G27=SUMPRODUCT($D$18:$M$18,D2:M2)
Cell G28=SUMPRODUCT($D$18:$M$18,D3:M3)
Cell G29=SUMPRODUCT($D$18:$M$18,D4:M4)

Cell I29=SUMPRODUCT($D$19:$M$19,D4:M4)
Cell I30=SUMPRODUCT($D$19:$M$19,D5:M5)
Cell I31=SUMPRODUCT($D$19:$M$19,D6:M6)
Cell I34=SUMPRODUCT($D$19:$M$19,D9:M9)
Cell I36=SUMPRODUCT($D$19:$M$19,D11:M11)
Cell I37=SUMPRODUCT($D$19:$M$19,D12:M12)
Cell I38=SUMPRODUCT($D$19:$M$19,D13:M13)

Cell K32=SUMPRODUCT($D$20:$M$20,D7:M7)
Cell K33=SUMPRODUCT($D$20:$M$20,D8:M8)
Cell K34=SUMPRODUCT($D$20:$M$20,D9:M9)
Cell K35=SUMPRODUCT($D$20:$M$20,D10:M10)
Cell K36=SUMPRODUCT($D$20:$M$20,D11:M11)
Cell K38=SUMPRODUCT($D$20:$M$20,D13:M13)

Cell M35=SUMPRODUCT($D$21:$M$21,D10:M10)
Cell M36=SUMPRODUCT($D$21:$M$21,D11:M11)
Cell M37=SUMPRODUCT($D$21:$M$21,D12:M12)
Cell M39=SUMPRODUCT($D$21:$M$21,D14:M14)
Cell M40=SUMPRODUCT($D$21:$M$21,D15:M15)

Figure 15.3 shows the Solver parameters for the spreadsheet shown in Fig. 15.2 where cells D18:M21 and cells D23:D39 are changing cells. In this case, two additional constraints are added into model (15.2). One is "fill rate ≤ 100 %", and the other "percentage of on-time delivery ≤ 100 %". The constraints include

$D$23:$D$26<=1
$D$34<=1
$D$36<=1
$F$27:$F$28=$D$27:$D$28
$F$30:$F$33=$D$30:$D$33
$F$39=$D$39
$G$27:$G$28<=$D$27:$D$28
$G$29>=$D$29
$I$29:$I$31<=$D$29:$D$31
$I$34>=$D$34
$I$36:$I$38>=$D$36:$D$38
$K$32:$K$34<=$D$32:$D$34
$K$35:$K$36>=$D$35:$D$36
$K$38<=$D$38
$M$35:$M$37<=$D$35:$D$37
$M$39<=$D$39
$M$40>=$B$40

Table 15.2 reports the efficiency scores, optimal values to models (15.1) and (15.2) with $w_i = 1$ ($i = 1, \ldots, 4$).

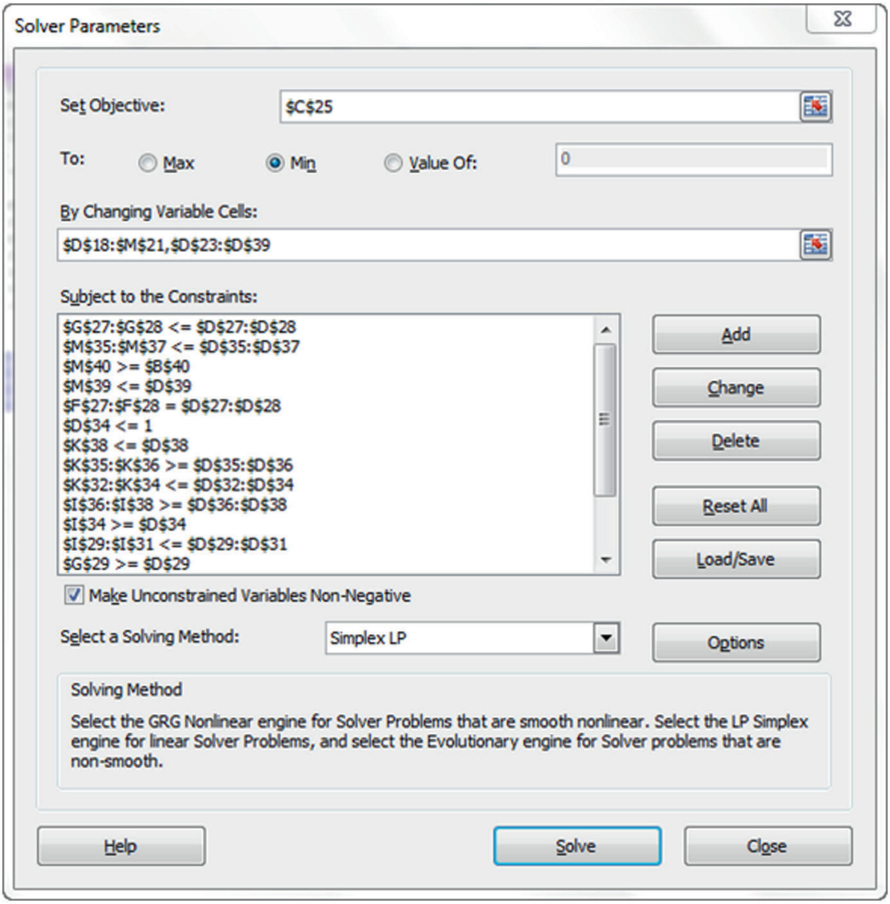


Fig. 15.3 Solver parameters for supply chain efficiency

Columns 2–5 characterize the performance of supply chain members based upon model (15.1). The sixth column reports the average efficiency score of the supply chain members. The supply chain performance is reported in the seventh column with Ω_i^* reported in the last four columns.

Although a number of observations on supply chain members are efficient, only one supply chain performance (observation 7) is efficient. i.e., the observation 7 represents the best practice of the supply chain system. Note that in this case, all supply chain members are efficient.

We observe that the average supply chain member efficiency score (column 6) is greater than the supply chain efficiency score (Ω^*). For example, consider observation 5 where two supply chain members (manufacturer and distributor) are efficiently operating. The average supply chain member efficiency score is 0.79456

Table 15.2 Supply chain efficiency

	Member efficiency					Supply chain efficiency				
	Supplier	Manufacturer	Distributor	Retailer	Average	Supply Chain	Supplier	Manufacturer	Distributor	Retailer
Observation	$\theta^{supplier*}$	$\theta^{manufacturer*}$	$\theta^{distributor*}$	$\theta^{retailer*}$		Ω^*	Ω_1^*	Ω_2^*	Ω_3^*	Ω_4^*
1	0.865	1	1	1	0.966	0.933	0.918	0.970	0.843	1
2	0.881	1	0.880	0.673	0.859	0.669	0.599	0.624	0.465	0.714
3	0.964	1	1	0.810	0.944	0.791	0.720	0.747	0.747	0.948
4	0.870	0.856	1	0.754	0.870	0.576	0.518	0.460	0.658	0.667
5	0.895	1	1	0.820	0.929	0.795	0.688	0.768	0.768	0.954
6	0.937	0.999	1	0.673	0.902	0.613	0.529	0.579	0.625	0.717
7	1	1	1	1	1	1	1	1	1	1
8	1	1	0.811	1	0.953	0.943	1	1	0.770	1
9	0.994	0.904	1	0.856	0.938	0.783	0.696	0.694	0.874	0.868
10	1	0.986	1	1	0.997	0.992	1	0.968	1	1

and the supply chain efficiency score is 0.79456, indicating that the supply chain system could achieve more input savings.

Model (15.2) yields optimal values on the performance measures for (an inefficient) supply chain to reach the best practice. Consider observation 1 in Fig. 15.8 where a set of optimal solutions is shown in cells D27:D39. Since $\Omega_4^* = 1$ indicating the retailer is efficient, no adjustments for measures related to the retailer are required. However, in order to reach the best practice, the supplier, the manufacturer and the distributor should reduce their "direct inputs" based upon Ω_i^* ($i =$ 1, 2, 3). In addition, the supplier and the manufacturer should reach an agreement on the selling price of raw materials to increase the supplier's revenue by 6 %. The distributor's fill rate should be increased to 90.95 % (from the current rate of 70 %). The products shipped from the manufacturer to the distributor should be reduced by 39 %. This solution indicates that based upon the best practice, the distributor should be able to maintain the fill rate of 90.95 % while the manufacturer reduces its shipment to the distributor.

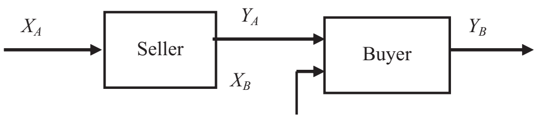
Additional managerial information is available from the optimal values of λ_j^*, β_j^*, δ_j^*, and γ_j^*, since they provide information on which observations of supply chain members are used as benchmarks. For example, when the observation 1 is under evaluation by model (15.2), we have (i) $\lambda_7^* =$ 0.884, indicating that the supplier in observation 7 is used as the benchmark; (ii) $\beta_2^* =$ 1.011, indicating that the manufacturer in observation 2 is used as the benchmark; (iii) $\delta_3^* =$ 0.08 and $\delta_9^* =$ 0.893, indicating that the distributor in observations 3 and 9 is used as the benchmark; and (iv) $\gamma_1^* = 1$, indicating that the retailer in observation 1 is efficient and itself is used as the benchmark.

Some supply chains may choose to operate with high cost and high availability while others are lean with lower levels of service. The notion of DEA efficiency (i) provides an approach for characterizing and measuring the efficiency of supply chain as well as supply chain members, and (ii) makes it clear that two supply chains may have different input-output mix yet both may be efficient. Model (15.2) enables supply chain members to collectively improve the supply chain performance. Through the use of model (15.2), any supply chains can find ways to achieve best-practice performance and to gain a competitive edge. The approach also provides information on which supply chain members are used as a benchmark when a specific supply chain observation is under evaluation.

15.3 Cooperative and Non-Cooperative Approaches

In this section, we present several models due to Liang, Feng, Cook and Zhu (2006) that directly evaluate the performance of the supply chain as well as its members, while considering the relationship between the buyer and the seller. The modeling processes are based upon the concept of non-cooperative and cooperative games (see, e.g., Simaan and Cruz 1973; Li et al. 1995; Huang 2000).

Fig. 15.4 Seller-buyer supply chain



Suppose there are N similar supply chains or N observations on one supply chain. Consider a buyer-seller supply chain as described in Fig. 15.4, where for $j=1,\ldots, N$, $X_A=(x_{ij}^A, i=1,\ldots, I)$ is the input vector of the seller, and $Y_A=(y_{rj}^A, \mathrm{r}=1,\ldots, \mathrm{R})$ is the seller's output vector. Y_A is also an input vector of the buyer. The buyer also has an input vector $X_B=(x_{sj}^B, s=1,\ldots, S)$ and the output vector for the buyer is Y_B $(=y_{tj}^B, t=1,\ldots, T)$.

15.3.1 The Non-Cooperative Model

We propose the seller-buyer interaction be viewed as a two-stage non-cooperative game with the seller as the leader and the buyer as the follower. First, we use the CRS (ratio) model to evaluate the efficiency of the seller as the leader:

$$\text{Maximize}\, E_{AA} = \frac{\sum_{r=1}^{R} u_r^A y_{r0}^A}{\sum_{i=1}^{I} v_i^A x_{i0}^A}$$

subject to

$$\frac{\sum_{r=1}^{R} u_r^A y_{rj}^A}{\sum_{i=1}^{I} v_i^A x_{ij}^A} \le 1 \qquad j = 1, \ldots, N \tag{15.3}$$

$$u_r^A, v_i^A \ge 0 \quad r = 1, \ldots, R, i = 1, \ldots, I$$

This model is equivalent to the following standard CRS multiplier model:

$$\text{Maximize}\, E_{AA} = \sum_{r=1}^{R} \mu_r^A y_{r0}^A$$

subject to

$$\sum_{i=1}^{I} \omega_i^A x_{ij}^A - \sum_{r=1}^{R} \mu_r^A y_{rj}^A \ge 0 \quad j = 1, \ldots, N \tag{15.4}$$

$$\sum_{i=1}^{I} \omega_i^A x_{i0}^A = 1$$

$$\mu_r^A, \omega_i^A \ge 0 \qquad r = 1, \ldots, R, i = 1, \ldots, I$$

Suppose we have an optimal solution of model (15.4) $\mu_r^{A*}, \omega_i^{A*}$, E_{AA}^* $(r=1, \ldots, R, i=1, \ldots, I)$, and denote the seller's efficiency as E_{AA}^*. We then use the following model to evaluate the buyer's efficiency:

$$
\begin{aligned}
& \textit{Maximize } E_{AB} = \frac{\sum_{t=1}^{T} u_t^B y_{t0}^B}{D \times \sum_{r=1}^{R} \mu_r^A y_{r0}^A + \sum_{s=1}^{S} v_s^B x_{s0}^B} \\
& \text{subject to} \\
& \frac{\sum_{t=1}^{T} u_t^B y_{tj}^B}{D \times \sum_{r=1}^{R} \mu_r^A y_{rj}^A + \sum_{s=1}^{S} v_s^B x_{sj}^B} \le 1 \quad j = 1, \ldots, N \\
& \sum_{r=1}^{R} \mu_r^A y_{r0}^A = E_{AA}^* \\
& \sum_{i=1}^{I} \omega_i^A x_{ij}^A - \sum_{r=1}^{R} \mu_r^A y_{rj}^A \ge 0 \qquad j = 1, \ldots, N \\
& \sum_{i=1}^{I} \omega_i^A x_{i0}^A = 1 \\
& \mu_r^A, \omega_i^A, u_t^B, v_s^B, D \ge 0 \\
& r = 1, \ldots, R, \; t = 1, \ldots, T, \; i = 1, \ldots, I, \; s = 1, \ldots, S
\end{aligned}
\tag{15.5}
$$

Note that in model (15.5), we try to determine the buyer's efficiency given that the seller's efficiency remains at E_{AA}^*. Model (15.5) is equivalent to the following non-linear model:

$$
\begin{aligned}
& \textit{Maximize } E_{AB} = \sum_{t=1}^{T} \mu_t^B y_{t0}^B \\
& \text{subject to} \\
& d \times \sum_{r=1}^{R} \mu_r^A y_{rj}^A + \sum_{s=1}^{S} \omega_s^B x_{sj}^B - \sum_{t=1}^{T} \mu_t^B y_{tj}^B \ge 0 \quad j = 1, \ldots, N \\
& d \times \sum_{r=1}^{R} \mu_r^A y_{r0}^A + \sum_{s=1}^{S} \omega_s^B x_{s0}^B = 1 \\
& \sum_{r=1}^{R} \mu_r^A y_{r0}^A = E_{AA}^* \\
& \sum_{i=1}^{I} \omega_i^A x_{ij}^A - \sum_{r=1}^{R} \mu_r^A y_{rj}^A \ge 0 \qquad j = 1, \ldots, N \\
& \sum_{i=1}^{I} \omega_i^A x_{i0}^A = 1 \\
& \mu_r^A, \omega_i^A, \mu_t^B, \omega_s^B, d \ge 0 \\
& r = 1, \ldots, R, \; t = 1, \ldots, T, i = 1, \ldots, I, \; s = 1, \ldots, S
\end{aligned}
\tag{15.6}
$$

Note that $d \times \sum_{r=1}^{R} \mu_r^A y_{r0}^A + \sum_{s=1}^{S} \omega_s^B x_{s0}^B = 1$ and $\sum_{r=1}^{R} \mu_r^A y_{r0}^A = E_{AA}^*$.

Thus, we have $0 \le d < 1 / \sum_{r=1}^{R} \mu_r^A y_{r0}^A = 1 / E_{AA}^*$. i.e, we have the upper and lower bounds on d. Therefore, d can be treated as a parameter, and model (15.6) can be solved as a parametric linear program.

In computation, we set the initial d value as the upper bound, namely, $d_0 = 1/E^*_{AA}$, and solve the resulting linear program. We then start to decrease d according to $d_t = 1/E^*_{AA} - \varepsilon \times t$ for each step t, where ε is a small positive number[1]. We solve each linear program of model (15.6) corresponding to d_t and denote the optimal objective value as $E^*_{BA}(d_t)$.

Let $E^*_{BA} = \underset{t}{Max} E^*_{BA}(d_t)$. Then we obtain a best heuristic search solution

E^*_{BA} to model (15.6)[2]. This E^*_{AB} represents the buyer's efficiency when the seller is given the pre-emptive priority to achieve its best performance. The efficiency of the supply chain can then be defined as

$$e_{AB} = \frac{1}{2}(E^*_{AA} + E^*_{AB})$$

Similarly, one can develop a procedure for the situation when the buyer is the leader and the seller the follower. For example, in the October 6, 2003 issue of the Business Week, its cover story reports that Walmart dominates its suppliers and not only dictates delivery schedules and inventory levels, but also heavily influences product specifications.

We first evaluate the efficiency of the buyer using the standard CRS ratio model:

$$\text{Maximize } E_{BB} = \frac{\sum_{t=1}^{T} u_t^B y_{t0}^B}{\sum_{r=1}^{R} u_r^A y_{r0}^A + \sum_{s=1}^{S} v_s^B x_{s0}^B} \tag{15.7}$$

subject to

$$\frac{\sum_{t=1}^{T} u_t^B y_{tj}^B}{\sum_{r=1}^{R} u_r^A y_{rj}^A + \sum_{s=1}^{S} v_s^B x_{sj}^B} \le 1 \qquad j = 1, \ldots, N$$

$$u_r^A, u_t^B, v_s^B \ge 0$$

$$r = 1, \ldots, R, \; t = 1, \ldots, T, \; s = 1, \ldots, S$$

[1] In the current study, we set $\varepsilon = 0.01$. If we use a smaller ε, the difference only shows in the fourth decimal place in the current study.

[2] The obtained solution can be regarded as the global solution using a heuristic technique, as it searches through the entire feasible region of d when d is decreased from its upper bound to lower bound of zero. It is likely that estimation error exists. The smaller the decreased step, the better the heuristic search solution will be.

Model (15.7) is equivalent to the following standard CRS multiplier model:

$$\begin{aligned}
&\text{Maximize } E_{BB} = \sum\nolimits_{t=1}^{T} \mu_t^B y_{t0}^B \\
&\text{subject to} \\
&\sum\nolimits_{r=1}^{R} \mu_r^A y_{rj}^A + \sum\nolimits_{s=1}^{S} \omega_s^B x_{sj}^B - \sum\nolimits_{t=1}^{T} \mu_t^B y_{tj}^B \geq 0 \qquad j = 1, \ldots, N \\
&\sum\nolimits_{r=1}^{R} \mu_r^A y_{r0}^A + \sum\nolimits_{s=1}^{S} \omega_s^B x_{s0}^B = 1 \\
&\mu_r^A, \mu_t^B, \omega_s^B \geq 0 \\
&r = 1, \ldots, R,\ t = 1, \ldots, T,\ s = 1, \ldots, S
\end{aligned} \tag{15.8}$$

Let μ_r^{A*}, μ_t^{B*}, ω_s^{B*}, E_{BB}^* ($r=1,\ldots, R$, $t=1,\ldots, T$, $s=1,\ldots, S$) be an optimal solution from model (15.8), where E_{BB}^* represents the buyer's efficiency score. To obtain the seller's efficiency given that the buyer's efficiency is equal to E_{BB}^*, we solve the following model:

$$\begin{aligned}
&\text{Maximize } E_{BA} = \frac{U \times \sum\nolimits_{r=1}^{R} \mu_r^A y_{r0}^A}{\sum\nolimits_{i=1}^{I} v_i^A x_{i0}^A} \\
&\text{subject to} \\
&\frac{U \times \sum\nolimits_{r=1}^{R} \mu_r^A y_{rj}^A}{\sum\nolimits_{i=1}^{I} v_i^A x_{ij}^A} \leq 1 \qquad j = 1, \ldots, N \\
&\sum\nolimits_{t=1}^{T} \mu_t^B y_{t0}^B = E_{BB}^* \\
&\sum\nolimits_{r=1}^{R} \mu_r^A y_{rj}^A + \sum\nolimits_{s=1}^{S} \omega_s^B x_{sj}^B - \sum\nolimits_{t=1}^{T} \mu_t^B y_{tj}^B \geq 0 \qquad j = 1, \ldots, N \\
&\sum\nolimits_{r=1}^{R} \mu_r^A y_{r0}^A + \sum\nolimits_{s=1}^{S} \omega_s^B x_{s0}^B = 1 \\
&v_i^A, \mu_r^A, \mu_t^B, \omega_s^B, U \geq 0 \\
&r = 1, \ldots, R,\ t = 1, \ldots, T, i = 1, \ldots, I,\ s = 1, \ldots, S
\end{aligned} \tag{15.9}$$

Model (15.9) is equivalent to the following non-linear program:

$$\begin{aligned}
&\text{Maximize } E_{BA} = u \times \sum\nolimits_{r=1}^{R} \mu_r^A y_{r0}^A \\
&\text{subject to} \\
&\sum\nolimits_{i=1}^{I} \omega_i^A x_{ij}^A - u \times \sum\nolimits_{r=1}^{R} \mu_r^A y_{rj}^A \geq 0 \quad j = 1, \ldots, N \\
&\sum\nolimits_{i=1}^{I} \omega_i^A x_{i0}^A = 1 \\
&\sum\nolimits_{t=1}^{T} \mu_t^B y_{t0}^B = E_{BB}^*
\end{aligned} \tag{15.10}$$

$$\sum_{r=1}^{R} \mu_r^A y_{rj}^A + \sum_{s=1}^{S} \omega_s^B x_{sj}^B - \sum_{t=1}^{T} \mu_t^B y_{tj}^B \geq 0 \quad j = 1, \ldots, N$$
$$\sum_{r=1}^{R} \mu_r^A y_{r0}^A + \sum_{s=1}^{S} \omega_s^B x_{s0}^B = 1$$
$$\omega_i^A, \mu_r^A, \mu_t^B, \omega_s^B, u \geq 0$$
$$r = 1, \ldots, R, \; t = 1, \ldots, T, i = 1, \ldots, I, \; s = 1, \ldots, S$$

This model (15.10) is similar to model (15.6) and can be treated as a linear program with u as the parameter. We next show how to select the initial value of this parameter.

We first solve the following model:

$$\begin{aligned} &\text{Maximize } EF_{BA} = \frac{U \times \sum_{r=1}^{R} \mu_r^{A*} y_{r0}^A}{\sum_{i=1}^{I} v_i^A x_{i0}^A} \\ &\text{subject to} \\ &\frac{U \times \sum_{r=1}^{R} \mu_r^{A*} y_{rj}^A}{\sum_{i=1}^{I} v_i^A x_{ij}^A} \leq 1 \, j = 1, \ldots, N \\ &v_i^A, U \geq 0 \; i = 1, \ldots, I \end{aligned} \tag{15.11}$$

where μ_r^{A*} $(r=1,\ldots, R)$ is an optimal solution from model (15.8).

Model (15.11) is equivalent to the following linear program:

$$\begin{aligned} &\text{Maximize } EF_{BA} = u \times \sum_{r=1}^{R} \mu_r^{A*} y_{r0}^A \\ &\text{subject to} \\ &\sum_{i=1}^{I} \omega_i^A x_{ij}^A - u \times \sum_{r=1}^{R} \mu_r^{A*} y_{rj}^A \geq 0 \quad j = 1, \ldots, N \\ &\sum_{i=1}^{I} \omega_i^A x_{i0}^A = 1 \\ &\omega_i^A, u \geq 0 \qquad i = 1, \ldots, I \end{aligned} \tag{15.12}$$

Let $\omega_i^{A*}, u^*, EF_{BA}^*$ $(i=1,\ldots, I)$ be an optimal solution from model (15.12). Note that the optimal value to model (15.12), EF_{BA}^*, may not be the maximum value for the seller because of possible multiple optima in model (15.8). We have $u \times \sum_{r=1}^{R} \mu_r^A y_{r0}^A \geq EF_{BA}^*$. Further, based upon $\sum_{r=1}^{R} \mu_r^A y_{r0}^A + \sum_{s=1}^{S} \omega_s^B x_{s0}^B = 1$, we have $\sum_{r=1}^{R} \mu_r^A y_{r0}^A \leq 1$. Therefore, $u \geq EF_{BA}^*$. We then utilize EF_{BA}^* as the lower bound for the parameter u when solving for seller's efficiency using model (15.10). However, this lower bound can be converted into an upper bound as follows.

Table 15.3 Numerical example

DMU	x_1^A	x_2^A	y_1^A	y_2^A	x^B	y^B
1	8	50	20%	10	8	100
2	10	18	10%	15	10	70
3	15	30	10%	20	8	95
4	8	25	20%	25	10	80
5	10	40	15%	20	15	85

Let $u \times \mu_r^A = c_r^A, g = 1/u$, then model (15.10) is equivalent to the following model:

$$
\begin{aligned}
&\text{Maximize } E_{BA} = \sum\nolimits_{r=1}^{R} c_r^A y_{r0}^A \\
&\text{subject to} \\
&\sum\nolimits_{i=1}^{I} \omega_i^A x_{ij}^A - \sum\nolimits_{r=1}^{R} c_r^A y_{rj}^A \geq 0 \qquad j = 1, \ldots, N \\
&\sum\nolimits_{i=1}^{I} \omega_i^A x_{i0}^A = 1 \\
&\sum\nolimits_{t=1}^{T} \mu_t^B y_{t0}^B = E_{BB}^* \\
&\frac{1}{g} \sum\nolimits_{r=1}^{R} c_r^A y_{rj}^A + \sum\nolimits_{s=1}^{S} \omega_s^B x_{sj}^B - \sum\nolimits_{t=1}^{T} \mu_t^B y_{tj}^B \geq 0 \quad j = 1, \ldots, N \\
&\frac{1}{g} \sum\nolimits_{r=1}^{R} c_r^A y_{r0}^A + \sum\nolimits_{s=1}^{S} \omega_s^B x_{s0}^B = 1 \\
&\omega_i^A, c_r^A, \mu_t^B, \omega_s^B, g \geq 0 \\
&r = 1, \ldots, R, \; t = 1, \ldots, T, i = 1, \ldots, I, \; s = 1, \ldots, S
\end{aligned}
\tag{15.13}
$$

where $0 \leq g \leq 1/EF_{BA}^*$ can be treated as a parameter.

We solve model (15.13) for the seller's efficiency. The computational procedure is similar to the one used in model (15.6). Denote the heuristic search solution to (15.13) as E_{BA}^*. Then the efficiency of the supply chain can be defined as

$$
e_{BA} = \frac{1}{2}(E_{BA}^* + E_{BB}^*)
$$

We now illustrate the above DEA procedures with five supply chain operations (DMUs) given in Table 15.3. The seller has two inputs, x_1^A (labor) and x_2^A (cost) and two outputs, y_1^A (buyer's fill rate) and y_2^A (number of product shipped). The buyer has another input x^B (labor) and one output: y^B.

Table 15.4 reports the efficiency scores obtained from the supply chain efficiency models.

Table 15.4 Leader-follower results

DMU	Model (15.4) Seller	Model (15.6) Buyer	e_{AB}	Model (15.8) Buyer	Model (15.13) Seller	e_{BA}
1	1	1	1	1	1	1
2	0.833	0.56	0.697	0.875	0.766	0.821
3	0.667	0.95	0.808	1	0.546	0.773
4	1	0.653	0.827	0.653	1	0.827
5	0.64	0.453	0.547	0.756	0.621	0.688

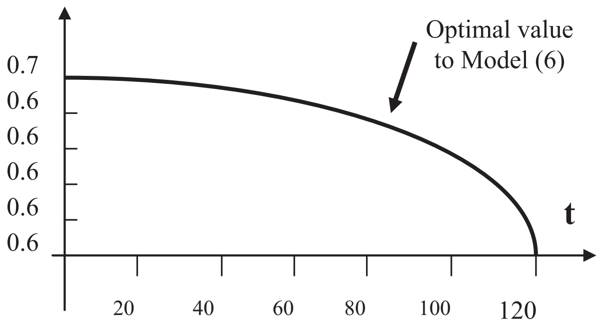


Fig. 15.5 Solving non-cooperative model for DMU2

When the seller is treated as the leader, two seller operations in DMUs 1 and 4 are efficient with only one efficient buyer operation in DMU1. This indicates that only DMU1 is the efficient supply chain.

When the buyer is treated as the leader, model (15.8) shows that three buyer operations are inefficient and model (15.13) shows that only two seller operations are efficient. This also implies that only DMU1 is efficient.

Figure 15.5 shows how the best heuristic search is obtained when solving model (15.6) for DMU2. We set $d_t = 1/E^*_{AA} - 0.01 \times t$, where $E^*_{AA} = 0.833$ and $t=0,\ldots,120$. Note that when $t=120$, the parameter $d=0$, the lower bound, and the optimal value to model (15.6) is 0.650. Therefore, we have completed the search over the entire feasible region of d and the best solution is obtained at $t=0$, that is $E^*_{AB} =$ 0.697.

15.3.2 The Cooperative Model

In game theory, when the buyer-seller relation was treated as leader-follower, the buyer does not have control over the seller, and the seller determines the optimal strategy (optimal weights for the intermediate measures). Recent studies however have demonstrated that many retailers (buyers) have increased their bargaining power relative to the manufactures' (sellers) bargaining power (Porter 1974; Li et al. 1996). The shift of power from manufacturers to retailers is one of the most significant phenomena in manufacturing and retailing. Walmart is an extreme case where the manufacturer becomes a "follower". Therefore, it is in the best interest of the supply chain to encourage cooperation. This section considers the case where both the seller and buyer have the same degree of power to influence the supply chain system. Our new DEA model seeks to maximize both the seller's and buyer's efficiency, subject to a condition that the weights on the intermediate measures must be equal:

$$
\begin{aligned}
&\text{Maximize } \frac{1}{2}\left[\frac{\sum_{r=1}^{R} c_r\, y_{r0}^A}{\sum_{i=1}^{I} v_i^A x_{i0}^A} + \frac{\sum_{t=1}^{T} u_t^B y_{t0}^B}{\sum_{r=1}^{R} c_r\, y_{r0}^A + \sum_{s=1}^{S} v_s^B x_{s0}^B}\right]\\
&\text{subject to}\\
&\frac{\sum_{r=1}^{R} c_r\, y_{rj}^A}{\sum_{i=1}^{I} v_i^A x_{ij}^A} \le 1 \qquad j = 1, \ \ldots, N\\
&\frac{\sum_{t=1}^{T} u_t^B y_{tj}^B}{\sum_{r=1}^{R} c_r\, y_{rj}^A + \sum_{s=1}^{S} v_s^B x_{sj}^B} \le 1 \quad j = 1, \ \ldots, N\\
&c_r, u_t^B, v_i^A, v_s^B \ge 0\\
&r = 1, \ \ldots, R, \ t = 1, \ \ldots, T, i = 1, \ \ldots, I, \ s = 1, \ldots, S
\end{aligned}
\tag{15.14}
$$

We call model (15.14) the cooperative efficiency evaluation model, because it maximizes the joint efficiency of the buyer and seller, and forces the two players to agree on a common set of weights on the intermediate measures[3].

We apply the following Charnes-Cooper transformation to model (15.14):

$$
\begin{aligned}
&t_1 = \frac{1}{\sum_{i=1}^{I} v_i^A x_{i0}^A} \quad t_2 = \frac{1}{\sum_{r=1}^{R} c_r^A y_{r0}^A + \sum_{s=1}^{S} v_s^B x_{s0}^B}\\
&\omega_i^A = t_1 v_i^A \quad c_r^A = t_1 c_r \quad \mu_t^B = t_2 u_t^B\\
&\omega_s^B = t_2 v_s^B \quad c_r^B = t_2 c_r\\
&r = 1, \ldots, R, t = 1, \ldots, T, i = 1, \ldots, I, s = 1, \ldots, S
\end{aligned}
$$

Note that in the above transformation, $c_r^A = t_1 c_r$ and $c_r^B = t_2 c_r$ imply a linear relationship between c_r^A and c_r^B. Therefore, we can assume $c_r^B = k \times c_r^A, \ k \ge 0$. Then model (15.14) can be changed into:

$$
\begin{aligned}
&\text{Maximize } \frac{1}{2}\left[\sum_{r=1}^{R} c_r^A y_{r0}^A + \sum_{t=1}^{T} \mu_t^B y_{t0}^B\right]\\
&\text{subject to}\\
&\sum_{i=1}^{I} \omega_i^A x_{ij}^A - \sum_{r=1}^{R} c_r^A y_{rj}^A \ge 0 \qquad j = 1, \ \ldots, N\\
&\sum_{r=1}^{R} c_r^B y_{rj}^A + \sum_{s=1}^{S} \omega_s^B x_{sj}^B - \sum_{t=1}^{T} \mu_t^B y_{tj}^B \ge 0 \quad j = 1, \ \ldots, N
\end{aligned}
\tag{15.15}
$$

[3] In cooperative game theory, the joint profit of seller and buyer is maximized.

$$\sum_{i=1}^{I} \omega_i^A x_{i0}^A = 1$$
$$\sum_{r=1}^{R} c_r^B y_{r0}^A + \sum_{s=1}^{S} \omega_s^B x_{s0}^B = 1$$
$$c_r^B = k \times c_r^A$$
$$c_r^A, c_r^B, \mu_t^B, \omega_i^A, \omega_s^B, k \geq 0$$
$$r = 1, \ldots, R, t = 1, \ldots, T, i = 1, \ldots, I, s = 1, \ldots, S$$

Model (15.15) is a non-linear programming problem, and can be converted into the following model:

$$\begin{aligned}
&\text{Maximize } \frac{1}{2}\left[\sum_{r=1}^{R} c_r^A y_{r0}^A + \sum_{t=1}^{T} \mu_t^B y_{t0}^B\right] \\
&\text{subject to} \\
&\sum_{i=1}^{I} \omega_i^A x_{ij}^A - \sum_{r=1}^{R} c_r^A y_{rj}^A \geq 0 \qquad j = 1, \ldots, N \\
&\sum_{r=1}^{R} k \times c_r^A y_{rj}^A + \sum_{s=1}^{S} \omega_s^B x_{sj}^B - \sum_{t=1}^{T} \mu_t^B y_{tj}^B \geq 0 \qquad j = 1, \ldots, N \\
&\sum_{i=1}^{I} \omega_i^A x_{i0}^A = 1 \\
&\sum_{r=1}^{R} k \times c_r^A y_{r0}^A + \sum_{s=1}^{S} \omega_s^B x_{s0}^B = 1 \\
&c_r^A, \mu_t^B, \omega_i^A, \omega_s^B, k \geq 0 \\
&r = 1, \ldots, R, \ t = 1, \ldots, T, i = 1, \ldots, I, \ s = 1, \ldots, S
\end{aligned} \tag{15.16}$$

Note that $\sum_{r=1}^{R} k \times c_r^A y_{r0}^A + \sum_{s=1}^{S} \omega_s^B x_{s0}^B = 1$, $\sum_{r=1}^{R} c_r^A y_{r0}^A \leq 1$, $\sum_{s=1}^{S} \omega_s^B x_{s0}^B > 0$ in model (15.15). We have $k = \left(1 - \sum_{s=1}^{S} \omega_s^B x_{s0}^B\right) \Big/ \sum_{r=1}^{R} c_r^A y_{r0}^A < 1 \Big/ \sum_{r=1}^{R} c_r^A y_{r0}^A$. Note also that the optimal $\sum_{r=1}^{R} c_r^{A*} y_{r0}^A$ in model (15.15) will not be less than E_{BA}^* in model (15.10). Thus, we have $0 \leq k < 1/E_{BA}^*$. That is, model (15.15) can be treated as a parametric linear program, and we can obtain a heuristic search solution using the procedure developed for models (15.6) and (15.13).

At the optima, let $\theta_A^* = \sum_{r=1}^{R} c_r^{A*} y_{r0}^A$ and $\theta_B^* = \sum_{t=1}^{T} \mu_t^{B*} y_{t0}^B$ represent the efficiency scores for the seller and buyer respectively. The following two remarks show that in general, the supply chain efficiency under the assumption of cooperation will not be less than the efficiency under the assumption of non-cooperation.

Remark 15.1: If we set $\sum_{r=1}^{R} c_r^A y_{r0}^A = E_{AA}^*$ as a constraint in model (15.15), then the feasible region of model (15.15) is the same as that of model (15.6). Therefore, $V_P^* = e_{AB}$.

Remark 15.2: If we set $\sum_{t=1}^{T} \mu_t^B y_{t0}^B = E_{BB}^*$ as a constraint in model (15.15), then the feasible region of model (15.15) is the same as that of model (15.13). Therefore, $V_P^* = e_{BA}$.

Table 15.5 Cooperative structure results

DMU	θ_A^*	θ_B^*	Supply chain
1	1	1	1
2	0.766	0.875	0.821
3	0.667	0.95	0.808
4	1	0.653	0.827
5	0.621	0.756	0.688

Table 15.6 Comparison of non-cooperative and cooperative results

DMU	e_{AB}	e_{BA}	Model (15)
1	1	1	1
2	0.697	0.821	0.821
3	0.808	0.773	0.808
4	0.827	0.827	0.827
5	0.547	0.688	0.688

We consider again the numerical example in Table 15.3. Table 15.6 reports the results from model (15.15), where columns 2 and 3 report the efficiency scores for the seller and buyer respectively, and the last column reports the optimal value to model (15.15), the supply chain efficiency (Table 15.5).

Table 15.6 compares the efficiency scores for the cooperative and non-cooperative assumptions. In this numerical example, for all DMUs, one of the two leader-follower models achieves efficiency under the cooperative assumption. This indicates that no better solution can be found to yield a higher efficiency in the cooperative assumption. However, in other examples, the supply chain is likely to show a better performance when assuming cooperative operation.

15.4 Additive Efficiency Decomposition to Network Structures

While the approaches discussed in Chap. 14 (namely, the approach of Kao and Hwang (2008), Liang, Cook and Zhu (2008), and Chen et al. (2009)) can be extended to DMUs that have more than two stages, such an extension requires that the multi-stage processes share the unique feature that all outputs from any stage represent the only inputs to the next stage. In other words, except for the first stage, all other stages do not have their own independent inputs (and/or outputs), that enter (exit) the process at that point. While these closed systems do exist, the more prevalent case is that where each stage is open, that is it has its own inputs (and/or outputs) in addition to the intermediate measures.

Such open multistage structures are relatively common, particularly in processing industries. Consider, for example, the situation in which a coal mining company wishes to evaluate the efficiency of a set of collieries (mining operations) in a large coal field. Typically, the process of delivering finished products to the customer is multistage in nature. In crude terms, Stage 1 would involve the extraction of the raw or run-of-mine (ROM) coal from underground or open pit coal reserves. At the mine site, the ROM is generally put through a process where screens separate the product into different size categories; e.g. a 'more than one inch in diameter' category, and a 'less than one inch' category. The resulting 'size grades', representing the outputs from this first stage, are then transported to an on-site washing facility, which might be deemed Stage 2. The washing process filters out any material below a certain specific gravity; this portion is unsuitable for sale and is discarded. A portion of the remaining usable coal (outputs from Stage 2) is sold to the open market as a finished product, and at management's discretion (based on estimates of the demand), the remaining product is sent to Stage 3, the crusher. The crushing process also produces waste or discard, with the remaining material, sometimes referred to as 'middlings', being sold or blended with other materials to make such products as briquettes. This latter process might be thought of as Stage 4.

Numerous such examples from processing industries exist. In many cases a portion of the outputs from one stage may be in 'finished' form and go to the consumer market, with the remainder being reprocessed at the next stage to get a more pure form of the product. The petrochemical industry, perfume manufacturing and so on, are examples.

It is important to note that the models of Kao and Hwang (2008), Liang, Cook and Zhu (2008), and Chen et al. (2009) concentrate specifically on pure serial processes. Cook et al. (2010) develop linear models for DMUs that have multiple stages, with each stage being open, having its own inputs and outputs. They also obtain an additive efficiency decomposition of the overall efficiency score. The advantage of additive efficiency decomposition is that we can also study performance under assumptions of both constant returns to scale (CRS) and variable returns to scale (VRS). They adopt a radial efficiency framework, as compared to the slacks-based framework of Tone and Tsutsui (2009). The current section presents the approach of Cook et al. (2010).

15.4.1 DEA Model for General Multi-Stage Serial Processes

Consider the P-stage process pictured in Fig. 15.6. We denote the input vector to stage 1 by z_o. The output vectors from stage p (p = 1,…, P) take two forms, namely z_p^1 and z_p^2. Here, z_p^1 represents that output that leaves the process at this stage and is not passed on as input to the next stage. The vector z_p^2 represents the amount of output that becomes input to the next (p + 1) stage. These types of intermediate measures are called *links* in Tone and Tsutsui (2009). In addition, there is the provision for new inputs z_p^3 to enter the process at the beginning of stage p + 1. Specifically, when p = 2,3,…, we define

1. z_{pr}^{j1} the rth component ($r = 1,....R_p$) of the R_p-dimensional **output** vector for DMU j flowing from stage p, that *leaves* the process at that stage, and is not passed on as an input to stage p + 1.

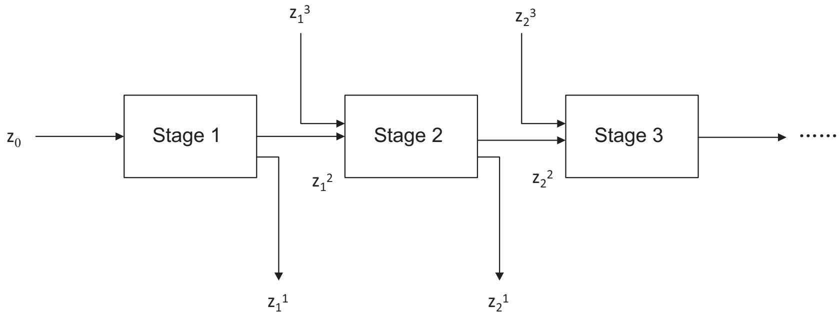


Fig. 15.6 Serial multi-stage DMU

2. z_{pk}^{j2} the kth component ($k = 1,...S_p$) of the S_p-dimensional **output** vector for DMU j flowing from stage p, and is passed on as a portion of the **inputs** to stage p+1.
3. z_{pi}^{j3} the ith component ($i = 1,...I_p$) of the I_p-dimensional **input** vector for DMU j at the stage p+1, that enters the process at the beginning of that stage.

Note that in the last stage P, all the outputs are viewed as z_{pr}^{j1}, as they leave the process.

We denote the multipliers (weights) for the above factors as

1. u_{pr} is the multiplier for the output component z_{pr}^{j1} flowing from stage p.
2. η_{pk} is the multiplier for the output component z_{pk}^{j2} at stage p, and is as well the multiplier for that same component as it becomes an input to stage p+1.
3. v_{pi} is the multiplier for the input component z_{pi}^{j3} entering the process at the beginning of stage p+1.

Thus, when p=2, 3,…, the efficiency ratio for DMU j (for a given set of multipliers) would be expressed as:

$$\theta_p = \left(\sum_{r=1}^{R_p} u_{pr} z_{pr}^{j1} + \sum_{k=1}^{S_p} \eta_{pk} z_{pk}^{j2} \right) \Bigg/ \left(\sum_{k=1}^{S_{p-1}} \eta_{p-1k} z_{p-1k}^{j2} + \sum_{i=1}^{I_p} v_{p-1i} z_{p-1i}^{j3} \right)$$

Note that there are no outputs flowing into stage 1. The efficiency measure for stage 1 of the process (namely, p=1), for DMU_j becomes

$$\theta_1 = \left(\sum_{r=1}^{R_1} u_{1r} z_{1r}^{j1} + \sum_{k=1}^{S_1} \eta_{1k} z_{1k}^{j2} \right) \Bigg/ \sum_{i=1}^{I_0} v_{0i} z_{0i}^{j}$$

where z_{0i}^{j} are the (only) inputs to the first stage represented by the input vector z_o.

We claim that the overall efficiency measure of the multistage process can reasonably be represented as a convex linear combination of the P (stage-level) measures, namely

$$\theta = \sum_{p=1}^{P} w_p \theta_p \quad \text{where} \quad \sum_{p=1}^{P} w_p = 1.$$

Note that the weights w_p are intended to represent the relative importance or contribution of the performances of individual stages p to the overall performance of the entire process. One reasonable choice for weights w_p is the proportion of total resources for the process that are devoted to stage p, and reflecting the relative size of that stage. To be more specific, $\sum_{i=1}^{I_0} \nu_{0i} z_{0i}^{j} + \sum_{p=2}^{P}\left(\sum_{k=1}^{S_{p-1}} \eta_{p-1k} z_{p-1k}^{j2} + \sum_{i=1}^{I_p} \nu_{p-1i} z_{p-1i}^{j3}\right)$ represents the total size of or total amount of resources consumed by the entire process, and we define the w_p to be the proportion of the total input used at the pth stage. We then have

$$w_1 = \sum_{i=1}^{I_0} \nu_{0i} z_{0i}^{j} \Bigg/ \left\{\sum_{i=1}^{I_0} \nu_{0i} z_{0i}^{j} + \sum_{p=2}^{P}\left(\sum_{k=1}^{S_{p-1}} \eta_{p-1k} z_{p-1k}^{j2} + \sum_{i=1}^{I_p} \nu_{p-1i} z_{p-1i}^{j3}\right)\right\},$$

$$w_p = \left(\sum_{k=1}^{S_{p-1}} \eta_{p-1k} z_{p-1k}^{j2} + \sum_{i=1}^{I_p} \nu_{p-1i} z_{p-1i}^{j3}\right) \Bigg/ \left\{\sum_{i=1}^{I_0} \nu_{0i} z_{0i}^{j} + \sum_{p=2}^{P}\left(\sum_{k=1}^{S_{p-1}} \eta_{p-1k} z_{p-1k}^{j2} + \sum_{i=1}^{I_p} \nu_{p-1i} z_{p-1i}^{j3}\right)\right\}, \ \mathrm{p} > 1$$

Thus, we can write the overall efficiency θ in the form

$$\theta = \sum_{p=1}^{P}\left(\sum_{r=1}^{R_p} u_{pr} z_{pr}^{j1} + \sum_{k=1}^{S_p} \eta_{pk} z_{pk}^{j2}\right) \Bigg/ \left\{\sum_{i=1}^{I_0} \nu_{0i} z_{0i}^{j} + \sum_{p=2}^{P}\left(\sum_{k=1}^{S_{p-1}} \eta_{p-1k} z_{p-1k}^{j2} + \sum_{i=1}^{I_p} \nu_{p-1i} z_{p-1i}^{j3}\right)\right\},$$

We then set out to optimize the overall efficiency θ of the multistage process, subject to the restrictions that the individual measures θ_p must not exceed unity, or in the linear programming format, after making the usual Charnes and Cooper transformation,

$$\begin{aligned}
&\max \sum_{p=1}^{P}\left(\sum_{r=1}^{R_p} u_{pr} z_{pr}^{o1} + \sum_{k=1}^{S_p} \eta_{pk} z_{pk}^{o2}\right)\\
&\text{subject to}\\
&\left\{\sum_{i=1}^{I_0} \nu_{0i} z_{0i}^{o} + \sum_{p=2}^{P}\left(\sum_{k=1}^{S_{p-1}} \eta_{p-1k} z_{p-1k}^{o2} + \sum_{i=1}^{I_p} \nu_{p-1i} z_{p-1i}^{o3}\right)\right\} = 1\\
&\left(\sum_{r=1}^{R_1} u_{1r} z_{1r}^{j1} + \sum_{k=1}^{S_1} \eta_{1k} z_{1k}^{j2}\right) \le \sum_{i=1}^{I_0} \nu_{0i} z_{0i}^{j}\\
&\left(\sum_{r=1}^{R_p} u_{pr} z_{pr}^{j1} + \sum_{k=1}^{S_p} \eta_{pk} z_{pk}^{j2}\right) \le \left(\sum_{k=1}^{S_{p-1}} \eta_{p-1k} z_{p-1k}^{j2} + \sum_{i=1}^{I_p} \nu_{p-1i} z_{p-1i}^{j3}\right) \forall j\\
&u_{pr}, \eta_{pk}, \nu_{pi}, \nu_{0i} \ge 0
\end{aligned} \tag{15.17}$$

Note that we should impose the restriction that the overall efficiency scores for each j should not exceed unity, but since these are redundant, this is unnecessary.

Note again that the w_p, as defined above, are variables related to the inputs and the intermediate measures. By virtue of the optimization process, it can turn out that some $w_p = 0$ at optimality. To overcome this problem, one can impose bounding restrictions $w_p \geq \beta$, where β is a selected constant. This is illustrated in the examples of Section 4.

15.4.2 General Multistage Processes

In the process discussed in the previous section it is assumed that the components of a DMU are arranged in series as depicted in Fig. 15.6. There, at each stage p, the inputs took one of two forms, namely (1) those that are outputs from the previous stage $p-1$, and (2) new inputs that enter the process at the start of stage p. On the output side, those (outputs) emanating from stage p take two forms as well, namely (1) those that leave the system as finished 'products', and (2) those that are passed on as inputs to the *immediate* next stage $p+1$.

The model presented to handle such strict serial processes is easily adapted to more general network structures. Specifically, the efficiency ratio for an overall process can be expressed as the weighted average of the efficiencies of the individual components. The efficiency of any given component is the ratio of the total output to the total input corresponding to that component. Again, the weight w_p to be applied to any component p is expressed as

$$w_p = (\text{component } p \text{ input}) / (\text{total input across all components}).$$

There is no convenient way to represent a network structure that would lend itself to a generic mathematical representation analogous to model (4) above. The sequencing of activities and the source of inputs and outputs for any given component will differ from one type of process to another. However, as a simple illustration, consider the following two examples of network structures:

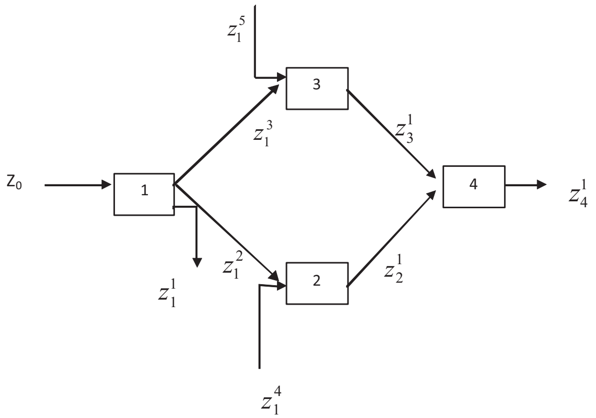
Parallel Processes Consider the process depicted in Fig. 15.7. Here, an initial input vector z_o enters component 1. Three output vectors exit this component, that is z_1^1 leaves the process, z_1^2 is passed on as an input to component 2, and z_1^3 as an input to component 3. Additional inputs z_1^4 and z_1^5 enter components 2 and 3 respectively, from outside the process. Components 2 and 3 have z_2^1 and z_3^1, respectively as output vectors which are passed on as inputs to component 4, where a final output vector z_4^1 is the result.

Component Efficiencies Component 1 efficiency ratio: $\theta_1 = \left(u_1 z_1^1 + \eta_1^2 z_1^2 + \eta_1^3 z_1^3\right) / \, v_o z_o$

Component 2 efficiency ratio: $\theta_2 = \eta_2^1 z_2^1 / \left(\eta_1^2 z_1^2 + v_1 z_1^4\right)$

Component 3 efficiency ratio: $\theta_3 = \eta_3^1 z_3^1 / \left(\eta_1^3 z_1^3 + v_2 z_1^5\right)$

Fig. 15.7 Multi-stage DMU with parallel processes



Component 4 efficiency ratio: $\theta_4 = u_4 z_4^1 / \left(\eta_2^1 z_2^1 + \eta_3^1 z_3^1\right)$

Component Weights Note that the total (weighted) input across all components is given by the sum of the denominators of θ_1 through θ_4, namely

$$I = v_o z_o + \eta_1^2 z_1^2 + v_1 z_1^4 + \eta_1^3 z_1^3 + v_2 z_1^5 + \eta_2^1 z_2^1 + \eta_3^1 z_3^1.$$

Now express the w_p as:

$$\begin{aligned} w_1 &= v_o z_o / I \\ w_2 &= (\eta_1^2 z_1^2 + v_1 z_1^4) / I \\ w_3 &= (\eta_1^3 z_1^3 + v_2 z_1^5) / I \\ w_4 &= (\eta_2^1 z_2^1 + \eta_3^1 z_3^1) / I \end{aligned}$$

With this, the overall network efficiency ratio is given by

$$\theta = \sum_{p=1}^{4} w_p \theta_p = (u_1 z_1^1 + \eta_1^2 z_1^2 + \eta_1^3 z_1^3 + \eta_2^1 z_2^1 + \eta_3^1 z_3^1 + u_4 z_4^1) / I,$$

And one then proceeds, as in (15.17) above, to derive the efficiency of each DMU and its components.

Non-Immediate Successor Flows In the previous example all flows of outputs from a stage or component either leave the process entirely or enter as an input to an *immediate successor* stage. In Fig. 15.6, stage p outputs flow to stage $p+1$. In Fig. 15.7, the same is true except that there is more than one immediate successor of stage 1.

Consider Fig. 15.8. Here, the inputs to stage 3 are of three types, namely outputs from stage 2, inputs coming from outside the process, and outputs from a previ-

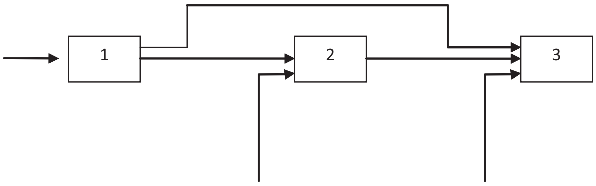


Fig. 15.8 Non-immediate successor flows

Table 15.7 Data set

	labor	Operating cost	Shipping cost	Product A	Product B	Product C	labor	Sales	Profits
DMU	z_{01}^{j}	z_{02}^{j}	z_{03}^{j}	z_{11}^{j2}	z_{12}^{j2}	z_{13}^{j2}	z_{11}^{j3}	z_{21}^{j1}	z_{22}^{j1}
1	9	50	1	20	10	5	8	100	25
2	10	18	10	10	15	7	10	70	20
3	9	30	3	8	20	2	8	96	30
4	8	25	1	20	20	10	10	80	20
5	10	40	5	15	20	5	15	85	15
6	7	35	2	35	10	5	5	90	35
7	7	30	3	10	25	8	10	100	30
8	12	40	4	20	25	4	8	120	10
9	9	25	2	10	10	5	15	110	15
10	10	50	1	20	15	9	10	80	20

ous, but not immediately previous stage. Again the above rationale for deriving weights w_p can be applied and a model equivalent to (15.17) solved to determine the decomposition of an overall efficiency score into scores for each of the components in the process.

15.4.3 An Illustrative Application

We here re-visit the supply chain data set used in Liang et al. (2006) , as shown in Table 15.7 This data set consists of a two-stage process, or a seller-buyer supply chain. The inputs to the first stage (seller) are labor (z_{01}^{j}), operating cost (z_{02}^{j}) and shipping cost (z_{03}^{j}). The outputs from the first stage are number of product A shipped (z_{11}^{j2}), number of product B shipped (z_{12}^{j2}) and number of product C shipped (z_{13}^{j2}). This data set assumes that all outputs from the first stage become inputs to the second stage, i.e., there is no z_1^1. There is one input to the second stage (buyer), labor (z_{11}^{j3}), and two outputs from the second stage, sales (z_{21}^{j1}) and profits (z_{22}^{j1}). Table 15.7 provides the data set.

In this case, we have, for DMU_o

Table 15.8 Results

DMU	Our results (Model (5))					Liang et al. (2006)	
	Overall score	w_1	w_2	θ_1	θ_2	θ_1	θ_2
1	0.92495	0.30843	0.69157	0.75666	1	1	0.89394
2	0.86486	0.51974	0.48026	0.92403	0.80082	0.92403	0.80082
3	0.85898	0.34817	0.65183	0.59497	1	0.69106	1
4	0.77381	0.5	0.5	1	0.54762	1	0.62786
5	0.62073	0.46194	0.53806	0.67595	0.57332	0.67595	0.57332
6	1	0.27992	0.72008	1	1	1	1
7	0.90405	0.5	0.5	1	0.80811	1	0.81888
8	0.92886	0.21477	0.78523	0.66875	1	0.74667	1
9	0.78091	0.43817	0.56183	0.5	1	0.5	1
10	0.75444	0.54281	0.45719	0.84226	0.65018	1	0.59596

$$w_1 = \sum_{i=1}^{3} v_{0i} z_{0i}^{o} \Bigg/ \left(\sum_{i=1}^{3} v_{0i} z_{0i}^{o} + \sum_{k=1}^{3} \eta_{1k} z_{1k}^{o2} + v_{11} z_{11}^{o3} \right),$$

$$w_2 = \left(\sum_{k=1}^{3} \eta_{1k} z_{1k}^{o2} + v_{11} z_{11}^{o3} \right) \Bigg/ \left(\sum_{i=1}^{3} v_{0i} z_{0i}^{o} + \sum_{k=1}^{3} \eta_{1k} z_{1k}^{o2} + v_{11} z_{11}^{o3} \right)$$

$$\begin{aligned}
&\text{Max} \sum_{k=1}^{3} \eta_{2k} z_{1k}^{o2} + \sum_{r=1}^{2} u_{2r} z_{2r}^{o1} \\
&\text{subject to} \\
&\sum_{i=1}^{3} v_{0i} z_{0i}^{o} + \sum_{k=1}^{3} \eta_{1k} z_{1k}^{o2} + v_{11} z_{11}^{o3} = 1 \\
&\sum_{k=1}^{3} \eta_{1k} z_{1k}^{j2} \le \sum_{i=1}^{3} v_{0i} z_{0i}^{j},\ j = 1, \ldots, 10\ (\textit{for stage } 1) \\
&\sum_{r=1}^{2} u_{2r} z_{2r}^{j1} \le \sum_{k=1}^{3} \eta_{1k} z_{1k}^{j2} + v_{11} z_{11}^{j3},\ j = 1, \ldots, 10\ (\textit{for stage } 2)
\end{aligned} \tag{15.18}$$

where efficiency scores for DMU_o in stages 1 and 2 can be expressed as

$$\theta_1 = \sum_{k=1}^{3} \eta_{1k} z_{1k}^{o2} \Bigg/ \sum_{i=1}^{3} v_{0i} z_{0i}^{o}$$

$$\theta_2 = \sum_{r=1}^{2} u_{2r} z_{2r}^{o1} \Bigg/ \left(\sum_{k=1}^{3} \eta_{1k} z_{1k}^{o2} + v_{11} z_{11}^{o3} \right)$$

Table 15.8 reports the results from model (15.18) where the last two columns display the efficiency scores derived from the cooperative model of Liang et al. (2006). Note that the differences between the two approaches are not significant. For example, the two approaches yield identical efficiency scores for the two stages for DMUs, 2, 5,

Table 15.9 Results with $\beta_1 = 0.5$, $\beta_2 = 0.5$

DMU	Overall score	w_1	w_2	θ_1	θ_2
1	0.86323	0.5	0.5	0.72645	1
2	0.85303	0.5	0.5	0.9222	0.78386
3	0.83629	0.5	0.5	0.67258	1
4	0.77381	0.5	0.5	1	0.54762
5	0.61749	0.5	0.5	0.67595	0.55903
6	0.99678	0.5	0.5	0.99357	1
7	0.90405	0.5	0.5	1	0.80811
8	0.81756	0.5	0.5	0.72772	0.9074
9	0.75	0.5	0.5	0.5	1
10	0.75435	0.5	0.5	0.85137	0.65732

6, and 9. The Liang et al.'s (2006) approach is based upon a non-linear program and its solution is obtained by using heuristic search. While the current approach uses a linear program and guarantees a global optimal solution.

Note that the average of the two stages' efficiency scores is used as the objective function in Liang et al.'s (2006) non-linear model, namely, the weights for the two individual efficiency scores are equal, $w_1 = w_2$. The current approach yields $w_1 = w_2 = 0.5$ for DMUs 4 and 7. Yet, our results are different from those obtained from Liang et al. (2006). For example, in DMU 7, the efficiency score for the second stage is 0.54762 compared to 0.81888 from Liang et al. (2006). This is due to the fact that our choice of weights actually introduces some sort of value judgment into the DEA model, and restricts the multiplier values in model (5). This is why Liang et al.'s (2006) score is larger than ours when $w_1 = w_2 = 0.5$ in optimality.

Note that weights w_p (p = 1, 2,…, P) defined in our paper are actually variables related to the multiplier decision variables. We next, therefore, impose additional restrictions on w_1 and w_2 in model (5) via

$$w_1 = \left\{ \sum_{i=1}^{3} v_{0i} z_{0i}^{o} \Bigg/ \left(\sum_{i=1}^{3} v_{0i} z_{0i}^{o} + \sum_{k=1}^{3} \eta_{1k} z_{1k}^{o2} + v_{11} z_{11}^{o3} \right) \right\} \geq \beta_1$$

$$w_2 = \left\{ \left(\sum_{k=1}^{3} \eta_{1k} z_{1k}^{o2} + v_{11} z_{11}^{o3} \right) \Bigg/ \left(\sum_{i=1}^{3} v_{0i} z_{0i}^{o} + \sum_{k=1}^{3} \eta_{1k} z_{1k}^{o2} + v_{11} z_{11}^{o3} \right) \right\} \geq \beta_2$$

where β_1 and β_2 are user-specified parameters. In this way, we can perform sensitivity analysis on w_1 and w_2.

We first impose $\beta_1 = \beta_2$ and change β_1 and β_2 0.1 to 0.5 with a 0.1 increment each time. Note that when $\beta_1 = \beta_2 = 0.5$, we explicitly require that $w_1 = w_2 = 0.5$ as in Liang et al. (2006). Table 15.9 reports the results when $\beta_1 = \beta_2 = 0.5$. Both our approach and Liang et al.'s (2006) yield identical efficiency scores for DMU9. Except for DMU1, Liang et al.'s (2006) score is larger than ours when $w_1 = w_2 = 0.5$ in optimality. For DMU1, the definition of our weights and restrictions on our weights

Table 15.10 Results with $\beta_1 = \beta_2 = 0.1$ (0.2, 0.3, 0.4)

DMU	Overall score	w_1	w_2	θ_1	θ_2
2	0.86486	0.51974	0.48026	0.92403	0.80082
4	0.77381	0.5	0.5	1	0.54762
5	0.62073	0.46194	0.53806	0.67595	0.57332
6	1	0.31591	0.68409	1	1
7	0.90405	0.5	0.5	1	0.80811
9	0.78091	0.43817	0.56183	0.5	1
10	0.75444	0.54281	0.45719	0.84226	0.65018

Table 15.11 Results for DMUs 1, 3, and 8

DMU	Overall score	w_1	w_2	θ_1	θ_2	
1	0.92495	0.30843	0.69157	0.75666	1	$\beta_1 = \beta_2 = 0.1, 0.2, 0.3$
1	0.90182	0.4	0.6	0.75455	1	$\beta_1 = \beta_2 = 0.4$
3	0.85898	0.34817	0.65183	0.59497	1	$\beta_1 = \beta_2 = 0.1, 0.2, 0.3$
3	0.85186	0.4	0.6	0.62966	1	$\beta_1 = \beta_2 = 0.4$
8	0.92886	0.21477	0.78523	0.66875	1	$\beta_1 = \beta_2 = 0.1, 0.2$
8	0.91627	0.3	0.7	0.72091	1	$\beta_1 = \beta_2 = 0.3$
8	0.89238	0.4	0.6	0.73095	1	$\beta_1 = \beta_2 = 0.4$

turn the efficient stage 1 under Liang et al.'s (2006) approach into an inefficient stage, and the inefficient stage 2 under Liang et al.'s (2006) approach into efficient.

Table 15.10 reports the results for DMUs 2, 4, 5, 6, 7, 9 and 10 whose efficiency scores along with the optimized weights remain unchanged when $\beta_1 = \beta_2 = 0.1$, 0.2, 0.3 and 0.4, respectively.

Table 15.11 reports the results for DMUs 1, 3 and 8 whose efficiency scores changed when β_1 and β_2 are changed (see the last column of Table 15.11). For DMUs 1 and 3, change in the efficiency scores does not occur until $\beta_1 = \beta_2 = 0.4$. For DMU 8, a change in the efficiency score for the first stage is observed when $\beta_1 = \beta_2 = 0.3$ and 0.4.

It can be seen that up to $\beta_1 = \beta_2 = 0.3$, most of the DMUs have the same weights and efficiency scores with respect to different values of β_1 and β_2. As expected, when $\beta_1 = \beta_2 = 0.4$, some of the resulting weights are different from the previous cases. However, we note that the efficiency scores do not change significantly. We also note that the efficiency scores for the second stage do not change when β_1 and β_2 are increased from 0.1 to 0.4.

We also performed calculations when β_1 is fixed at 0.2 and β_2 is changed from 0.3 to 0.8 with an increment of 0.1 each time (results are not reported here). In overall, the efficiency scores do not change significantly.

The above sensitivity analysis indicates that efficiency scores obtained based upon our approach are robust with respect to our choice of weights.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_15) contains supplementary material, which is available to authorized users.

References

Chen, Y., & Zhu, J. (2004). Measuring information technology's indirect impact on firm performance. *Information Technology & Management Journal, 5*(1–2), 9–22.

Chen, Y., & Cook, W. D., Li, N., & Zhu, J. (2009). Additive efficiency decomposition in two-stage DEA. *European Journal of Operational Research, 196,* 1170–1176.

Cheung, K. L., & Hansman, W. H. (2000). An exact performance evaluation for the supplier in a two-echelon inventory system. *Operations Research, 48,* 646–653.

Cook, W. D., Zhu, J., Yang, F., & Bi, G.-B. (2010). Network DEA: Additive efficiency decomposition. *European Journal of Operational Research, 207*(2), 1122–1129.

Färe, R., & Grosskopf, S. (2000). Network DEA. *Socio-Economic Planning Sciences, 34,* 35–49.

Golany, B., Hackman, S. T., & Passy, U. (2006). An efficiency measurement framework for multi-stage production systems. *Annals of Operations Research, 145*(1), 51–68.

Huang, Z. (2000). Franchising cooperation through chance cross-constrained games. *Naval Research Logistics, 47,* 669–685.

Kao, C., Hwang, S. -N. (2008). Efficiency decomposition in two-stage data envelopment analysis: An application to non-life insurance companies in Taiwan. *European Journal of Operational Research, 185*(1), 418–429.

Lee, H. L., & Billington, C. (1992). Managing supply chain inventory: pitfalls and opportunities. *Sloan Management Review, 33*(3), 65–73.

Li, S. X., Huang, Z., & Ashley, A. (1995). Seller-buyer system co-operation in a monopolistic market. *Journal of the Operational Research Society, 46,* 1456–1470.

Li, S. X., Huang, Z., & Ashley, A. (1996). Improving buyer-seller system cooperation through inventory control. *International Journal of Production Economics, 43,* 37–46.

Liang, L., Yang, F., Cook, W. D., & Zhu, J. (2006). DEA models for supply chain efficiency evaluation. *Annals of Operations Research, 145*(1), 35–49.

Liang, L., Cook, W. D., & Zhu, J. (2008). DEA models for two-stage processes: Game approach and efficiency decomposition. *Naval Research Logistics, 55,* 643–653.

Parlar, M., & Weng, Z. K. (1997). Designing a firm's coordinated manufacturing and supply decisions with short product life cycles. *Management Science, 43*(10), 1329–1344.

Porter, M. E. (1974). Consumer behavior, retailer power and market performance in consumer goods industries. *Review of Economics and Statistics, LVI,* 419–436.

Seiford, L. M., & Zhu, J. (1999a). Profitability and marketability of the top 55 US commercial banks. *Management Science, 45*(9), 1270–1288.

Seiford, L. M., & Zhu, J. (1999b). Sensitivity and stability of the classification of returns to scale in data envelopment analysis. *Journal of Productivity Analysis, 12*(1), 55–75.

Simaan, M., & Cruz, J. B. (1973). On the Stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and Applications, 11,* 533–555.

Thomas, D. J., & Griffin, P. M. (1996). Coordinated supply chain management. *European Journal of Operational Research, 94*(1), 1–15.

Tone, K., Tsutsui, M. (2009). Network DEA: A slacks-based measure approach. *European Journal of Operational Research, 197*(1), 243–252.

Chapter 16
Congestion

16.1 Congestion Measure

Congestion, as used in economics, refers to situations where reductions in one or more inputs generate an increase in one or more outputs. Examples can be found in underground mining and agriculture. For example, too much fertilizer applied to a given plot could reduce the overall output. We here adopt the following definition of congestion from Cooper et al. (1996).

Definition 16.1 (Congestion) Evidence of congestion is present when *reductions* in one or more inputs can be associated with *increases* in one or more outputs—or, proceeding in reverse, when *increases* in one or more inputs can be associated with *decreases* in one or more outputs—without worsening any other input or output.

Färe and Grosskopf (1983) apply this concept to DEA using strong and weak input disposabilities. The envelopment DEA models discussed in Chap. 1 are strong input/output disposability models. We re-write the VRS envelopment models as

$$
\begin{aligned}
&\theta^* = \min \theta \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le \theta x_{io} \qquad i = 1, 2, \ldots, m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} \qquad r = 1, 2, \ldots, s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \ge 0 \qquad j=1, \ldots, n.
\end{aligned}
\tag{16.1}
$$

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_16,

$$
\begin{aligned}
&\phi^* = \max \phi \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io} i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge \phi y_{ro} \ r = 1,2,\ldots,s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \ge 0 \ j = 1,\ldots,n.
\end{aligned} \tag{16.2}
$$

If we assume weak disposability of inputs and outputs in models (16.1) and (16.2), respectively, we obtain

$$
\begin{aligned}
&\tilde{\theta}^* = \min \tilde{\theta} \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} = \tilde{\theta} x_{io} \ i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} \ r = 1,2,\ldots,s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \ge 0 \ j = 1,\ldots,n.
\end{aligned} \tag{16.3}
$$

$$
\begin{aligned}
&\tilde{\phi}^* = \max \tilde{\phi} \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} = \tilde{\phi} y_{ro} \ r = 1,2,\ldots,s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \ge 0 \qquad\qquad j = 1,\ldots,n.
\end{aligned} \tag{16.4}
$$

Note that, for example, the difference between models (16.1) and (16.3) is that input inequalities are changed into input equalities. If we apply weak disposability to other envelopment models, we obtain the weak disposability DEA models shown in Table 16.1.

Table 16.1 Weak disposability DEA models

Frontier type	Weak input disposability	Weak output disposability
CRS	$\min \tilde{\theta}$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} = \tilde{\theta} x_{io}$ $i = 1,2,\ldots,m;$ $\sum_{j=1}^{n} \lambda_j y_{rj} \geq y_{ro}$ $r = 1,2,\ldots,s;$ $\lambda_j \geq 0$ $j = 1,2,\ldots,n.$	$\max \tilde{\phi}$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} \leq x_{io}$ $i = 1,2,\ldots,m;$ $\sum_{j=1}^{n} \lambda_j y_{rj} = \tilde{\phi} y_{ro}$ $r = 1,2,\ldots,s;$ $\lambda_j \geq 0$ $j = 1,2,\ldots,n.$
VRS	Add $\sum_{j=1}^{n} \lambda_j = 1$	
NIRS	Add $\sum_{j=1}^{n} \lambda_j \leq 1$	
NDRS	Add $\sum_{j=1}^{n} \lambda_j \geq 1$	

The input and output congestion measures are then defined as $C(\theta^*, \tilde{\theta}^*) = \theta^* / \tilde{\theta}^*$, and $C(\phi^*, \tilde{\phi}^*) = \phi^* / \tilde{\phi}^*$, respectively. Note that we must have $\theta^* \leq \tilde{\theta}^*$ because the latter is associated with equalities. As shown by Färe et al. (1994), we can use $C(\theta^*, \tilde{\theta}^*)$ (or $C(\phi^*, \tilde{\phi}^*)$) as a measure of congestion with the following properties. If $C(\theta^*, \tilde{\theta}^*) = 1$ ($C(\phi^*, \tilde{\phi}^*) = 1$), then input (output) is not congested; alternatively, if $C(\theta^*, \tilde{\theta}^*) < 1$ ($C(\phi^*, \tilde{\phi}^*) > 1$), then input (output) congestion is present.

Byrnes et al. (1984) study the congestion of 15 Illinois coal mines. Figure 16.1 presents the 15 mines with one output (thousands tons) and five inputs, namely, labor (thousand miner-days), dragline capacity (K1) (cubic yards), power-shovel capacity (K2) (cubic yards), thickness of first-seam mined (T1) (feet), and reciprocal of depth to first-seam mined (1/D1) (D1 in feet).

In Fig. 16.1, cells B20:B26 contain the formulas

Cell B20 = SUMPRODUCT(B2:B16,$I$2:$I$16)
Cell B21 = SUMPRODUCT(C2:C16,$I$2:$I$16)
Cell B22 = SUMPRODUCT(D2:D16,$I$2:$I$16)
Cell B23 = SUMPRODUCT(E2:E16,$I$2:$I$16)
Cell B24 = SUMPRODUCT(F2:F16,$I$2:$I$16)
Cell B25 = SUMPRODUCT(H2:H16,$I$2:$I$16)
Cell B26 = SUM(I2:I16)

where cells I2:I16 represent the changing cells, λ_j (j = 1, …, 15).

Cells D20:D25 contain the formulas

D20 = $F$19*INDEX($B$2:$F$16,$E$18,1)
D21 = $F$19*INDEX($B$2:$F$16,$E$18,2)
D22 = $F$19*INDEX($B$2:$F$16,$E$18,3)

	A	B	C	D	E	F	G	H	I	J
1	Mines	Labor	K1	K2	T1	1/D1		Output	λ	Weak
2	1	98.5	142	245	6	0.016		3264	0	1
3	2	96.5	30	215	6	0.016		3065	0	1
4	3	57.6	18	105	5.6	0.026		2275	0	1
5	4	59.2	160	0	5.9	0.025		1978	0	1
6	5	57.6	200	0	8	0.022		1833	0	1
7	6	49.9	27	85	4.5	0.019		1218	0	1
8	7	53.5	143	65	6	0.01		928	0	1
9	8	34	70	65	6	0.02		919	0	1
10	9	39.6	67.5	40	6.5	0.013		777	0	1
11	10	51.3	0	145	3.2	0.019		745	0	1
12	11	74.2	110	65	2.1	0.014		742	0	1
13	12	24	25	65	4.4	0.012		488	0	1
14	13	26.5	58	0	3	0.014		407	0	1
15	14	43.1	70	0	6.5	0.012		402	0	1
16	15	20.7	236	0	5.7	0.01		396	1	1
17										
18		Reference		DMU under	**15**	Efficiency				
19	**Constraints**	set		Evaluation		1				
20	Labor	20.7	=	20.7						
21	K1	236	=	236						
22	K2	0	=	0		Weak				
23	T1	5.7	=	5.7						
24	1/D1	0.01	=	0.01						
25	output	396	≥	396						
26	Σλ	1	=	1						

Fig. 16.1 VRS weak input disposability spreadsheet model

D23 = $F$19*INDEX($B$2:$F$16,$E$18,4)
D24 = $F$19*INDEX($B$2:$F$16,$E$18,5)
D25 = INDEX(H2:H16,$E$18,1)

where F19 is the target cell $(\tilde{\theta})$, and cell E18 indicates the DMU under evaluation.

Figure 16.2 shows the Solver parameters for model (16.3) shown in Fig. 16.1.

The optimal value to model (16.3) in this case is equal to one across all DMUs. i.e., each mine is on the frontier. To obtain the congestion measure, we also need to calculate model (16.1). Figure 16.3 shows the results. Figure 16.3 is the input-oriented VRS envelopment model where the inputs are strongly disposable. The related Solver parameters can be obtained by changing the equalities to inequalities in Fig. 16.2, as shown in Fig. 16.4. The efficiency scores are reported in cells J2:J16. The efficiency scores for weak input disposability are reported in cells K2:K16. It can be seen that congestion is present at DMUs 6 and 8.

When input congestion is present, we need to identify sources and amounts of congestion. Färe et al. (1994) suggest a procedure for identifying input measure responsible for the input congestion.

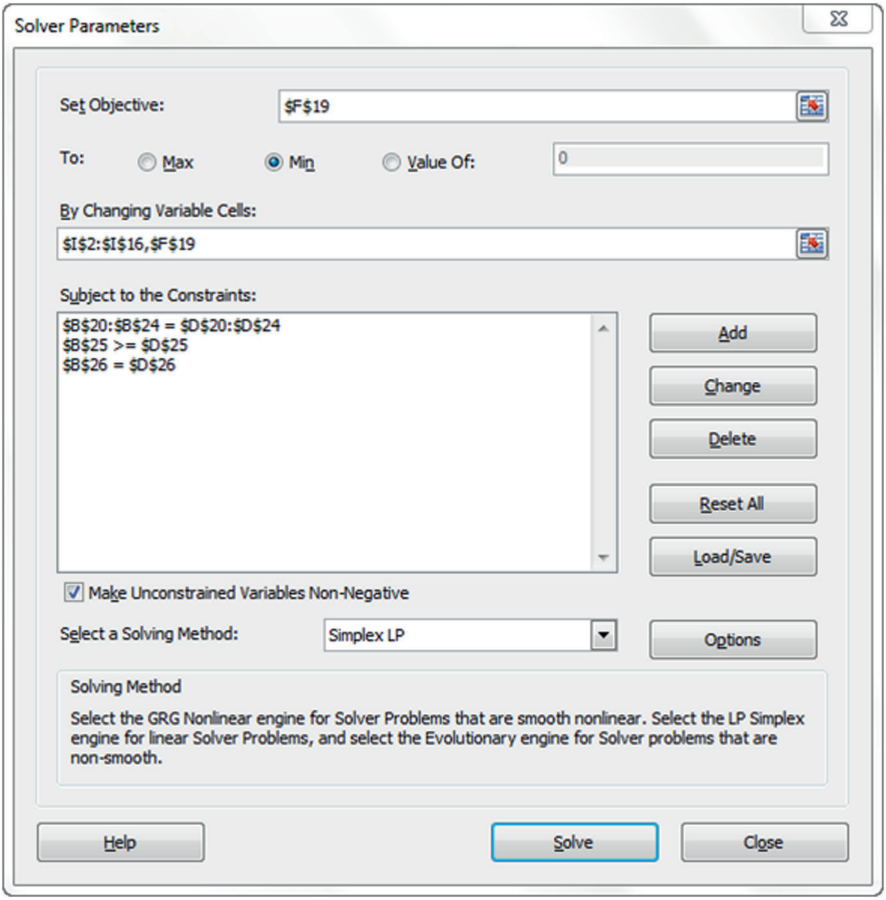


Fig. 16.2 Solver parameters for VRS weak input disposability model

$$
\begin{aligned}
&\alpha^* = \min \alpha \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} = \alpha x_{io} \qquad i \in \mathrm{A}; \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le \alpha x_{io} \qquad i \in \overline{\mathrm{A}}; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} \qquad r = 1,\ldots,s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \ge 0 \qquad\qquad j=1,\ldots,n.
\end{aligned}
\tag{16.5}
$$

	A	B	C	D	E	F	G	H	I	J	K
1	Mines	Labor	K1	K2	T1	1/D1		Output	λ	Strong	Weak
2	1	98.5	142	245	6	0.016		3264	0	1	1
3	2	96.5	30	215	6	0.016		3065	0	1	1
4	3	57.6	18	105	5.6	0.026		2275	0	1	1
5	4	59.2	160	0	5.9	0.025		1978	0	1	1
6	5	57.6	200	0	8	0.022		1833	0	1	1
7	6	49.9	27	85	4.5	0.019		1218	0	0.97129	1
8	7	53.5	143	65	6	0.01		928	0	1	1
9	8	34	70	65	6	0.02		919	0	0.93747	1
10	9	39.6	67.5	40	6.5	0.013		777	0	1	1
11	10	51.3	0	145	3.2	0.019		745	0	1	1
12	11	74.2	110	65	2.1	0.014		742	0	1	1
13	12	24	25	65	4.4	0.012		488	0	1	1
14	13	26.5	58	0	3	0.014		407	0	1	1
15	14	43.1	70	0	6.5	0.012		402	0	1	1
16	15	20.7	236	0	5.7	0.01		396	1	1	1
17											
18		Reference		DMU under	15	Efficiency					
19	Constr	set		Evaluation		1					
20	Labor	20.7	≤	20.7							
21	K1	236	≤	236							
22	K2	0	≤	0		Strong					
23	T1	5.7	≤	5.7							
24	1/D1	0.01	≤	0.01							
25	output	396	≥	396							
26	Σλ	1	=	1							

Fig. 16.3 Congestion measure For 15 mines

where $A \subseteq \{1, 2, \cdots, m\}$ and $\overline{A}$ is the complement. Using $\tilde{\theta}^*$ and α^* for each $A \subseteq \{1, 2, \ldots, m\}$, if $C(\theta^*, \tilde{\theta}^*) < 1$, and $\theta^* = \alpha^*$, as obtained from (16.1) and (16.5), the components of the subvectors associated with $\overline{A}(= \{i \mid i \notin A\})$ then identify sources and amounts of congestion. Similar models can be established for different RTS frontier and orientation assumptions. For example, if we remove $\sum \lambda_j = 1$, we obtain the model under VRS.

The suggested route requires additional computation which can be onerous because it involves obtaining solutions over all possible partitions of A. In fact, the route followed by Färe et al. (1994) emphasizes efficiency measurements with identification of sources and amounts of inefficiencies to be undertaken as an additional job.

16.2 Congestion and Slacks

We first provide the following definition.

Definition 16.2 (DEA Slacks) An optimal value of s_i^- and s_r^+ in (2.3) (or (2.6), which we represent by s_i^{-*} and s_r^{+*}, are respectively called DEA input and output slack values. i.e., we refer to the slacks obtained in the second stage of DEA calculation as DEA slacks.

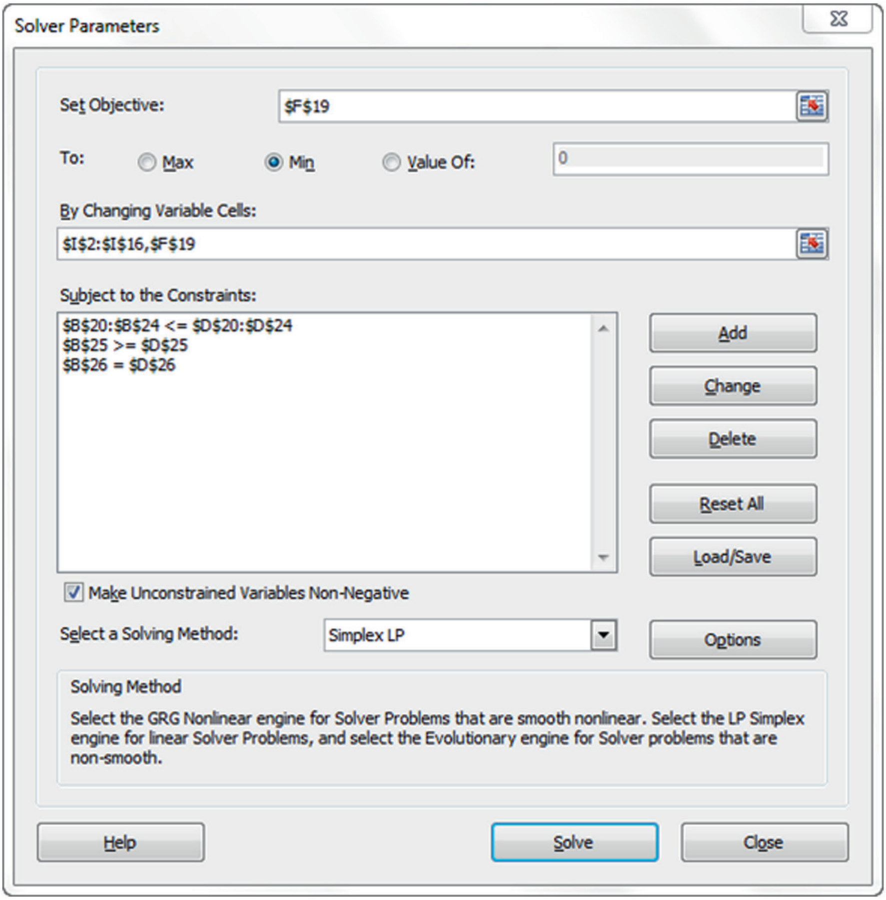


Fig. 16.4 Solver parameters for input-oriented VRS strong input disposability model

To illustrate the nature of congestion. Figures 16.5 and 16.6 plot an input isoquant. The input isoquant bends at point A in Fig. 16.5 because of the weak input disposability. As a result, AC is part the frontier, and for C, the optimal value to model (16.3) is one. However, for C, the optimal value to model (16.1) is less than one. Thus, input congestion is presented at C in Fig. 16.5.

In Fig. 16.6, the isoquant bends at point D. Because of the existence of D, the optimal values to models (16.1) and (16.3) are equal. Thus, input congestion is absent at C in Fig. 16.6.

Furthermore, note that if the efficient reference set consists of A, point C will have a positive DEA slack value for the second input x2. Because of the presence of the weakly efficient point D (a frontier point with non-zero DEA slacks), if the efficient reference set consists of points A and D, point C will not have slack values. (The (input) slacks do not necessarily represent DEA slack values.)

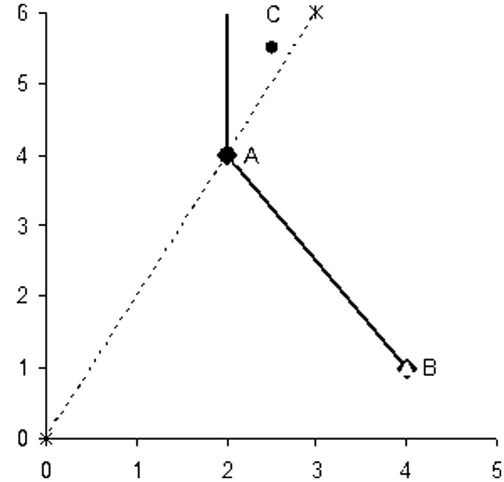


Fig. 16.5 Congestion at point C

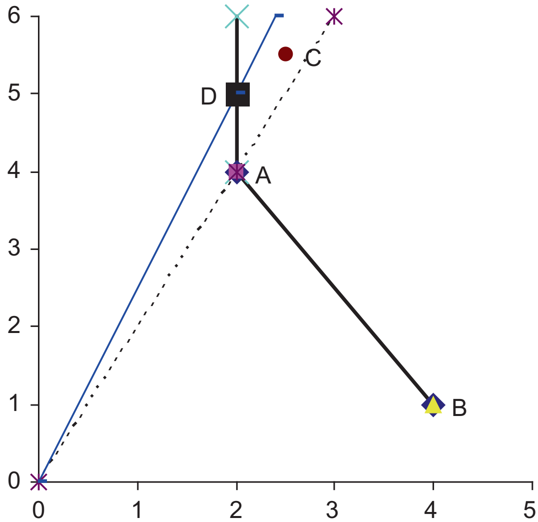


Fig. 16.6 No congestion at point C

However, if all frontier DMUs are extreme efficient, e.g., A and B, in Fig. 16.5, then the input slacks are the same as the DEA slack values. In Fig. 16.6, because C can be compared to a convex combination of D (weakly efficient) and A, no input slack is detected.

Theorem 16.1 Input congestion as defined by $C(\theta^*, \tilde{\theta}^*)$ is not present in the performance of DMU_o *if and only if* an optimal solution is associated with referent frontier DMUs such that non-zero input slack values are *not* detected in model (16.1).

[Proof]: Recall that the only difference between (16.1) and (16.3) is that the input inequalities are changed to equalities. The referent frontier DMUs are those in the basis when calculating the strong disposability model (16.1). If we have some referent DMUs such that no non-zero input slack values are detected for DMU_o, then we have, at optimality,

$$\sum_{j \in B} \lambda_j^* x_{ij} = \theta^* x_{io} \quad \text{for} \quad i = 1, \ldots, m$$

where B represents the set of referent DMUs, $B = \{j \mid \lambda_j^* > 0\}$. Obviously, λ_j^* and θ^* are also optimal for (16.3). Therefore $\theta^* = \beta^*$. Thus, no input congestion occurs. This completes the *if* part.

To establish the *only if* part, we note that if no input congestion is identified when an optimum is associated with a basis *B'* such that $\sum_{j \in B'} \lambda_j^* x_{ij} = \beta^* x_{io} = \theta^* x_{io}$, then this same optimum provides referent DMUs such that the input constraints are binding in (16.1). Therefore no non-zero input slack values are detected by reference to those DMUs in *B'*.

It is well know that in the single input and the single output situation, no input or output slack will occur for CRS envelopment models, whereas non-zero slack values may occur for VRS models. That is to say, in the single input and the single output situation, congestion will never occur with CRS but can possibly happen with VRS.

Based upon Theorem 16.1, we have

Corollary 16.1 If the observed values on the efficient frontier are composed only of extreme efficient DMUs, then congestion can occur *if and only if* non-zero DEA slack values are detected. Furthermore, the sources of congestion can then only be found in these non-zero DEA slack values.

Corollary 16.1 can be important in real world applications, since the frontiers in most real world data sets contain only the extreme efficient DMUs. Consequently, the congestion and its amount can simply be represented by the DEA slacks (see Ray et al. 1998).

The discussion here is based upon the VRS envelopment model and input congestion measure. The discussion for output congestion measures is the same.

16.3 Slack-Based Congestion Measure

The previous section indicates that there is a strong relationship between (input) slacks and the measure of (input) congestion. In fact, Brockett et al. (1998) develop a new slack-based approach to capture input congestion and identify its sources and amounts. Cooper et al. (2000) study the relationship between these two DEA congestion approaches, and show that the work of Brockett et al. (1998) improves upon the work of Färe et al. (1994) in that it not only (i) detects congestion but also (ii) determines the amount of congestion, and simultaneously, (iii) identifies factors responsible for congestion and distinguishes congestion amounts from other components of inefficiency.

The following model is employed by Brockett et al. (1998) after solving the input-oriented VRS envelopment model

$$
\begin{aligned}
& max \sum_{i=1}^{m} \delta_i^+ \\
& \text{subject to} \\
& \sum_{j=1}^{n} \lambda_j x_{ij} - \delta_i^+ = \theta^* x_{io} - s_i^{-*} = \hat{x}_{io} \qquad i = 1, 2, \ldots, m; \\
& \sum_{j=1}^{n} \lambda_j y_{rj} = y_{ro} + s_r^{+*} = \hat{y}_{ro} \qquad r = 1, 2, \ldots, s; \\
& \sum_{j=1}^{n} \lambda_j = 1 \\
& \lambda_j \ge 0, \quad s_i^{-*} \ge \delta_i^+
\end{aligned} \tag{16.6}
$$

where θ^* is obtained from (16.1) while s_i^{-*} and s_r^{+*} are obtained from (2.3). The amount of congestion in each input can then be determined by the difference between each pair of s_i^{-*} and δ_i^{+*}, where δ_i^{+*} are optimal values in (16.6). That is,

$$
s_i^c = s_i^{-*} - \delta_i^{+*}, i = 1, 2, \ldots, m \tag{16.7}
$$

Definition 16.3 (Congestion Slacks) s_i^c defined in (16.7) are called input congestion slacks.

Similarly, we can calculate the output congestion slacks by

$$
\begin{aligned}
& max \sum_{r=1}^{s} \delta_r^- \\
& \text{subject to} \\
& \sum_{j=1}^{n} \lambda_j x_{ij} = x_{io} - s_i^{-*} = \hat{x}_{io} \qquad i = 1, 2, \ldots, m; \\
& \sum_{j=1}^{n} \lambda_j y_{rj} + \delta_r^- = \phi^* y_{ro} + s_r^{+*} = \hat{y}_{ro} \quad r = 1, 2, \ldots, s; \\
& \sum_{j=1}^{n} \lambda_j = 1 \\
& \lambda_j \ge 0, s_r^{+*} \ge \delta_r^-
\end{aligned}
$$

where ϕ^* is obtained from (16.2) while s_i^{-*} and s_r^{+*} are obtained from (2.6).

To establish the relationship between model (16.6) and $C(\theta^*, \tilde{\theta}^*)$, we proceed as follows. Let $x\left(s_i^c\right)$ be an input subvector in which its ith component corresponds to $s_i^c \neq 0$, i.e., $x\left(s_i^c\right)$ is a congesting subvector. Next, let $\mathbf{X}^{\mathbf{C}}$ be the set of all congesting subvectors obtained via (16.5). We have

Theorem 16.2 $x\left(s_i^c\right) \in \mathbf{X}^{\mathbf{C}}$. Furthermore, if (16.6) yields a unique optimal solution, then $\mathbf{X}^{\mathbf{C}} = \left\{x\left(s_i^c\right)\right\}$.

[Proof]: Let $A = \left\{i \mid s_i^c = 0\right\}$ and $\bar{A} = \left\{i \mid s_i^c \neq 0\right\}$. Then the constraints of (16.6) become

$$
\begin{aligned}
&\sum_{j=1}^{n} \lambda_j x_{ij} = \theta^* x_{io} \quad i \in A; \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \leq \theta^* x_{io} \quad i \in \bar{A}; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} = y_{ro} + s_r^{+*} \qquad r = 1, 2, \ldots, s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \geq 0, \ s_i^{-*} \geq \delta_i.
\end{aligned}
\tag{16.8}
$$

where θ^* is the optimal value to (16.1). This implies that θ^* is a feasible solution to (16.5). Thus, $\alpha^* \leq \theta^*$, where α^* is the optimal value to (16.5) associated with A and $\bar{A}$. On the other hand, any optimal solution to (16.5) is a feasible solution (1), therefore $\alpha^* \geq \theta^*$. Thus, $\theta^* = \alpha^*$ indicating that the input subvector associated with $\bar{A}$, $x\left(s_i^c\right)$, is a source of congestion. Therefore, $x\left(s_i^c\right) \in \mathbf{X}^{\mathbf{C}}$.

Moreover, if (16.6) yields a unique optimal solution, then the solution in (16.8) is also unique. This means that $\theta^* = \alpha^*$ does not hold for other input subvectors. Thus, $\mathbf{X}^{\mathbf{C}} = \left\{x\left(s_i^c\right)\right\}$.

Theorem 16.2 indicates that under the condition of uniqueness, congestion will occur in the Brockett et al. (1998) approach if and only if it appears in Färe et al. (1994) approach. However, the Brockett et al. (1998) approach identifies technical or mix inefficiencies and distinguishes these from congestion components via (16.7).

We observe that the use of (16.5) may result in different congestion factors because of possible multiple optimal solutions. Theorem 16.2 indicates that the results from (16.6) then yield one of the congesting subvectors obtained from (16.5). As a result, the procedure by Färe et al. (1994) for detecting the factors responsible for the congestion may be replaced by model (16.6) and one can more easily find and identify congestion and its sources without having to conduct a series of solutions as required for (16.5).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Mines	Labor	K1	K2	T1	1/D1		Output	λ	Strong		Labor	K1	K2	T1	1/D1	Output
2	1	98.5	142	245	6	0.016		3264	0	1		0.000	0.000	0.000	0.000	0.000	0.000
3	2	96.5	30	215	6	0.016		3065	0	1		0.000	0.000	0.000	0.000	0.000	0.000
4	3	57.6	18	105	5.6	0.026		2275	0	1		0.000	0.000	0.000	0.000	0.000	0.000
5	4	59.2	160	0	5.9	0.025		1978	0	1		0.000	0.000	0.000	0.000	0.000	0.000
6	5	57.6	200	0	8	0.022		1833	1E-15	1		0.000	0.000	0.000	0.000	0.000	0.000
7	6	49.9	27	85	4.5	0.019		1218	0	0.97129		5.384	0.000	0.000	0.000	0.000	0.000
8	7	53.5	143	65	6	0.01		928	0	1		0.000	0.000	0.000	0.000	0.000	0.000
9	8	34	70	65	6	0.02		919	0	0.93747		0.000	0.000	0.000	0.693	0.004	0.000
10	9	39.6	67.5	40	6.5	0.013		777	0	1		0.000	0.000	0.000	0.000	0.000	0.000
11	10	51.3	0	145	3.2	0.019		745	0	1		0.000	0.000	0.000	0.000	0.000	0.000
12	11	74.2	110	65	2.1	0.014		742	0	1		0.000	0.000	0.000	0.000	0.000	0.000
13	12	24	25	65	4.4	0.012		488	0	1		0.000	0.000	0.000	0.000	0.000	0.000
14	13	26.5	58	0	3	0.014		407	0	1		0.000	0.000	0.000	0.000	0.000	0.000
15	14	43.1	70	0	6.5	0.012		402	0	1		0.000	0.000	0.000	0.000	0.000	0.000
16	15	20.7	236	0	5.7	0.01		396	1	1		0.000	0.000	0.000	0.000	0.000	0.000
17																	
18		Reference		DMU under	15	Sum of Slacks											
19	Constraints	set		Evaluation		0.0000		DEA Slack									
20	Labor	20.7	=	20.7		0.0000											
21	K1	236	=	236		0.0000											
22	K2	0	=	0		0.0000											
23	T1	5.7	=	5.7		0.0000											
24	1/D1	0.01	=	0.01		0.0000											
25	output	396	=	396		0.0000											
26	Σλ	1	=	1													

Fig. 16.7 DEA slacks for 15 mines

Consider the mine example again. Before we solve model (16.6), we need to determine the DEA slacks for the spreadsheet shown in Fig. 16.3. i.e., we need to perform the second stage calculation for the input-oriented VRS envelopment model.

Figure 16.7 shows the spreadsheet for calculating the DEA slacks. Cells F20:F25 and F26 represent the input slacks and output slack, respectively. Cell F19 represent the sum of slacks and is the target cell in the Solver parameters shown in Fig. 16.8.

The formulas for cells B20:B25 are

Cell B20 = SUMPRODUCT(B2:B16,$I$2:$I$16)+F20
Cell B21 = SUMPRODUCT(C2:C16,$I$2:$I$16)+F21
Cell B22 = SUMPRODUCT(D2:D16,$I$2:$I$16)+F22
Cell B23 = SUMPRODUCT(E2:E16,$I$2:$I$16)+F23
Cell B24 = SUMPRODUCT(F2:F16,$I$2:$I$16)+F24
Cell B25 = SUMPRODUCT(H2:H16,$I$2:$I$16)-F25

Cell F26 represents the sum of λ_j (=SUM(I2:I16)). Cells D20:D25 contains

D20 = INDEX(J2:J16,E18,1)*INDEX($B$2:$F$16,$E$18,1)
D21 = INDEX(J2:J16,E18,1)*INDEX($B$2:$F$16,$E$18,2)
D22 = INDEX(J2:J16,E18,1)*INDEX($B$2:$F$16,$E$18,3)
D23 = INDEX(J2:J16,E18,1)*INDEX($B$2:$F$16,$E$18,4)
D24 = INDEX(J2:J16,E18,1)*INDEX($B$2:$F$16,$E$18,5)
D25 = INDEX(H2:H16,$E$18,1)

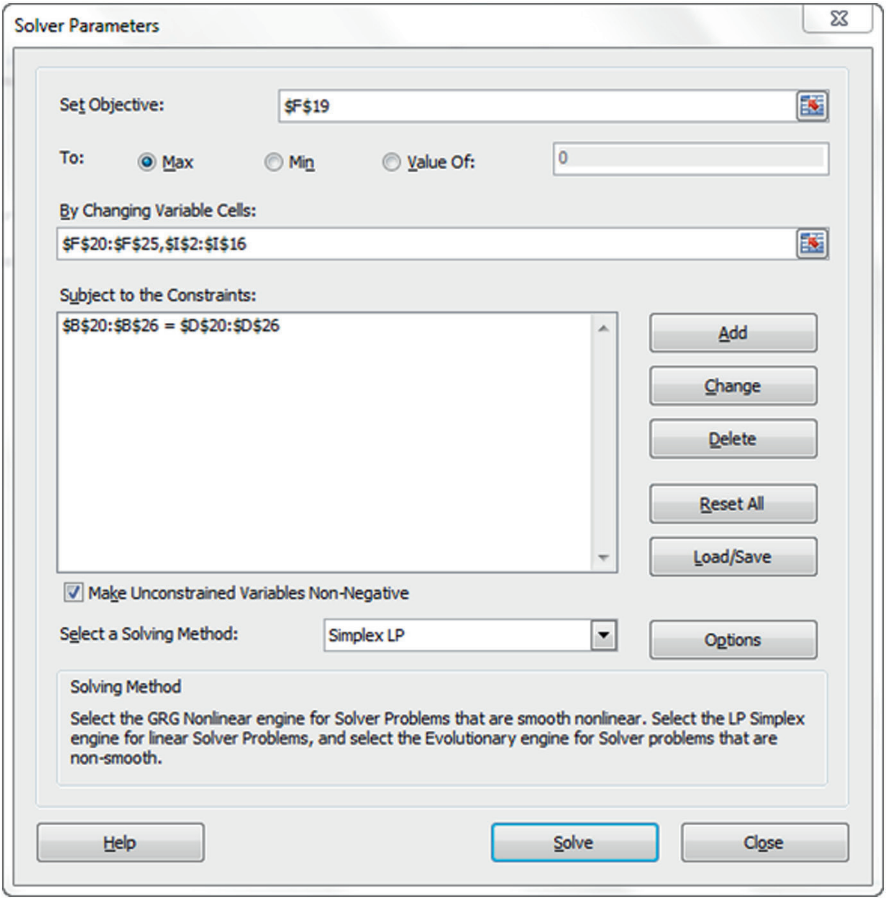


Fig. 16.8 Solver parameters for calculating DEA slacks for 15 mines

The DEA slack calculation is performed by the VBA procedure “DEASlack”

```
Sub DEASlack()
Dim i As Integer
For i = 1 To 15
Range("E18") = i
SolverSolve UserFinish:=True
Range("F20:F25").Copy
Range("L" & i + 1).Select
Selection.PasteSpecial Paste:=xlPasteValues, Transpose:=True
Next i
End Sub
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Mines	Labor	K1	K2	T1	1/D1		Output	λ	Strong		Labor	K1	K2	T1	1/D1	Output
2	1	98.5	142	245	6	0.016		3264	0	1		0.000	0.000	0.000	0.000	0.000	0.000
3	2	96.5	30	215	6	0.016		3065	0	1		0.000	0.000	0.000	0.000	0.000	0.000
4	3	57.6	18	105	5.6	0.026		2275	0.2522751	1		0.000	0.000	0.000	0.000	0.000	0.000
5	4	59.2	160	0	5.9	0.025		1978	0	1		0.000	0.000	0.000	0.000	0.000	0.000
6	5	57.6	200	0	8	0.022		1833	7.9E-17	1		0.000	0.000	0.000	0.000	0.000	0.000
7	6	49.9	27	85	4.5	0.019		1218	0	0.97129		5.384	0.000	0.000	0.000	0.000	0.000
8	7	53.5	143	65	6	0.01		928	0	1		0.000	0.000	0.000	0.000	0.000	0.000
9	8	34	70	65	6	0.02		919	0	0.93747		0.000	0.000	0.000	0.693	0.004	0.000
10	9	39.6	67.5	40	6.5	0.013		777	0	1		0.000	0.000	0.000	0.000	0.000	0.000
11	10	51.3	0	145	3.2	0.019		745	0	1		0.000	0.000	0.000	0.000	0.000	0.000
12	11	74.2	110	65	2.1	0.014		742	0	1		0.000	0.000	0.000	0.000	0.000	0.000
13	12	24	25	65	4.4	0.012		488	0.5299453	1		0.000	0.000	0.000	0.000	0.000	0.000
14	13	26.5	58	0	3	0.014		407	0.0200163	1		0.000	0.000	0.000	0.000	0.000	0.000
15	14	43.1	70	0	6.5	0.012		402	0	1		0.000	0.000	0.000	0.000	0.000	0.000
16	15	20.7	236	0	5.7	0.01		396	0.1977633	1		0.000	0.000	0.000	0.000	0.000	0.000
17																	
18		Reference		DMU under	8	Sum of δi		DEA	Congestion								
19	**Constraints**	set		Evaluation		0.0000		Slacks	Slacks								
20	Labor	31.8739	=	31.873864		0.0000	≤	0	0.0000								
21	K1	65.6227	=	65.622662		0.0000	≤	0	0.0000								
22	K2	60.9353	=	60.935329		0.0000	≤	0	0.0000								
23	T1	4.9318	=	4.9317994		0.0000	≤	0.693	0.6930								
24	1/D1	0.0152	=	0.0151764		0.0000	≤	0.004	0.0036								
25	output	919	=	919.000													
26	Σλ	1	=	1													

Fig. 16.9 Congestion slack spreadsheet model

We next calculate model (16.6) for DMUs 6 and 8. Based upon the DEA slacks in cells L2:Q16, Fig. 16.9 shows the spreadsheet for calculating the congestion slacks.

Cells F20:F24 now represent δ_i^+. Cell F19 contains the formula "=SUM (F20:F25)", and is the target cell. We change the formulas for cells B20:B25 and D20:D25 to

Cell B20 = SUMPRODUCT(B2:B16,$I$2:$I$16)-F20
Cell B21 = SUMPRODUCT(C2:C16,$I$2:$I$16)-F21
Cell B22 = SUMPRODUCT(D2:D16,$I$2:$I$16)-F22
Cell B23 = SUMPRODUCT(E2:E16,$I$2:$I$16)-F23
Cell B24 = SUMPRODUCT(F2:F16,$I$2:$I$16)-F24
Cell B25 = SUMPRODUCT(H2:H16,$I$2:$I$16)
D20 = INDEX(J2:J16,E18,1)*INDEX($B$2:$F$16,$E$18,1)-INDEX(L2:L16,E18,1)
D21 = INDEX(J2:J16,E18,1)*INDEX($B$2:$F$16,$E$18,2)-INDEX(M2:M16,E18,1)
D22 = INDEX(J2:J16,E18,1)*INDEX($B$2:$F$16,$E$18,3)-INDEX(N2:N16,E18,1)
D23 = INDEX(J2:J16,E18,1)*INDEX($B$2:$F$16,$E$18,4)-INDEX(O2:O16,E18,1)
D24 = INDEX(J2:J16,E18,1)*INDEX($B$2:$F$16,$E$18,5)-INDEX(P2:P16,E18,1)
D25 = INDEX(H2:H16,$E$18,1)+INDEX(Q2:Q16,E18,1)

Cells H20:H24 represent the DEA slacks for a DMU under evaluation and return the DEA slacks reported in cells L2:Q16. The formulas are

Cells H20 =INDEX(L2:L16,E18,1)
Cells H21 =INDEX(M2:M16,E18,1)
Cells H22 =INDEX(N2:N16,E18,1)
Cells H23 =INDEX(O2:O16,E18,1)
Cells H24 =INDEX(P2:P16,E18,1)

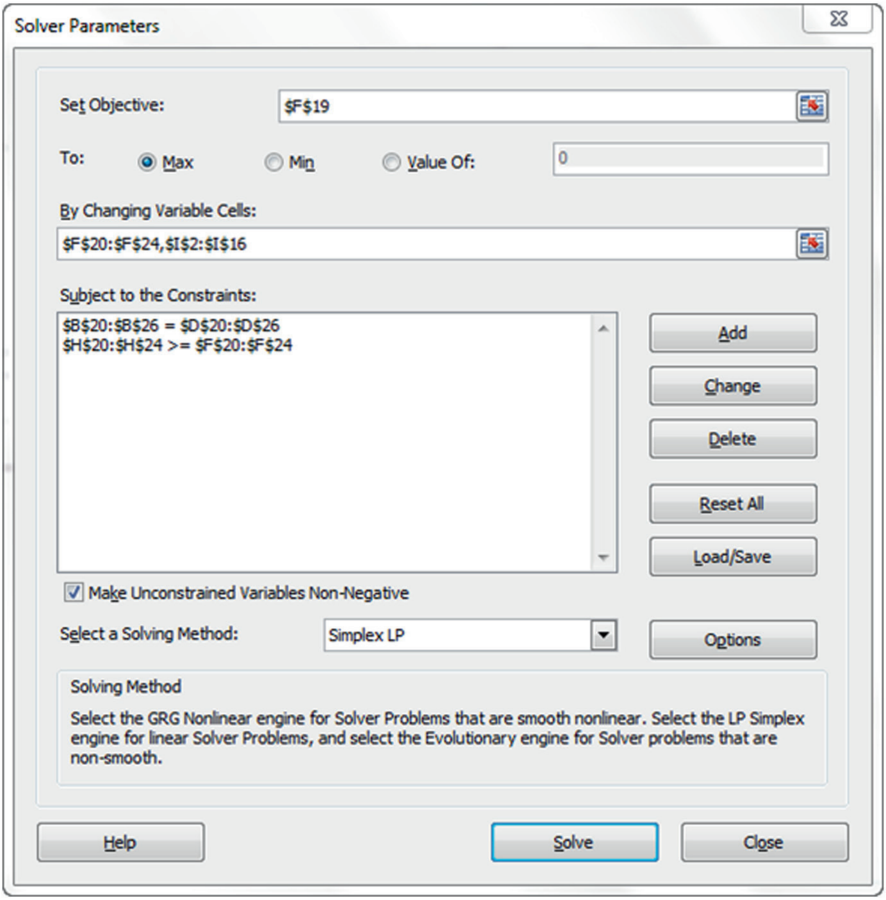


Fig. 16.10 Solver parameters for calculating congestion slacks

Figure 16.10 shows the Solver parameters for calculating the congestion slacks. The congestion slacks are reported in cells I20:I24. In this example, the congestion slacks are equal to the DEA slacks for DMUs 6 and 8, because $\delta_i^+ = 0$ in optimality. For example, for DMU 6, the congestion factor is labor with a congestion slack of 5.384.

16.4 Solving Congestion Using DEAFrontier

To run the **weak disposability models** presented in Table 16.1, select the "Weak Disposability" menu item. The results are reported in the "Efficiency" sheet.

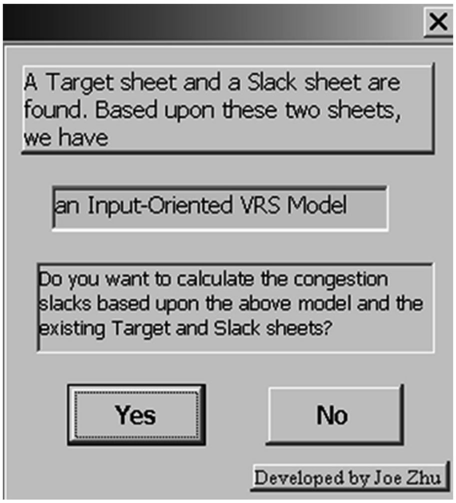


Fig. 16.11 Congestion

To calculate the **congestion slacks**, select the "Congestion" menu item. The Congestion will use the Slack and Target sheets. If there exist a slack sheet and a target sheet that are generated by the same envelopment model, you will be prompted a form shown in Fig. 16.11.

If you choose Yes, then the software will calculate the congestion slacks based upon the information stored in the "Slack", "Target" and "Data" sheets.

If you choose No, then the software will ask you to select an envelopment model. Then, the software will calculate the specified envelopment model, generate the "Slack" and "Target" sheets, and report the congestion slacks in "Congestion Slacks" sheet. The same procedure will be applied if there do not exist the "Slack" sheet and the "Target" sheet, or the "Slack" and "Target" sheets are generated by different envelopment models.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_16) contains supplementary material, which is available to authorized users.

References

Brockett, P. L., Cooper, W. W., Shin, H. C., & Wang, Y. (1998). Inefficiency and congestion in Chinese production before and after the 1978 economic reforms. *Socio-Economic Planning Sciences, 32,* 1–20.

Byrnes, P., Färe, R., & Grosskopf, S. (1984). Measuring productive efficiency: An application to Illinois strip mines. *Management Science, 30,* 671–681.

Cooper, W. W., Thompson, R. G., & Thrall, R. M. (1996). Introduction: Extensions and new developments in DEA. *Annals of Operations Research, 66,* 3–45.

Cooper, W. W., Seiford, L. M., & Zhu, J. (2000). A unified additive model approach for evaluating inefficiency and congestion with associated measures in DEA. *Socio-Economic Planning Sciences, 34*(1), 1–25.

Färe, R., & Grosskopf, S. (1983). Measuring congestion in production. *Journal of Economics, 43,* 257–271.

Färe, R., Grosskopf, S., & Lovell, C. A. K. (1994). *Production frontiers*. Cambridge: Cambridge University Press.

Ray, S., Seiford, L. M., & Zhu, J. (1998). Market entity behavior of Chinese state-owned enterprises. *OMEGA, 26*(2), 263–278.

Chapter 17
Identifying Critical Measures in DEA

17.1 Introduction

Since each DMU has its own inherent tradeoffs among the multiple measures that significantly influence the performance, it is extremely important for the management to know the critical measures. The current chapter introduces the approach of Chen and Zhu (2003) for identifying the critical measures to DMUs' performance. Note that once the DEA evaluation is done, the management needs to either (i) maintain the best practice for the efficient DMUs or (ii) achieve the best practice for the inefficient DMUs. Thus, when a set of multiple performance measures is determined, measures that are influential to maintaining and achieving the best practice should be regarded as critical to the performance of DMUs. Also, it is believed that a critical measure is signaled by whether changes in its value affect the performance, not by whether inclusion or exclusion of the measure affects the performance. Under the framework of DEA sensitivity analysis, Chen and Zhu (2003) develop an alternative approach, which is independent of identifying DEA weights or DEA multipliers, to identify such critical measures.

17.2 Performance Evaluation and DEA

Regression-based methods can be used in evaluating performance of a set of DMUs. However, they are limited to only one dependent variable. For example,

$$y = \beta_o + \sum_{i=1}^{m} \beta_i x_i + \varepsilon \tag{17.1}$$

where β_i are estimated coefficients which can be used to determine whether an independent variable has a positive effect on the dependent variable or makes an important contribution. i.e., by estimating the coefficients, we may identify the critical

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_17, © Springer International Publishing Switzerland 2014

performance measures under the context of average behavior. Also, the estimated regression line can be served as the benchmark in performance evaluation.

In fact, formula (17.1) can be viewed as a performance frontier or tradeoff curve where x_i are inputs and y is the output. However, we are very likely to have multiple outputs $y_r (r=1, \ldots, s)$. We may rewrite (17.1) as (Wilkens and Zhu 2001)

$$\sum_{r=1}^{s} u_r y_r = \alpha + \sum_{i=1}^{m} v_i x_i \tag{17.2}$$

where u_r and v_i are unknown weights representing the relative importance or tradeoffs among y_r and x_i.

Suppose we can estimate u_r and v_i, then for each DMU_j, we can define

$$h_j = \frac{\alpha + \sum_{i=1}^{m} v_i x_{ij}}{\sum_{r=1}^{s} u_r y_{rj}} \tag{17.3}$$

as a performance index, where x_{ij}, $(i=1, 2, \ldots, m)$ are multiple inputs, y_{rj}, $(r=1, 2, \ldots, s)$ are multiple outputs for DMU_j $(j=1, 2, \ldots, n)$.

In order to estimate u_r and v_i, and further evaluate the performance of j_o th DMU, (denoted as DMU_o) by (17.2), DEA uses the following linear fractional programming problem

$$\begin{aligned} & \min_{\alpha, v_i, u_r} \frac{\alpha + \sum_{i=1}^{m} v_i x_{io}}{\sum_{r=1}^{s} u_r y_{ro}} \\ & \text{subject to} \\ & \frac{\alpha + \sum_{i=1}^{m} v_i x_{ij}}{\sum_{r=1}^{s} u_r y_{rj}} \geq 1, j = 1, \ldots, n \\ & u_r, v_i \geq 0 \quad \forall\, r, i \end{aligned} \tag{17.4}$$

where, x_{io} and y_{ro} are respectively the ith input and rth output for DMU_o under evaluation.

When $h_o^* = 1$, DMU_o is efficient or on the performance frontier. Otherwise, if $h_o^* > 1$, then DMU_o is inefficient. All the efficient DMUs constitute the performance frontier.

Note that when $h_o^* = 1$, we have

$$\sum_{r=1}^{s} u_r^* y_{ro} = \alpha^* + \sum_{i=1}^{m} v_i^* x_{io} \tag{17.5}$$

where (*) represents optimal values in model (17.4). That is, DEA estimates the "coefficients" in (17.2). It can be seen that while (17.1) estimates one set of coefficients, DEA model (17.4) estimates one set of coefficients for each DMU, resulting a piecewise linear tradeoff curve represented by several (17.5)-like equations associated with efficient DMUs. Equation (17.5) is theoretically available, but very difficult to obtain empirically.

Obviously, u_r^* and v_i^* represent the tradeoffs among various outputs and inputs. If we can obtain the exact information on u_r^* and v_i^*, the critical performance measures can be easily identified. However, the exact information on u_r^* and v_i^* cannot be obtained because of multiple optimal solutions in the multiplier models.

However, in order to solve model (17.4), the following transformation is used

$$\mathrm{t} = \frac{1}{\sum_{r=1}^{s} u_r y_{ro}},\ \omega_i = \mathrm{t}\mathrm{v}_i,\ \omega_o = \mathrm{t}\alpha,\ \mu_r = \mathrm{t}\mu_r \tag{17.6}$$

Based upon (17.6), model (17.4) is solved in the following equivalent linear programming problem (VRS multiplier model, see Chap. 3)

$$\begin{aligned} &\min_{\omega_o,\omega_i,\mu_r} \ \omega_o + \sum_{i=1}^{m} \omega_i x_{io} \\ &\text{subject to} \\ &\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{m} \omega_i x_{ij} - \omega_o \le 0 \quad \forall j \\ &\sum_{r=1}^{s} \mu_r y_{ro} = 1 \\ &\mu_r, \omega_i \ge 0 \quad \forall\, r, i \end{aligned} \tag{17.7}$$

or the dual to model (17.7) (VRS envelopment model, see Chap. 2)

$$\begin{aligned} &\phi_o^* = \max \phi_o \\ &\text{subject to} \\ &\sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io} \qquad i = 1,2,\ldots,m; \\ &\sum_{j=1}^{n} \lambda_j y_{rj} \ge \phi_o y_{ro} \quad r = 1,2,\ldots,s; \\ &\sum_{j=1}^{n} \lambda_j = 1 \\ &\lambda_j \ge 0 \qquad\qquad j = 1,\ldots,n. \end{aligned} \tag{17.8}$$

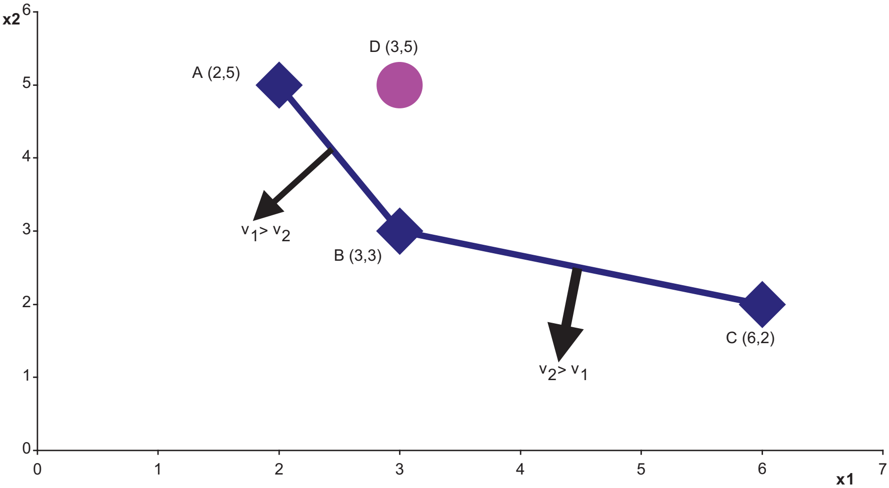


Fig. 17.1 Critical measures and tradeoffs

Based upon (17.6), we have $\frac{\omega_i}{\omega_k} = \frac{v_i}{v_k}$ and $\frac{\mu_r}{\mu_d} = \frac{u_r}{u_d}$. Thus, μ_r^* and ω_i^* are not the exact weights representing the tradeoffs in model (4). In addition, for efficient DMUs, model (17.7) often yields multiple optimal solutions on multipliers μ_r and ω_i. Also, $\sum_{r=1}^{s} \mu_r^* y_{rj} - \sum_{i=1}^{m} \omega_i^* x_{ij} - \omega_o^* = 0$ may only represent supporting hyperplanes rather than the performance frontier in empirical studies. This further leads to an incomplete tradeoff information. Because of possible multiple optimal solutions in (17.7) and the transformation in (17.6), it is very difficult to back out the tradeoffs represented by u_r^* and v_i^* in model (17.4), i.e., the performance frontier expressed by (17.5) is very difficult to obtain in empirical applications. Chen and Zhu (2003) therefore develop an alternative approach to identifying the critical measures.

Suppose that we obtain the performance frontier. In this case, for example $v_k^* > v_i^*$ indicates that the kth input measure is more influential in order for DMU_o to achieve the best-practice. i.e., the kth input is more important to DMU_o's performance which is characterized by the efficiency score (h_o^*). Note also that the DEA model (17.4) always tries to assign larger v_i and u_r to smaller x_{io} and larger y_{ro} respectively in order to achieve the optimality. This indicates that when a set of multiple performance measures (inputs and outputs) is determined, the relative importance or tradeoffs is determined by the magnitudes of the inputs and outputs.

It can be seen from model (17.4) that for a specific DMU under evaluation, when a specific input increases, the associated input weight will not increase and when a specific output decreases, the associated output weight will not increase. Consider the frontier represented by ABC in Fig. 17.1 with two inputs and a single output. In Fig. 17.1, $v_1 > v_2$ remains true for facet AB if DMU A's x_2 (uncritical one) changes

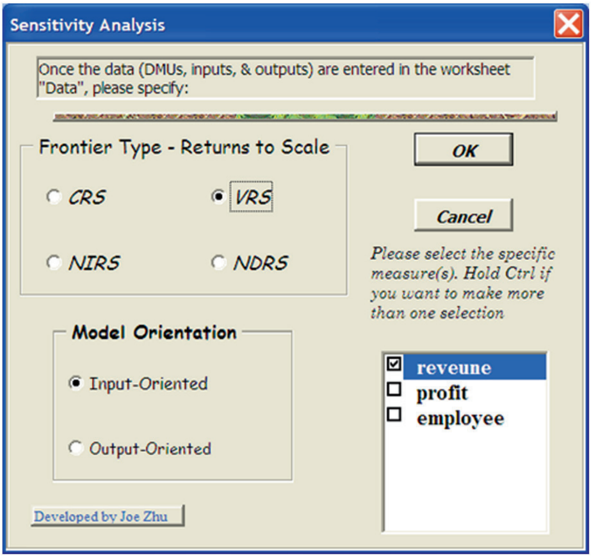


Fig. 17.2 Critical measures and tradeoffs

its value, and $v_2 > v_1$ remains true for facet BC if DMU C's x_1 (uncritical one) changes its value. Meanwhile, DMUs A and C remain efficient when the uncritical inputs changes their value, respectively[1]. However, if we increase the x_1 of DMU A or x_2 of DMU C to a certain level, DMU A or DMU C becomes inefficient.

The example in Fig. 17.2 indicates that (a) for efficient DMUs, the performance is determined and characterized by the best-practice status, and (b) for inefficient DMUs, the performance is determined and characterized by the distance to the frontier. Thus, a measure that is critical to the performance should be characterized by whether the measure is critical to (i) maintaining the best-practice for efficient DMUs and (ii) achieving the best-practice for inefficient DMUs.

Because a set of multiple performance measures is given prior to the evaluation, a critical measure is signaled by whether changes in its value affect the performance, not by whether inclusion or exclusion of the measure affects the performance.

Definition 17.1 When a set of multiple performance measures is given, a specific measure is said to be critical if changes in its value may alter the efficiency status of a specific DMU.

For efficient DMUs, the performance is determined and characterized by the best practice status. For inefficient DMUs, the performance is determined and characterized by the distance to the frontier. Thus, a measure that is critical to the performance should be characterized by whether the measure is critical to (i) maintaining the best practice for efficient DMUs and (ii) achieving the best practice for inefficient DMUs.

[1] Note that for example, if the second input of DMU A decreases its current level to 3, the level used by DMU B, then we no longer have the efficient facet AB. Since DMU B becomes inefficient.

17.3 Identifying Critical Output Measures

Consider the following super-efficiency model where the dth output is given the pre-emptive priority to change

$$\begin{aligned}
&\max \sigma_d \\
&\text{subject to} \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j y_{dj} \geq \sigma_d y_{do} \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r \neq d \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j x_{ij} \leq x_{io} \qquad i = 1,\ldots,m \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j = 1
\end{aligned} \tag{17.9}$$

Four possible cases are associated with (17.9): (i) $\sigma_d^* > 1$, (ii) $\sigma_d^* = 1$, (iii) $\sigma_d^* < 1$ and (iv) model (17.9) is infeasible. When $\sigma_d^* > 1$, DMU_o has inefficiency in its dth output, since potential output increase can be achieved by DMU_o. Cases (ii), (iii) and (iv) indicate that no inefficiency exists in dth output.

Now, we consider the efficient DMUs and assume that DMU_o is efficient. Based upon model (17.6) the set of s outputs can be grouped into two subsets: set $O = \{d : \sigma_d^* \leq 1\}$ and set $\bar{O} = \{d : \text{model (17.9) is infeasible for } d\text{th output}\}$.

We have when model (17.9) is infeasible, the magnitude of the dth output across all DMUs has nothing to do with the efficiency status of DMU_o.

This indicates that the outputs in set $\bar{O}$ are not critical to the efficency status of DMU_o, since changes in the outputs in set $\bar{O}$ do not change the efficiency classification of DMU_o. The efficiency classification of DMU_o is stable to any changes in the dth output across all DMUs when d belongs to set $\bar{O}$.

However, decreases in outputs in set O to certain magnitudes result in a change of efficiency status (performance) of DMU_o. For example, when the dth output of DMU_o is decreased from the current level y_{do} to a level which is less than $\sigma_d^* y_{do}$ ($\sigma_d^* < 1$), then DMU_o becomes inefficient. This in turn indicates that the outputs in set O are critical to the performance of DMU_o.

Now, let $\mathrm{P}_{d*} = \max\{\sigma_d^*\}$ for the outputs in set O. From the above discussion, we conclude that the d*th output is the most critical output measure to the efficiency of DMU_o. Because, DMU_o's efficiency status is most sensitive to changes in the d*th output.

Next, we consider inefficient DMUs and assume that DMU_o is inefficient. For inefficient DMUs, the issue is how to improve the inefficiency to achieve the best practice. Since the focus here is how each individual output measure contributes to the performance of DMU_o, we solve model (17.9) for each d and obtain $\sigma_d^* > 1(d = 1,\ldots,d)$ where σ_d^* measures how far DMU_o is from the frontier in terms of dth output.

As a matter of fact, model (17.9) provides an alternative way to characterize the inefficiency of DMU_o. Each σ_d^* indicates possible inefficiency existing in each associated output when other outputs and inputs are fixed at their current levels. We then can rank the inefficiency by each optimal σ_d^*. Let $G_{d*} = \min\left\{\sigma_d^*\right\}$ That is, the d*th output indicates the least inefficiency. If the DMU_o is to improve its performance through single output improvement, the d*th output will yield the most effective way. Because G_{d*} represents the shortest path onto the best practice frontier when each output is given the pre-emptive priority to improve. We therefore define that the d*th output is the most critical output to reach the performance frontier and to DMU_o's performance.

In summary, the critical output is identified as the output associated with max $\left\{\sigma_d^*\right\}$ for efficient DMUs and $\min\left\{\sigma_d^*\right\}$ for inefficient DMUs.

17.4 Identifying Critical Input Measures

Consider the following super-efficiency model when the kth input measure is of interest.

$$
\begin{aligned}
&\min \tau_k \\
&\text{subject to} \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j x_{kj} \leq \tau_k x_{ko} \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j x_{ij} \leq x_{io} \qquad i \neq k \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r = 1,\ldots,s \\
&\sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j = 1
\end{aligned}
\tag{17.10}
$$

Based upon model (17.10), we have (i) $\tau_k^* < 1$, (ii) $\tau_k^* = 1$, (iii) $\tau_k^* > 1$, and (iv) (17.10) is infeasible. Case (i) indicates that inefficiency exists in DMU_o's kth input, since DMU_o needs to decrease its kth input to $\tau_k^* x_{ko}$ in order to reach the performance frontier. Cases (ii), (iii) and (iv) indicate that no inefficiency exists in DMU_o's kth input.

Table 17.1 Critical measures for the numerical example

DMU	τ_1^*	τ_2^*
A	3/2	Infeasible
B	14/9	17/12
C	Infeasible	2
D	2/3	3/5

Now, suppose DMU_o is efficient. Based upon model (17.10), the set of m inputs can be grouped into two subsets: set $I = \{k : \tau_k^* \geq 1\}$ and set $\overline{I} = \{k:$ model (17.10) is infeasible for kth input$\}$.

We have when model (17.10) is infeasible, the magnitude of the kth input across all DMUs has nothing to do with the efficiency status of DMU_o.

This indicates that the inputs in set $\overline{I}$ are not critical to the efficiency status of DMU_o, since changes in the inputs in set $\overline{I}$ do not change the efficiency classification of DMU_o. Let $\mathrm{T}_{k^*} = \min\{\tau_k^*\}$ for inputs in set I. We conclude that the k^*th input is the most critical input measure to the efficiency of DMU_o. Because, DMU_o's efficiency status is most sensitive to changes in the k^*th input.

Next, suppose DMU_o is inefficient. We solve model (17.10) for each k and obtain $\tau_k^* < 1(k=1,\ldots,m)$, where τ_k^* measures how far DMU_o is from the frontier in terms of kth input. Each τ_k^* indicates possible inefficiency existing in each associated input when other inputs and outputs are fixed at their current levels. We then can rank the inefficiency by each optimal τ_k^*. Let $H_{k^*} = \max_k\{\tau_k^*\}$. Similar to the discussion on identifying the critical output measure, we say that the k^*th input is the most critical input to reach the performance frontier and to DMU_o's performance, since the k^*th input indicates the least inefficiency.

In summary, the critical input is identified as the input associated with $\min\{\tau_k^*\}$ for efficient DMUs and $\max\{\tau_k^*\}$ for inefficient DMUs.

17.5 Numerical Example and Extension

To further illustrate the rationale of the approach, consider again the four DMUs shown in Fig. 17.1. Table 17.1 reports the optimal value to model (17.10). It can be seen that for DMU D, the first input is the critical measure since DMU D's efficiency can be easily improved if the first input is given the pre-emptive priority to change. For DMU A, the infeasibility associated with the second input indicates that the first input is the critical measure. Note that the efficient facet AB shows that the first input is more important than the second one, since $v_1 > v_2$. Our approach also indicates that the second input is the critical measure to DMU C's performance. This finding is confirmed by the fact that $v_2 > v_1$ in BC. As for DMU B, since it is located at the intersect of AB and BC, it is very difficult to determine which input is the critical factor by looking at the coefficients of efficient facets. Our approach indicates that the second input is the critical one for DMU B, since $\tau_2^* < \tau_1^*$ (17/12 < 14/9).

The above discussion assumes that DMUs are able to adjust each input and each output while other inputs and outputs are fixed. Situations when some measures are strongly related with each other may occur. In that case, a set of inputs or outputs has to be adjusted simultaneously and we need to consider the measures in groups. We use the following models.

$$
\begin{aligned}
& \min \mathrm{T}_M \\
s.t. \quad & \sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j x_{ij} \le \mathrm{T}_M x_{io} && i \in M \\
& \sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j x_{ij} \le x_{io} && i \notin M \\
& \sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j y_{rj} \le y_{ro} && r = 1,\ldots,s \\
& \sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j = 1
\end{aligned}
\qquad (17.11)
$$

and

$$
\begin{aligned}
& \max \Omega_Q \\
s.t. \quad & \sum_{j=1}^{n} \lambda_j y_{rj} \ge \Omega_Q y_{ro} && r \in Q \\
& \sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} && r \notin Q \\
& \sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io} && i = 1,\ldots,m \\
& \sum_{\substack{j=1 \\ j\neq o}}^{n} \lambda_j = 1
\end{aligned}
\qquad (17.12)
$$

where inputs represented by set M and outputs represented by set Q are of interest.

Similar to the previous discussions, when DMU_o is inefficient, we use $\max\left\{T_M^*\right\}$ and $\min\left\{\Omega_Q^*\right\}$ to identify the most critical input and output measures, respectively. When DMU_o is efficient, infeasibility associated with (17.11) and (17.12) indicates the non-critical inputs and outputs.

The above discussion is based upon the assumption that the DEA frontier exhibits VRS. The development can be applied to other DEA models with non-VRS frontiers discussed in Chap. 11.

17.6 Application to Fortune E-Companies

To capture the Internet's effect on the economy, at the end of year 1999, Fortune magazine launched the Fortune e-50 index which consists of 50 corporations who integrate the Internet, computers and enterprise softwares to do the business. As stated in the 1999 December Fortune issue, each of the e-50 is or has the potential to be a major player in the Internet economy. The list of e-corporation is decided by that a company must have been public for at least 6 months and must have a market capital value that exceeds $100 million. Table 17.2 provides the list of the e-50.

Market capital, profit, revenue and number of employees are provided by the Fortune as the four standard measures to fully characterize the performance of the e-50 corporations. We therefore use them as a set of multiple performance measures. The data on profit, employee and market capital are not available for Ariba (DMU26), and therefore Ariba is excluded from the following analysis.

Because we are interested in the contribution of revenue, profit and employee to the market value, we select the market capital as the DEA output and the other measures as the DEA inputs. Output-oriented DEA model is used, because higher market values are desirable given the current levels of revenue, profit and the number of employees.

The third column of Table 17.3 reports the optimal value to the output-oriented VRS envelopment model. Ten e-corporations are on the performance frontier.

Next, we apply the newly developed method to identify the critical input measures to the market capital under the context of best-practice. Columns 3, 4 and 5 of Table 17.4 report the results from model (17.10).

We use the DEAFrontier software to do the calculation. Once the data are entered into the "data" sheet, we select "Perform Sensitivity Analysis" and then select the input as shown in Fig. 17.2. The results are reported in the "Sensitivity Report" sheet. We can select one input at a time.

For example, consider MCI WorldCom (DMU48), model (17.10) is infeasible when revenue and employee are under consideration (selected) respectively and model (17.10) yields the optimal value of 96.98 when profit is under consideration. This indicates that once the three input measures are determined, the magnitudes of revenue and employee do not affect the efficiency status of MCI WorldCom. However, the value of profit affects MCI WorldCom's efficiency status given the current levels of market value, revenue and employee. Thus, profit is the critical factor to MCI WorldCom's performance.

Table 17.2 Fortune's e-corporations

DMU no.	Name	Revenue $ millions	Profits $ millions	Employees	Market capital $ millions	Year founded
E-Companies						
1	America Online	4777	762	12100	164308	1985
2	Charles Schwab	4113	498	13300	34194	1986
3	Amazon.com	1015	−291	2100	21202	1994
4	E*Trade Group	621	−54	1735	8341	1982
5	Knight/Trimark Group	618	119	446	4389	1995
6	Yahoo	341	22	803	47946	1995
7	Ameritrade Holding	301	12	985	3740	1992
8	EarthLink Network	254	−88	1343	1409	1994
9	Priceline.com	189	−125	194	7963	1998
10	CMGI	176	476	1024	12567	1986
11	Lycos	136	−52	456	5687	1995
12	Excite@Home	129	−324	570	14647	1995
13	eBay	125	7	138	17106	1995
14	DoubleClick	103	−22	482	5947	1996
15	RealNetworks	89	−4	434	9148	1994
16	CNet	79	40	491	3481	1995
17	Healtheon	68	−68	648	2347	1995
18	eToys	38	−47	306	6276	1996
19	VerticalNet	8	−21	220	2515	1995
Net software and service companies						
20	Microsoft	19747	7785	31396	471573	1975
21	Oracle	9063	1332	44000	85776	1977
22	Intuit	848	377	3675	5942	1983
23	Network Associates	785	−127	2700	2871	1992
24	Cambridge Tech. Partners	628	35	4444	726	1991
25	TMP Worldwide	585	10	5200	2976	1967
26	Ariba	45.4	*	*	*	1996

Table 17.2 (continued)

DMU no.	Name	Revenue $ millions	Profits $ millions	Employees	Market capital $ millions	Year founded
27	Citrix Systems	323	93	620	7169	1989
28	Macromedia	167	24	553	2690	1992
29	Network Solutions	142	17	385	4801	1979
30	Concentric Network	110	−82	508	1054	1991
31	Exodus Communications	108	−82	472	7080	1992
32	BroadVision	71	10	271	6777	1993
33	Inktomi	71	−24	185	5709	1996
34	Security First Technologies	44	−19	312	1345	1995
35	Razorfish	36	2	414	1896	1995
Net hardware companies						
36	IBM	87448	7701	291067	167567	1911
37	Lucent Technologies	38303	4766	153000	211415	1995
38	Intel	28194	7371	64500	285803	1968
39	Dell Computer	21670	1750	24400	110530	1984
40	Cisco Systems	12154	2096	21000	237215	1984
41	Sun Microsystems	11726	1031	29700	85861	1982
42	EMC	4459	967	9700	75371	1979
43	Qualcomm	3937	201	11600	43919	1981
44	Network Appliance	335	42	816	6327	1992
45	Broadcom	335	40	436	15994	1991
46	Juniper Networks	31	−30	190	14455	1992
Net communication companies						
47	AT&T	56968	6037	107800	154791	1875
48	MCI WorldCom	30720	−883	77000	162492	1983
49	Qwest Communications	3424	−5	8700	27404	1997
50	Global Crossing	691	79	10000	26109	1997

Table 17.3 Performance evaluation of Fortune's e-corporations

DMU no.	Name	VRS
1	America Online	1.00000
2	Charles Schwab	3.83409
3	Amazon.com	1.05723
4	E*Trade Group	5.31514
5	Knight/Trimark Group	7.15192
6	Yahoo	1.00000
7	Ameritrade Holding	11.63487
8	EarthLink Network	25.09020
9	Priceline.com	1.00000
10	CMGI	2.39677
11	Lycos	4.30319
12	Excite@Home	1.00000
13	eBay	1.00000
14	DoubleClick	3.71566
15	RealNetworks	2.26509
16	CNet	5.64226
17	Healtheon	7.23136
18	eToys	2.32675
19	VerticalNet	1.00000
20	Microsoft	1.00000
21	Oracle	2.31196
22	Intuit	10.30718
23	Network Associates	13.81890
24	Cambridge Tech. Partners	72.12135
25	TMP Worldwide	16.68021
27	Citrix Systems	5.50414
28	Macromedia	10.83562
29	Network Solutions	5.34769
30	Concentric Network	19.73886
31	Exodus Communications	2.90959
32	BroadVision	2.74452
33	Inktomi	2.77061
34	Security First Technologies	11.79142
35	Razorfish	7.90885
36	IBM	2.79636
37	Lucent Technologies	1.72137
38	Intel	1.59834
39	Dell Computer	2.03503
40	Cisco Systems	1.00000
41	Sun Microsystems	2.24478
42	EMC	1.94255
43	Qualcomm	2.22318
44	Network Appliance	1.93360
45	Broadcom	7.47555
46	Juniper Networks	1.00000
47	AT&T	2.64384
48	MCI WorldCom	1.00000
49	Qwest Communications	2.61124
50	Global Crossing	2.18328

Table 17.4 Critical measures for Fortune's e-corporations

DMU no.	Name	Revenue	Profit	Employee	Critical measures	
					(17.10)	(17.11)
1	America Online	Infeasible	Infeasible	Infeasible		{profit, revenue}
2	Charles Schwab	0.0520	0.3619	0.0381	{profit}	
3	Amazon.com	0.5563	0.9817	0.7884	{profit}	
4	E*Trade Group	0.0463	0.6708	0.0945	{profit}	
5	Knight/Trimark	0.0188	0.6297	0.3094	{profit}	
6	Yahoo	Infeasible	Infeasible	Infeasible		{profit,revenue,employee}
7	Ameritrade Holding	0.0344	0.6283	0.1401	{profit}	
8	EarthLink Network	0.1368	0.7066	0.1328	{profit}	
9	Priceline.com	Infeasible	1.1200	1.5994	{profit}	{profit, revenue}
10	CMGI	0.1555	0.4180	0.1348	{profit}	
11	Lycos	0.1736	0.7581	0.3700	{profit}	
12	Excite@Home	Infeasible	1.4459	Infeasible	{profit}	{profit, revenue}
13	eBay	Infeasible	Infeasible	1.7284	{employee}	Infeasible
14	DoubleClick	0.1419	0.7375	0.3361	{profit}	
15	RealNetworks	0.2335	0.7661	0.3639	{profit}	
16	CNet	0.1248	0.7460	0.3329	{profit}	
17	Healtheon	0.3937	0.8759	0.3337	{profit}	
18	eToys	0.6037	0.9469	0.6886	{profit}	
19	VerticalNet	3.8750	Infeasible	Infeasible	{revenue}	{profit, revenue}
20	Microsoft	Infeasible	Infeasible	Infeasible		{profit,revenue,employee}
21	Oracle	0.1968	0.3002	0.0803	{profit}	
22	Intuit	0.0172	0.4408	0.0376	{profit}	
23	Network Associates	0.0641	0.7296	0.0733	{profit}	
24	Cambridge Tech.	0.0127	0.6063	0.0311	{profit}	
25	TMP Worldwide	0.0152	0.6238	0.0265	{profit}	
27	Citrix Systems	0.0525	0.5797	0.2226	{profit}	

Table 17.4 (continued)

DMU no.	Name	Revenue	Profit	Employee	Critical measures	
					(17.10)	(17.11)
28	Macromedia	0.0499	0.6331	0.2496	{profit}	
29	Network Solutions	0.0874	0.7455	0.3584	{profit}	
30	Concentric Network	0.2942	0.7667	0.3922	{profit}	
31	Exodus Comm.	0.3326	0.7938	0.4261	{profit}	
32	BroadVision	0.2283	0.8749	0.6195	{profit}	
33	Inktomi	0.5639	0.9775	0.9381	{profit}	
34	Security First Tech.	0.1818	0.9023	0.5859	{profit}	
35	Razorfish	0.2222	0.8968	0.4523	{profit}	
36	IBM	0.0564	0.0185	0.0324	{revenue}	
37	Lucent Technologies	0.1846	0.2452	0.0824	{profit}	
38	Intel	0.3794	0.4202	0.2788	{profit}	
39	Dell Computer	0.1258	0.3242	0.2181	{profit}	
40	Cisco Systems	Infeasible	1.0754	Infeasible	{profit}	{profit, revenue}
41	Sun Microsystems	0.1524	0.2609	0.1192	{profit}	
42	EMC	0.3110	0.4847	0.2870	{profit}	
43	Qualcomm	0.0772	0.5508	0.0617	{profit}	
44	Network Appliance	0.1351	0.7314	0.3165	{profit}	
45	Broadcom	0.0458	0.6096	0.1691	{profit}	
46	Juniper Networks	3.5327	Infeasible	Infeasible	{revenue}	{profit, revenue}
47	AT&T	0.0775	0.0025	0.0790	{employee}	
48	MCI WorldCom	Infeasible	96.9787	Infeasible	{profit}	{profit, revenue}
49	Qwest Comm.	0.0441	0.5771	0.0443	{profit}	
50	Global Crossing	0.2010	0.6601	0.0332	{revenue}	

Consider Charles Schwab (DMU2) which is an inefficient unit. The optimal values to model (17.10) indicate that the profit measure is the critical one for Charles Schwab to achieve the performance frontier.

The sixth column of Table 17.4 reports the critical measure identified on the basis of model (17.10). However, for efficient DMUs, it is likely that model (17.10) is infeasible for each input measure. Samples can be found in America Online (DMU1), Yahoo (DMU6) and Microsoft (DMU20). This may imply that some measures must be considered in groups. We therefore employ model (17.11) for all possible combinations of the three input measures. The last column of Table 17.4 reports the results based upon model (17.11). Note that model (17.11) is not applied to the inefficient DMUs.

For Yahoo and Microsoft, model (17.11) is feasible (has optimal solutions) when only all three inputs are in set M. For America Online, model (17.7) is feasible (has optimal solutions) when profit and revenue are in set M.

Model (17.11) is also applied to the remaining 7 efficient e-corporations, namely, Excite@Home (DMU12), Vertical Net (DMU19), Cisco System (DMU40), Juniper Networks (DMU46) and MCI WorldCom (DMU48). Model (17.10) is feasible when profit and revenue are in set M.

Except for America Online, Yahoo, eBay, Vertical Net, Microsoft, IBM, Juniper Networks, AT&T and Global Crossing, all the e-corporations indicate profit as their critical measure. This confirms that for the majority of the e-corporations that are rely on the Internet for business, revenue does not necessarily mean profit. In fact, about 40% of the e-corporations had negative profit in year 1999. (The negative values are treated by the translation invariance property in DEA. See Chap. 8.)

A closer look at Table 17.4 indicates that America Online, Yahoo and Microsoft have distinguished themselves from the e-corporations, because the results from model (17.11) imply that their high revenue means profit. Note that among the inefficient units, employee is identified as the critical measure for eBay and AT&T, and revenue is identified as the critical measure for IBM.

The e-corporations actually represent the twenty-first century new economy where the electronic and information technologies are heavily used. To further illustrate the approach, we next apply models (17.10) and (17.11) to the Fortune 1000 companies in 1995 who represent old economy where the companies design, build and deliver physical, molecular-based products to customer. The purpose is to see whether the new economy e-corporations behave differently compared to the old economy companies in terms of the critical measures.

Since the e-corporations belong to computer and telecommunication industries, we exclude all those Fortune's 1000 companies who are in the computer and telecommunication industries from the analysis. We also exclude those Fortune 1000 companies who do not have complete data on the four performance measures. As a result, we have 51 industries with 760 companies which are different from the e-corporations (see the first column in Table 17.5).

Table 17.5 summarizes the results from the new approach. The second column reports the number of companies in each industry. The third, fourth and fifth columns report how many companies indicate revenue, profit and employee as their

Table 17.5 Critical measures for Fortune's 1000 companies

Industry	Companies	Revenue (%)	Profit (%)	Employee (%)
Advertising, marketing	4	100	0	0
Aerospace	11	90.91	9.09	0
Airlines	9	100	0	0
Apparel	5	100	0	0
Beverages	7	100	0	0
Brokerage	7	100	0	0
Building materials, glass	4	100	0	0
Chemicals	39	97.44	2.56	0
Commercial banks	55	98.18	1.82	0
Diversified financials	14	92.86	7.14	0
Electric and gas utilities	73	98.63	0	1.37
Electronics, electrical equipment	41	95.12	4.88	0
Engineering, construction	11	90.91	0	9.09
Entertainment	3	33.33	33.33	33.33
Food	27	92.59	0	7.41
Food and drug stores	20	100	0	0
Food services	5	80.00	20.00	0
Forest and paper products	30	100	0	0
Furniture	5	100	0	0
General merchandisers	16	87.50	12.50	0
Health care	18	100	0	0
Hotels, casinos, resorts	7	100	0	0
Industrial and farm equipment	27	100	0	0
Insurance: life & health	19	94.74	5.26	0
Insurance: prop. & casualty	24	87.50	12.50	0
Mail, package and freight delivery	3	100	0	0

Table 17.5 (continued)

Industry	Companies	Revenue (%)	Profit (%)	Employee (%)
Marine services	2	100	0	0
Metal products	11	100	0	0
Metals	21	100	0	0
Mining, crude-oil production	7	100	0	0
Motor vehicles and parts	21	90.48	9.52	0
Petroleum refining	18	50.00	33.33	16.67
Pharmaceuticals	14	85.71	14.29	0
Pipelines	10	80.00	0	20.00
Publishing, printing	17	100	0	0
Railroads	5	100	0	0
Rubber and plastic products	8	100	0	0
Savings institutions	8	100	0	0
Scientific, photo., control equip.	18	94.44	5.56	0
Soaps, cosmetics	8	87.50	12.50	0
Specialist retailers	30	100	0	0
Temporary help	5	100	0	0
Textiles	6	100	0	0
Tobacco	4	75.00	25.00	0
Toys, sporting goods	3	100	0	0
Transportation equipment	5	100	0	0
Truck leasing	2	100	0	0
Trucking	3	100	0	0
Waste management	3	100	0	0
Wholesalers	40	90.00	0	10.00
Miscellaneous	7	100	0	0
Total	760	94.61	3.55	1.84

critical measures respectively. For example, the second row in Table 17.5 indicates that (i) there are 4 companies in the advertising and marketing industry, and (ii) revenue is identified as the critical measure for all companies. In the motor vehicle industry, only two companies (General Motor and Ford) (9.52 %; two out of 21) indicate that profit is the critical measure while other 19 companies indicate that revenue is the critical measure.

Our approach indicates that revenue is the critical factor to 95 % of the 760 companies in the Fortune's top 1000 list. In fact, these "old-economy" companies sever relatively mature market or command a lead in markets where they compete. Our finding is consistent with the belief that revenue means a stable proportion of the profit for the old economy companies. Also, our approach does indicate that the e-corporations and the Fortune's 1000 companies behave differently.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_17) contains supplementary material, which is available to authorized users.

References

Chen, Y., & Zhu, J. (2003). DEA models for identifying critical performance measures. *Annals of Operations Research, 124*(1–4), 225–244.

Wilkens, K., & Zhu, J. (2001). Portfolio evaluation and benchmark selection: A mathematical programming approach. *Journal of Alternative Investments, 4*(1), 9–19.

Chapter 18
Interval and Ordinal Data in DEA

18.1 Introduction

So far, all previous chapters have assumed that data in DEA are known exactly. In the DEA literature, there are models for dealing with rank data and interval data. For example, some outputs and inputs may be only known as in forms of bounded or interval data, ordinal data, and ratio bounded data. Cook et al. (1993, 1996) were the first who developed a modified DEA structure where the inputs and outputs are represented as rank positions in an ordinal, rather than numerical sense.

If we incorporate such imprecise data information directly into the standard linear DEA model, the resulting DEA model is a non-linear and non-convex program. Such a DEA model is called imprecise DEA (IDEA) in Cooper et al. (1999) who discuss how to deal with bounded data and weak ordinal data and provide a unified IDEA model when weight restrictions are also present[1]. In a similar work, Kim et al. (1999) discuss how to deal with bounded data, (strong and weak) ordinal data, and ratio bounded data.

As shown in Cook and Zhu (2006), the IDEA approach of Kim et al. (1999) and Cooper et al. (1999) approach is actually a direct result of Cook et al. (1993, 1996) with respect to the use of variable alternations.

Zhu (2003a, 2004) on the other hand shows that the non-linear IDEA can be solved in the standard linear DEA model via identifying a set of exact data from the imprecise input and output data. This approach allows us to use all existing DEA techniques to analyze the performance of DMUs and additional evaluation information (e.g., performance benchmarks, paths for efficiency improvement, and returns to scale (RTS) classification) can be obtained.

Part of the materials is based upon Chen, Y. and Zhu, J., Interval and ordinal data, in Modeling Data Irregularities and Structural Complexities in Data Envelopment Analysis, Chapter 3, 35–62, eds J. Zhu and W.D. Cook, Springer, Boston, 2007.

---

[1] Zhu (2003a) shows that such weight restrictions are redundant when ordinal and ratio bounded data are present. This can substantially reduce the computation burden.

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking,* International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_18,

Chen (2007) calls the existing IDEA approaches multiplier IDEA (MIDEA) because these approaches are based upon the DEA multiplier models. Chen (2007) also shows that IDEA models can be built on the envelopment DEA models. That is, the interval data and ordinal data can be introduced directly into the envelopment DEA model. We call the resulting DEA approach as envelopment IDEA (EIDEA). It is shown that EIDEA yields the worst scores whereas the MIDEA yields the best efficiency scores. Using the techniques developed in Zhu (2003a, 2004), the EIDEA can also be converted into linear DEA models.

Despotis and Smirlis (2002) also develop a general structure to convert interval data in dealing with the imprecise data in DEA. Kao and Liu (2000) treat the interval data as fuzzy DEA approach.

The current chapter will only focus on the approach Zhu (2003a, b, 2004) and Chen (2007) where identification of a set of exact data allows us to use the existing standard DEA codes. For other approaches to interval data and ordinal data, the interested reader is referred to Cook and Zhu (2006).

The remainder of this chapter is organized as follows. The next section presents the multiplier and primal DEA models with some specific forms of imprecise data. We then present the Multiplier IDEA (MIDEA) approach. We show how to convert the MIDEA model into linear programs. We then present the Envelopment IDEA (EIDEA) approach described in Chen (2007)[2]. Conclusions are given in the last section.

18.2 Imprecise Data

Suppose we have a set of n peer DMUs, $\{\text{DMU}_j: j=1, 2, \ldots, n\}$, which produce multiple outputs y_{rj}, $(r=1, 2, \ldots, s)$, by utilizing multiple inputs x_{ij}, $(i=1, 2, \ldots, m)$. When a DMU_o is under evaluation by the CRS multiplier model, we have

$$
\begin{aligned}
&\text{Maximize } \pi_o = \sum_{r=1}^{s} \mu_r y_{ro} \\
&\text{subject to} \\
&\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{m} \omega_i x_{ij} \le 0 \ \forall \ j \\
&\sum_{i=1}^{m} \omega_i x_{io} = 1 \\
&\mu_r, \omega_i \ge 0 \quad \forall \ r, i
\end{aligned}
\tag{18.1}
$$

[2] In Chen (2007), envelopment IDEA (EIDEA) is called primal IDEA (PIDEA).

The dual program to (18.1)—the envelopment DEA model can be written as

$$\begin{aligned}
&\theta_o^* = \min \theta_o \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le \theta_o x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} \qquad r = 1,2,\ldots,s; \\
&\lambda_j \ge 0 \qquad j = 1,2,\ldots,n.
\end{aligned} \tag{18.2}$$

In the discussion to follow, we suppose the imprecise data take the forms of bounded data, ordinal data, and ratio bounded data as follows:

Interval or Bounded data

$$\underline{y}_{rj} \le y_{rj} \le \overline{y}_{rj} \text{ and } \underline{x}_{ij} \le x_{ij} \le \overline{x}_{ij} \text{ for } r \in \boldsymbol{BO}, i \in \boldsymbol{BI} \tag{18.3}$$

where $\underline{y}_{rj}$ and $\underline{x}_{ij}$ are the lower bounds and $\overline{y}_{rj}$ and $\overline{x}_{ij}$ are the upper bounds, and ***BO*** and ***BI*** represent the associated sets for bounded outputs and bounded inputs respectively.

Weak Ordinal Data

$$y_{rj} \le y_{rk} \text{ and } x_{ij} \le x_{ik} \text{ for } j \ne k, r \in \boldsymbol{DO}, i \in \boldsymbol{DI}$$

or to simplify the presentation,

$$y_{r1} \le y_{r2} \le \ldots \le y_{rk} \le \ldots \le y_{rn} (r \in \boldsymbol{DO}) \tag{18.4}$$

$$x_{i1} \le x_{i2} \le \ldots \le x_{ik} \le \ldots \le x_{in} (i \in \boldsymbol{DI}) \tag{18.5}$$

where ***DO*** and ***DI*** represent the associated sets for weak ordinal outputs and inputs respectively.

Strong Ordinal Data

$$y_{r1} < y_{r2} \ldots < y_{rk} \ldots < y_{rn} \ (r \in \boldsymbol{SO}) \tag{18.6}$$

$$x_{i1} < x_{i2} \ldots x_{ik} \ldots x_{in} \ (i \in \boldsymbol{SI}) \tag{18.7}$$

where ***DO*** and ***DI*** represent the associated sets for strong ordinal outputs and inputs respectively.

Ratio Bounded Data

$$L_{rj} \le \frac{y_{rj}}{y_{rj_o}} \le U_{rj} (j \ne j_o) \ \ (r \in \boldsymbol{RO}) \tag{18.8}$$

$$G_{ij} \le \frac{x_{ij}}{x_{ij_o}} \le H_{ij} (j \ne j_o) \ \ (i \in \boldsymbol{RI}) \tag{18.9}$$

where L_{rj} and G_{ij} represent the lower bounds, and U_{rj} and H_{ij} represent the upper bounds. ***RO*** and ***RI*** represent the associated sets for ratio bounded outputs and inputs respectively.

If we incorporate (18.3–18.9) into model (18.1), we have the multiplier IDEA (MIDEA) model

$$\begin{aligned}
&\max \pi_o = \sum_{r=1}^{s} \mu_r y_{ro} \\
&s.t. \sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{m} \omega_i x_{ij} \le 0 \\
&\sum_{i=1}^{m} \omega_i x_{io} = 1 \\
&(x_{ij}) \in \Theta_i^- \\
&(y_{rj}) \in \Theta_r^+ \\
&\mu_r, \omega_i \ge 0
\end{aligned} \tag{18.10}$$

where $(x_{ij}) \in \Theta_i^-$ and $(y_{rj}) \in \Theta_r^+$ represent any of or all of (18.3–18.9).

If we incorporate (18.3–18.9) into model (18.2), we then have the envelopment IDEA (EIDEA) model. Obviously, model (18.10) is non-linear and non-convex, because some of the outputs and inputs become unknown decision variables. We will discuss how to solve these two non-linear IDEA models.

Cooper et al. (1999) and Kim et al. (1999) show that model (18.10) can be converted into the following linear programming problem when scale transformations and variable alternations are applied:

$$\begin{aligned}
&\max \quad \sum_r Y_{r0} \\
&\text{subject to} \\
&\sum_r Y_{rj} - \sum_i X_{ij} \le 0 \forall j \\
&\sum_i X_{i0} = 1 \\
&(X_{ij}) \in \mathrm{H}_i^- \quad \forall i \\
&(Y_{rj}) \in \mathrm{H}_r^+ \quad \forall r \\
&X_{ij}, Y_{rj} \ge 0 \quad \forall i, r
\end{aligned} \tag{18.11}$$

where $X_{ij} = \hat{x}_{ij}\hat{\omega}_i$, $Y_{rj} = \hat{y}_{rj}\hat{\mu}_r$, $\hat{\omega}_i = \omega_i \cdot \max_j\{x_{ij}\}$, $\hat{\mu}_r = \mu_r \cdot \max_j\{y_{rj}\}$, $\hat{x}_{ij} = x_{ij} / \max_j \{x_{ij}\}$, $\hat{y}_{rj} = y_{rj} / \max_j \{y_{rj}\}$, $X^o_{ij} = \hat{x}^o_{ij}\hat{\omega}_i$, $Y^o_{rj} = \hat{y}^o_{rj}\hat{\mu}_r$, $\hat{x}^o_{ij} = \max_j\{\hat{x}_{ij}\}$, and $\hat{y}^o_{rj} = \max_j\{\hat{y}_{rj}\}$. Also, Θ^+_r and Θ^-_i are transformed into H^+_r and H^+_r.

Obviously, the standard (linear) CRS DEA model cannot be used and a set of special computation codes is needed for each evaluation, since a different objective function $\left(\sum Y_{ro}\right)$ and a new constraint $\left(\sum X_{io}\right)$ are present in model (18.11) for each DMU under evaluation. Note also that the number of new variables (Y_{rj} and X_{ij}) increases substantially as the number of DMUs increases.

Zhu (2003a) provides an improvement by only using variable alternations. That is, define $X_{ij} = x_{ij}\omega_i$, $Y_{rj} = y_{rj}\mu_r$ in model (18.1) when imprecise data are present, and the scale transformation is not needed. This simple approach is actually used in Cook et al. (1993). The interested reader is referred to Cook and Zhu (2006) for the detailed discussion and a general framework for dealing with ordinal data.

18.3 Multiplier IDEA (MIDEA): Standard DEA Model Approach

The following theorem provides the theoretical foundation to the approach developed in Zhu (2003a, 2004) when the standard multiplier CRS model (18.1) is used to solve the IDEA model (18.10).

Theorem 1 Suppose Θ^+_r and Θ^-_i are given by (3), then for DMU_o the optimal value to (10) can be achieved at $y_{ro} = \overline{y}_{ro}$ and $x_{io} = \underline{x}_{io}$ for DMU_o and $y_{rj} = \underline{y}_{rj}$ and $x_{ij} = \overline{x}_{ij}$ for DMU_j ($j \neq$ o).

[Proof] See Zhu (2003a)

Theorem 1 is true due to that fact that increases on output values (decreases on input values) for DMU_o under evaluation or (and) decreases on output values (increases on input values) for other DMUs will not deteriorate the efficiency of DMU_o under evaluation by the multiplier DEA model.

18.3.1 Converting the Bounded Data into a Set of Exact Data

Theorem 1 shows that when DMU_o is under evaluation, we can have a set of exact data via setting $y_{ro} = \overline{y}_{ro}$ and $x_{io} = \underline{x}_{io}$ for DMU_o and $y_{ij} = \underline{y}_{rj}$ and $x_{ij} = \overline{x}_{ij}$ for DMU_j ($j \neq$ o) while model (18.10) maintains the efficiency rating for DMU_o. Note that in this case, model (18.10) is no longer a non-linear program, but a (linear) multiplier CRS model

$$
\begin{aligned}
&\pi_o^* = \text{Maximize} \sum_{r\in BO} \mu_r \overline{y}_{ro} + \sum_{r\notin BO} \mu_r y_{ro} \\
&\text{subject to} \\
&\sum_{r\in BO} \mu_r \underline{y}_{rj} + \sum_{r\notin BO} \mu_r y_{rj} - \sum_{i\in BI} \omega_i \overline{x}_{ij} - \sum_{i\notin BI} \omega_i x_{ij} \le 0 \ \forall \ j \ne o \\
&\sum_{r\in BO} \mu_r \overline{y}_{ro} + \sum_{r\notin BO} \mu_r y_{ro} - \sum_{i\in BI} \omega_i \underline{x}_{io} - \sum_{i\notin BI} \omega_i x_{io} \le 0 \\
&\sum_{i\in BO} \omega_i \underline{x}_{io} + \sum_{i\notin BO} \omega_i x_{io} = 1 \\
&\mu_r, \omega_i \ge 0 \ \ \forall \ \ r, \ i
\end{aligned}
\tag{18.12}
$$

where $y_{rj}\,(r \notin \boldsymbol{BO})$, and $x_{ij}\,(i \notin \boldsymbol{BI})$ are exact data.

We can also use the obtained exact data and apply them to the envelopment model (18.2), namely

$$
\begin{aligned}
&\theta_o^* = \min \theta_o \\
&\text{subject to} \\
&\sum_{j\ne o} \lambda_j \overline{x}_{ij} + \lambda_o \underline{x}_{io} \le \theta_o \underline{x}_{io} && i \in BI; \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \le \theta_o x_{io} && i \notin BI; \\
&\sum_{j\ne o} \lambda_j \underline{y}_{rj} + \lambda_o \overline{y}_{ro} \ge \overline{y}_{ro} && r \in BO; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} && r \notin BO; \\
&\lambda_j \ge 0 && j = 1, 2, \ldots, n.
\end{aligned}
\tag{18.13}
$$

18.3.2 Converting the Weak Ordinal Data into a Set of Exact Data

Consider DMU_k. Suppose we solve model (18.10) when Θ_i^- and Θ_r^+ are in forms of (18.4) and (18.5), and obtain a set of optimal solutions y_{rj}^* and x_{ij}^* with the optimal value π_k^*. We have

$$
y_{r1}^* \le y_{r2}^* \le \ldots \le y_{r,k-1}^* \le y_{rk}^* \le y_{r,k+1}^* \le \ldots \le y_{rn}^* (r \in \boldsymbol{DO}) \tag{18.14}
$$

$$
x_{i1}^* \le x_{i2}^* \le \ldots \le x_{i,k-1}^* \le x_{ik}^* \le x_{i,k+1}^* \le \ldots \le x_{in}^* (i \in \boldsymbol{DI}) \tag{18.15}
$$

Note that $\rho y_{rj}^{*}(r \in \boldsymbol{DO})$ and $\rho x_{ij}^{*}(i \in \boldsymbol{DI})$ are also optimal for DMU_k where ρ is a positive constant, because of the units invariant property. Therefore, we can always set $y_{rk}^{*} = x_{ik}^{*} = 1$. Then, we have a set of optimal solutions on weak ordinal outputs and inputs such that (18.14) and (18.15) can be expressed as[3]

$$0 \le y_{r1}^{*} \le y_{r2}^{*} \le \ldots \le y_{r,k-1}^{*} \le y_{rk}^{*}(=1) \le y_{r,k+1}^{*} \le \ldots \le y_{rn}^{*} \le M \quad (r \in \boldsymbol{DO}) \tag{18.16}$$

$$0 \le x_{i1}^{*} \le x_{i2}^{*} \le \ldots \le x_{i,k-1}^{*} \le x_{ik}^{*}(=1) \le x_{i,k+1}^{*} \le \ldots \le x_{in}^{*} \le M \quad (i \in \boldsymbol{DI}) \tag{18.17}$$

where M is very close to $+\infty$[4].

Now, for the outputs and inputs in weak ordinal relations, we set up the following intervals,

$$y_{rj} \in [0,1] \text{ and } x_{ij} \in [0,1] \text{ for } DMU_j (j = 1,\ldots,k-1) \tag{18.18}$$

$$y_{rj} \in [1,M] \text{ and } x_{ij} \in [1,M] \text{ for } DMU_j (j = k+1,\ldots,n) \tag{18.19}$$

Based upon Theorem 1, we know that for $r \in$ ***DO*** and $i \in$ ***DI***, π_k^{*} remains the same and (18.18) and (18.19) are satisfied if $y_{rk} = x_{ik} = 1$ for DMU_k and $y_{rj} = 0$ (lower bound, $\underline{y}_{rj}$), $x_{ij}=1$ (upper bound, $\overline{x}_{ij}$) for DMU_j ($j=1, \ldots, k-1$) and $y_{rj}=1$ (lower bound, $\underline{y}_{rj}$), $x_{ij} = M$ (upper bound, $\overline{x}_{ij}$) for DMU_j ($j=k+1, \ldots, n$)[5].

18.3.3 Numerical Illustration

Consider the numerical example in Table 18.1. Suppose we have one input with exact value (input-1) and one input with interval values (input-2). Output-1 has exact value and output-2 is in ordinal relations. Based on Theorem 1, we use the lower bound of input-2 as the exact input value for each DMU under evaluation and the upper bounds as the exact input values for all other DMUs. For example, for DMU2, we use $x_{22}=0.4$ (lower bound) and, $x_{23}=0.8$, $x_{24}=0.9$, $x_{26}=0.6$, (upper bounds) for DMUs 1, 3, 4, 5 and 6. In addition to the efficiency scores, Table 18.2 presents the slacks and referent DMUs based upon model (18.13).

Note that it is very difficult to retrieve the optimal values on the bounded input (output) if one uses the variable-alternation algorithm. However, based upon

[3] This procedure appears to be unworkable when weight restrictions are present. However, we will see in Theorem 2, such weight restrictions are redundant and should be removed before the analysis. As a result, the current procedure is not affected.

[4] In computation, M does not have to be set equal to a very large number. In the application section in this chapter, M is set equal to 33.

[5] See Chen (2007) for detailed discussion and alternative ways of setting the exact data when weak ordinal relations are present.

Table 18.1 Exact and imprecise data

	Outputs		Inputs	
	Exact	Ordinal[a]	Exact	Bound
	Output-1 (y_{1j})	Output-2 (y_{2j})	Input-1 (x_{1j})	Input-2 (x_{2j})
DMU1	1000	3	200	1
DMU2	3000	4	400	[0.4, 0.7]
DMU3	2500	1	250	[0.6,0.8]
DMU4	1300	2	300	[0.7, 0.9]
DMU5	900	5	150	1
DMU6	1500	6	250	[0.5,0.6]

[a] a rank of "1" indicates a DMU has the lowest rank

Table 18.2 MIDEA results when bounded data are present

DMU	Efficiency score	slack	Referent DMU
	θ_o^*	Judgment	λ_j^*
1	0.5	0.18	$\lambda_3^* = 0.4$
2	1	0	$\lambda_2^* = 1$
3	1	0	$\lambda_3^* = 1$
4	0.5095	0	$\lambda_2^* = 0.2284, \quad \lambda_3^* = 0.2459$
5	0.6	0.312	$\lambda_3^* = 0.36$
6	0.7565	0	$\lambda_2^* = 0.3913, \quad \lambda_3^* = 0.1304$

Model (18.13) is used with two inputs of cost and judgment and one output of revenue. The ordinal output of satisfaction is not included in calculations

Theorem 1 and the recent development on sensitivity analysis by Zhu (2001), we can determine the range of multiple optimal solutions on bounded data for *DMUo* (and other DMUs). That is, we calculate the following linear program (Zhu 2001).

$$
\begin{aligned}
&\tilde{\theta}_o^* = \min \tilde{\theta}_o \\
&\text{subject to} \\
&\sum_{j \neq o} \lambda_j \overline{x}_{ij} \leq \tilde{\theta}_o \underline{x}_{io} && i \in BI; \\
&\sum_{j \neq o} \lambda_j x_{ij} \leq x_{io} && i \notin BI \ (exact\ data); \\
&\sum_{j \neq o} \lambda_j \underline{y}_{rj} \geq \overline{y}_{ro} && r \in BO; \\
&\sum_{j \neq o} \lambda_j y_{rj} \geq y_{ro} && r \notin BO \ (exact\ data); \\
&\lambda_j \geq 0 && j \neq o
\end{aligned}
\tag{18.20}
$$

Table 18.3 Alternative optimal solutions

	DMU under evaluation					
	DMU1	DMU2	DMU3	DMU4	DMU5	DMU6
DMU2	[0.4,0.7]	[0.4,0.7]	[0.4,0.7]	0.7	[0.4,0.7]	0.7
DMU3	[0.6,0.8]	[0.6,0.8]	[0.6,0.8]	0.8	[0.6,0.8]	0.8
DMU4	[0.7, 0.9]	[0.7, 0.9]	[0.7, 0.9]	0.7	[0.7, 0.9]	[0.7, 0.9]
DMU6	[0.5,0.6]	[0.5,0.6]	[0.5,0.6]	[0.5,0.6]	[0.5,0.6]	0.5

Consider DMU2 in Table 18.1, we have (assuming that we only have two inputs and one output of output-1)

$$\begin{aligned}
&\min \tilde{\theta}_2 \\
&\text{subject to} \\
&200\lambda_1 + 250\lambda_3 + 300\lambda_4 + 150\lambda_5 + 250\lambda_6 \le 400 && \text{(Input-1)} \\
&1\lambda_1 + 10.8\lambda_3 + 0.9\lambda_4 + 1\lambda_5 + 0.6\lambda_6 \le 0.4\tilde{\theta}_2 && \text{(Input-2)} \\
&1000\lambda_1 + 2500\lambda_3 + 1300\lambda_4 + 900\lambda_5 + 1500\lambda_6 \ge 3000 && \text{(Output-1)} \\
&\lambda_j \ge 0, j = 1,3,4,5,6.
\end{aligned}$$

Now, suppose the lower bound of input-2 for DMU2 can be increased by σ, and the upper bounds of input-2 for other DMUs can be decreased by σ'. Based on Zhu (2001), if σ $\sigma' = \tilde{\theta}_2^* = 2.4$, then DMU2 remains efficient ($\pi_1^* = \theta_1^* = 1$). Thus, the efficiency stability region for input-2 is larger than the range of [0.4, 0.7] for DMU2. When DMU2 is under evaluation, any DMU2's input-2 value within the range of [0.4, 0.7] is an optimal solution such that $\pi_1^* = \theta_1^* = 1$ remains true. Table 18.3 reports the optimal values of input-2 by using model (18.20). As a result, the (multiple) optimal solutions on the bounded input can be retrieved for the variable-alternation algorithm.

We next convert the ordinal data into a set of exact data using three DMUs, namely, DMU1, DMU3, and DMU6, in Table 18.1. We have (i) DMU3 has the lowest rank. We use "1" as the output-2 value for all DMUs; (ii) DMU6 has the highest rank. We use "1" for the output-2 value for DMU6 and use 0 for other DMUs; and (iii) DMU1 is ranked third. We use "1" for the output-2 value for DMUs 1, 2, 5, and 6 and use "0" for other DMUs.

Table 18.4 reports the set of exact data for the satisfaction output across all five DMUs when a specific DMU is under evaluation. Table 18.5 reports the results from model (18.13) when we use the exact data from Table 18.4. It can be seen that both MIDEA approaches yield the identical efficiency scores. The standard DEA approach indicates that DMUs 1 and 4 have non-zero slack values.

Table 18.4 Converting ordinal data into exact data

	DMU under evaluation					
Satisfaction	DMU1	DMU2	DMU3	DMU4	DMU5	DMU6
y_{21}	1	0	1	1	0	0
y_{22}	1	1	1	1	0	0
y_{23}	0	0	1	0	0	0
y_{24}	0	0	1	1	0	0
y_{25}	1	1	1	1	1	0
y_{26}	1	1	1	1	1	1

Table 18.5 MIDEA results

	Efficiency score	Slack		Referent DMU
DMU	θ_o^*	Revenue	Cost	λ_j^*
1	0.8889	66.6667	0	$\lambda_5^* = 0.7222$, $\lambda_6^* = 0.2778$
2	1	0	0	$\lambda_2^* = 1$
3	1	0	0	$\lambda_3^* = 1$
4	0.8571	200	7.1429	$\lambda_6^* = 1$
5	1	0	0	$\lambda_5^* = 1$
6	1	0	0	$\lambda_6^* = 1$

18.3.4 Converting the Strong Ordinal Data and Ratio Bounded Data into a Set of Exact Data

Recall that $y_{rk} - y_{r,k-1} \geq \chi$ and $x_{ik} - x_{i,k-1} \geq \chi$ are not valid forms to represent strict rankings under model (18.10). We propose the following correct and valid modifications when DMU_k is under evaluation

$$y_{r,k+1} \geq \chi_r y_{r,k} (\chi_r > 1) \text{ and } _{ik} \geq \eta_i x_{i,k-1} (\eta_i > 1) \tag{18.21}$$

Since (18.21) is units invariant, it allows scale transformations. $y_{r,k+1} \geq \chi_r y_{r,k}$ and $x_{ik} \geq \eta_i x_{i,k-1}$ in (18.10) are equivalent to $Y_{r,k+1} \geq \chi_r Y_{r,k}$ and $X_{ik} \geq \eta_i X_{i,k-1}$ in (18.11) respectively, if the scale-transformation and variable-alternation based approach is used. Note that (18.21) may allow all data equal to zero. However, the proposed method of finding exact data does not allow such cases to occur.

Based upon the discussion on converting weak ordinal data into a set of exact data (see also Zhu 2003b), we can set $y_{rk} = 1$ and $x_{ik} = 1$, and further we have[6]

[6] We can easily select a set of exact data for y_{rj} ($j=1, \ldots, k$) and y_{ij} ($j=k+1, \ldots, n$). For example, we can set these y_{rj} very close to zero and meanwhile (18.22) is satisfied.

$$\begin{cases} y_{rj} = \chi_r^{j-k} & for \quad DMU_j \ (j = k+1, \ldots, n) \\ x_{ij} = \eta_i^{j-k} & for \quad DMU_j (j = 1, \ldots, k-1) \end{cases}$$

when χ_r and η_i are given.

Furthermore, parts of (18.8) and (18.9) actually represent strong ordinal relations when $L_{rj} = \chi_r$ or $1/H_{ij} = \eta_i$.

In Kim et al. (1999), $L_4 = 1.07$ for the forth output and $L_5 = 8.04$ for the fifth output. Thus, if only strong ordinal relations are assumed, we have

$$y_{r,k+1} \geq L_r y_{r,k} \text{ or } \frac{y_{r,k+1}}{y_{r,k}} \geq L_r \ (r = 4, 5) \tag{18.22}$$

When DMU_k is under evaluation, we let $y_{rk} = 1$ and we have a set of exact data consisting of (i) $y_{rj} = L_r^{j-k} (r = 4, 5; j = k+1, \ldots, n)$ and (ii) $y_{rj} = \varepsilon_j \approx 0$ such that $y_{r,j+1} \geq L_r y_{rj} \ (j = 1, \ldots, k-1)$. The fourth and fifth column of Table 18.2 present a set of exact data on y_4 and y_5 when DMU29 is under evaluation and strong ordinal relations in (18.22) are imposed.

Moreover, note that if we assume strong ordinal relations as in (18.22), too much flexibility may still be allowed in $y_{rj} = \varepsilon_j \approx 0 \ (j = 1, \ldots, k-1)$. Therefore, we introduce

$$y_{r,k} \leq U_r y_{r,k-1} \quad or \quad \frac{y_{r,k}}{y_{r,k-1}} \leq U_r \ (r = 4, 5) \tag{18.23}$$

to further restrict the values on y_{rj} ($j = 1, \ldots, k-1$).

Ratio bounded data (18.22) and (18.23) can also be converted into a set of exact data via the following two steps.

Step 1: Set $y_{rj_o} = 1$ and $x_{ij_o} = 1$

Step 2: We have bounded data for other DMUs: $L_{rj} \leq y_{rj} \leq U_{rj}$ and $G_{ij} \leq x_{ij} \leq H_{ij} \ (j \neq j_o)$ which can further be converted into exact data.

Step 1 is valid is because there are no other constraints on y_{rj_o} and x_{ij_o}. However, if y_{rj_o} and x_{ij_o} can take values within certain ranges as given in (18.3), we have two cases associated with step 1. (Case 1) If DMU_{j_o} is under evaluation, we set $y_{rj_o} = \overline{y}_{rj_o}$ and $x_{ij_o} = \underline{x}_{ij_o}$. (Case 2) If DMU_{j_o} is not under evaluation, we set $y_{rj_o} = \underline{y}_{rj_o}$ and $x_{ij_o} = \overline{x}_{ij_o}$.

18.4 Treatment of Weight Restrictions

The above discussion and the proposed method assume that weight restrictions related to imprecise data in forms (18.4–18.9) are not present. The following theorem shows that adding weight restrictions related to imprecise outputs and inputs in forms (18.4–18.9) does not change the efficiency ratings. Thus, these particular weight restrictions are redundant and can be removed before solving the model (18.10). As a result, the standard DEA model based approach can be used.

Theorem 2 Suppose the *f*th input and the *d*th output are imprecise data given by (18.4–18.9), and π_o^* is the optimal value to (18.10), then π_o^* remains unchanged if the following weight restrictions related to the *f*th input and *d*th output are added into model (18.10)

$$\alpha_f \le \frac{f(\mu_r, r \notin DO \cap RO)}{\mu_f} \le \beta_f \quad (f \in DO \text{ or } RO) \tag{18.24}$$

$$A_d \le \frac{f(\omega_i, i \notin DI \cap RI)}{\omega_d} \le B_d \quad (d \in DI \text{ or } RI) \tag{18.25}$$

where $f(\bullet)$ is a function on μ_r (ω_i) related to exact outputs (inputs).

[Proof]: See Zhu (2003a).[7]

Theorem 2 indicates that the optimal value to model (18.10) is equal to the optimal value to the following model (model (18.10) with weight restrictions (18.24) and (18.25))

$$\begin{aligned}
&\max \pi_o = \sum_{r=1}^{s} \mu_r y_{ro} \\
&s.t. \sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{m} \omega_i x_{ij} \le 0 \\
&\sum_{i=1}^{m} \omega_i x_{io} = 1 \\
&(x_{ij}) \in \Theta_i^- \\
&(y_{rj}) \in \Theta_r^+ \\
&\alpha_f \le \frac{f(\mu_r, r \notin DO \cap RO)}{\mu_f} \le \beta_f \qquad f \in DO \ \ or \ \ RO \\
&A_d \le \frac{f(\omega_i, i \notin DI \cap RI)}{\omega_d} \le B_d \qquad d \in DI \ \ or \ \ RI \\
&\mu_r, \omega_i \ge 0 \quad \forall r, i
\end{aligned} \tag{18.26}$$

In other words, the same efficiency ratings can be obtained by either solving model (18.10) or model (18.25). If one obtains the efficiency ratings under model (18.10), the same efficiency ratings are obtained for model (18.26). The method developed in the previous sections can be applied to solve model (18.10) without affecting the efficiency ratings. As a result, model (18.26) is solved indirectly.

[7] This theorem is true because (18.4–18.9) are units invariant.

Table 18.6 Four DMUs and their MIDEA scores

	Output (y)	Input 1 (x_1)	Input 2 (x_2)	MIDEA score
DMU1	1	[2,4]	4	1
DMU2	1	[3,5]	3	1
DMU3	1	7	2	1
DMU4	1	[1,3]	6	1
DMU5	1	[6,8]	7	0.6316

Although we cannot set exact data for partial data in model (18.26), Theorem 2 provides an alternative where the efficiency ratings under (18.26) are obtained via setting exact data in model (18.10) and solving model (18.10). i.e., the proposed approach is not affected by the presence of weight restrictions, since solving model (18.10) with proposed approach is equivalent to solving model (18.26) directly[8]. To obtain the efficiency ratings under model (18.26), we only need to solve model (18.10) via setting exact data in model (18.10), because the weight restrictions represented by (18.24) and (18.25) are redundant.

18.5 Envelopment IDEA (EIDEA)

Chen (2007) points out that when the IDEA is developed based upon the envelopment DEA model, e.g., model (18.2), we get different efficiency results. When we assume that all output and input values are exact, models (18.1) and (18.2) yield the same efficiency score for a specific DMU under evaluation. However, the presence of imprecise data invalidates the linear duality between models (18.1) and (18.2) and consequently, model (18.1) is not equivalent to model (18.2). The EIDEA yields the worst efficiency scores. The invalidation of linear duality leads to an efficiency gap.

Consider five DMUs as shown in Table 18.6 with two inputs and a single output of unity. Only DMU3 has exact data on the first input. The last column shows the efficiency scores based upon MIDEA.

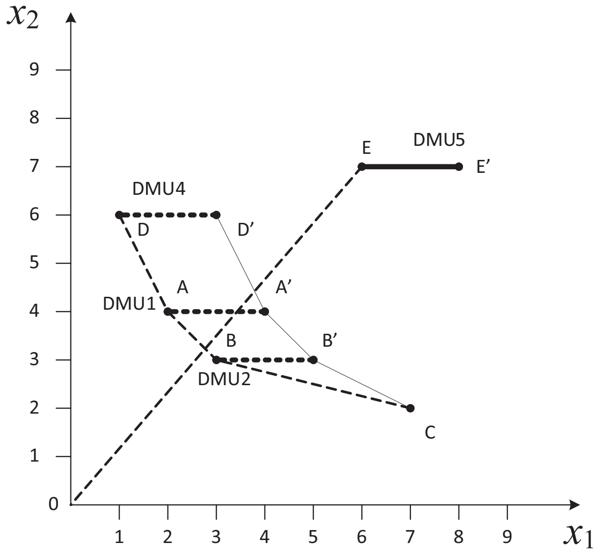
Figure 18.1 plots the five DMUs. Because of the bounded data, DMU1, DMU2, and DMU4 are represented by AA′, BB′, and DD′ respectively[9]. The (imprecise) efficient frontier is represented by the area bounded by DD′, D′A′, A′B′, B′C, CB, BA, and AD.

When DMU5 is under evaluation by the MIDEA, DMU5 is replaced by point E and D′A′B′C is used as the exact efficient frontier. For DMU5, A′ and D′ are referent DMUs which can be achieved via setting $x_{11} = 4$ (upper bound) for DMU1 and

[8] Note that the following two cases are different. Case I: Setting the variables by exact data in model (18.10) and then finding the efficiency ratings. (This provides the same efficiency ratings as those obtained from solving model (18.26) directly.) Case II: Setting the variables by a set of exact data in model (18.26) and finding the efficiency scores under model (18.26) (This leads to a different problem.) Case I represents the objective of the current study or the objective of solving model (18.26).

[9] DMU5 is represented by a line segment.

Fig. 18.1 DEA frontier with interval data



x_{14}=3 (upper bound) for DMU4 and then using the convex combination of A′ and D′ with $\lambda_1^* = 0.7895, \lambda_4^* = 0.2105$.

When DMU5 is under evaluation by the EIDEA, DMU5 is evaluated against DABC rather than D′ABC. Consequently, the efficiency of DMU5 decreases. Upper bound on the first input of DMU5 and lower bounds on the first input of other DMUs are used as the exact data when we evaluate DMU5 using EIDEA.

Theorem 3 Suppose for DMU_o, θ_o^* is the optimal value to (18.2) when (some) outputs and inputs are only known to be within specific bounds given by (18.3). This θ_o^* can be achieved with

I. $x_{ij} = x_{ij}$ for DMU_j ($j \neq$ o);
II. $y_{rj} = \overline{y}_{rj}$ for DMU_j ($j \neq$ o);
III. $y_{ro} = y$ for DMU_o;
IV. $x_{io} = \overline{x}_{io}$ for DMU_o.

[Proof] See Chen (2007).

Theorem 3 is true due to the fact that input decreases/output increases in DMU_o or (and) input decreases/output increases in other DMUs will deteriorate the efficiency of DMU_o.

Theorem 4 indicating that EIDEA can be executed by setting the lower output bounds and upper input bounds as the exact output and input values for DMU_o and by setting upper output bounds and lower input bounds as the exact output and input values for the remaining DMUs.

Recall that in MIDEA, ordinal data and ratio bound data are converted into a set of exact data via the bounded data. By the same fashion, ordinal data and ratio bound data can be converted into a set of exact data under EIDEA.

18.6 Conclusions

The current chapter presents how the standard linear DEA models can be used to deal with interval or bounded data or ordinal data. Although the current chapter discusses specific forms of imprecise data[10], the results are true for any types of imprecise data (see Theorem 1 in Zhu 2004).

The IDEA approach using the standard DEA model indicates that one has to decide whether the multiplier or envelopment DEA model will be used to deal with the imprecise data. The multiplier IDEA (MIDEA) yields the best efficiency scores whereas the envelopment IDEA (EIDEA) yields the worst efficiency scores. We should note that the MIDEA can also yield the same worst EIDEA efficiency scores if we set the exact data in a reversed direction.

The current chapter discusses IDEA procedure based upon the CRS model. Similar discussion can be developed based upon other DEA models.

We finally provide the following theorem to show that for inefficient DMUs, their multiplier DEA efficiency is always achieved at the bounds for interval data.

Theorem 4 For DMU_o, if $\pi_o^* < 1$ in (18.1), then the optimality must be achieved at $x_{io} = \underline{x}_{io}$ and $y_{ro} = \overline{y}_{ro}$ for DMU_o.

[Proof] Note that by defining $\omega_i = \mathrm{t}v_i$, $\mu_r = \mathrm{t}u_r$, and $\mathrm{t} = \left(\sum v_i x_{io}\right)^{-1}$, model (18.1) is equivalent to $\pi_o^* = \max \dfrac{\sum_{r=1}^{s} u_r y_{ro}}{\sum_{i=1}^{m} v_i x_{io}}$ subject to $\dfrac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \le 1 \quad \forall\ j$ and $u_r, v_i \ge \varepsilon$.

Next, suppose the optimality is achieved at u_r^*, v_i^*, y_{rj}^* $(r \in \boldsymbol{BO})$ and x_{ij}^* $(i \in \boldsymbol{BI})$ with $\pi_o^* = \dfrac{\sum_{r\in BO} u_r^* y_{ro}^* + \sum_{r\notin BO} u_r^* y_{ro}}{\sum_{i\in BI} v_i^* x_{io}^* + \sum_{i\notin BI} v_i^* x_{io}} < 1$ and $\underline{y}_{ro} \le y_{ro}^* < \overline{y}_{ro}$ $(r \in \boldsymbol{BO})$ and

$\underline{x}_{io} < x_{io}^* \le \overline{x}_{io}$ $(i \in \boldsymbol{BI})$.

Obviously, when $x_{io} = \underline{x}_{io}$ and $y_{ro} = \overline{y}_{ro}$, we have

[10] We should note that the specific forms of imprecise data are probably the only imprecise data types that will occur in real application.

$\pi_o^* = \dfrac{\sum_{r\in BO} u_r^* y_{ro}^* + \sum_{r\notin BO} u_r^* y_{ro}}{\sum_{i\in BI} v_i^* x_{io}^* + \sum_{i\notin BI} v_i^* x_{io}} < \dfrac{\sum_{r\in BO} u_r^* \overline{y}_{ro}^* + \sum_{r\notin BO} u_r^* y_{ro}}{\sum_{i\in BI} v_i^* \underline{x}_{io}^* + \sum_{i\notin BI} v_i^* x_{io}}$. A contradiction[11]. This shows that the optimality must be achieved at $x_{io} = \underline{x}_{io}$ and $y_{ro} = \overline{y}_{ro}$ for DMU_o.

Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_18) contains supplementary material, which is available to authorized users.

References

Chen, Y. (2007). Imprecise DEA—Envelopment and multiplier model approaches. *Asian Pacific Journal of Operations Research, 24*(2), 279–291.

Cook, W. D., & Zhu, J. (2006). Rank order data in DEA: A general framework. *European Journal of Operational Research, 174*(2), 1021–1038.

Cook, W. D., Kress, M., & Seiford, L. M. (1993). On the use of ordinal data in data envelopment analysis. *Journal of Operational Research Society, 44,* 133–140.

Cook, W. D., Kress, M., & Seiford, L. M. (1996). Data envelopment analysis in the presence of both quantitative and qualitative factors. *Journal of Operational Research Society, 47,* 945–953.

Cooper, W. W., Park, K.S., & Yu, G. (1999). IDEA and AR-IDEA: Models for dealing with imprecise data in DEA. *Management Science, 45,* 597–607.

Despotis, D. K., & Smirlis, Y. G. (2002). Data envelopment analysis with imprecise data. *European Journal of Operational Research, 140,* 24–36.

Kao, C., & Liu, S. T. (2000). Fuzzy efficiency measures in data envelopment analysis. *Fuzzy Sets and Systems, 113,* 427–437.

Kim, S. H., Park, C. G.& Park, K. S. (1999). An application of data envelopment analysis in telephone offices evaluation with partial data. *Computers Operations Research,* 26, 59–72.

Zhu, J. (2001). Super-efficiency and DEA sensitivity analysis. *European Journal of Operational Research, 129*(2), 443–455.

Zhu, J. (2003a). Imprecise data envelopment analysis (IDEA): A review and improvement with an application. *European Journal of Operational Research, 144*(3), 513–529.

Zhu, J. (2003b). Efficiency evaluation with strong ordinal input and output measures. *European Journal of Operational Research, 146*(3), 477–485.

Zhu, J. (2004). Imprecise DEA via standard linear DEA models with a revisit to a Korean mobile telecommunication company. *Operations Research, 52*(2), 323–329.

[11] If , $\dfrac{\sum_{r\in BO} u_r^* \overline{y}_{ro} \sum_{r\notin BO} u_r^* y_{ro}}{\sum_{i\in BI} v_i^* \underline{x}_{io} \sum_{i\notin BI} v_i^* x_{io}} = \tilde{h}_o > 1$ we can always redefine the weights by dividing each u_r^* by $\tilde{h}_o$. As a result, the new optimal value is equal to one, and all $\dfrac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}}$ decrease and are not greater than one. This will still give a contradiction. Thus, the theorem is true.

Chapter 19
DEAFrontier Software

19.1 Introduction

Previous chapters include how to use DEAFrontier software to solve DEA models discussed. This version of DEAFrontier software requires Excel 2007–2013 9 (under Windows) and can solve up to 50 DMUs with unlimited number of inputs and outputs (subject to the capacity of the standard Excel Solver). To install the software, copy the file "DEAFrontier.xlam" to your hard drive. Please visit www.deafrontier.net for software support.

Open the file "DEAFrontier.xlam" to load DEAFrontier. To locate the DEA Menu, the user must select the Add-Ins tab and navigate to the DEA menu option as shown in Fig. 19.1

In this chapter, we present how to use DEAFrotnier to solve DEA models that are not discussed in the previous chapters. They include DEA bootstrapping, free disposal hull (FDH), and Malmquist approach.

We finally present a list DEAFrontier models along with the chapters that discuss the related DEA models.

19.2 DEA Bootstrapping

The data sheet format should follow the standard DEA's format described in Chap. 2. The approach is based upon the standard DEA envelopment/multiplier model.

There are two DEA bootstrapping approaches. The SW-algorithm is based upon Simar and Wilson (1998). The LT-algorithm is based upon Lothgren and Tambour (1999) (see Fig. 19.2)

The results are reported in sheet "Bootstrapping_Results" as shown in Fig. 19.3.

"Bootstrapping_Results" reports the original DEA efficiency scores in column C. Column D reports the Bias that is calculated based upon the original DEA

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*,
International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9_19,

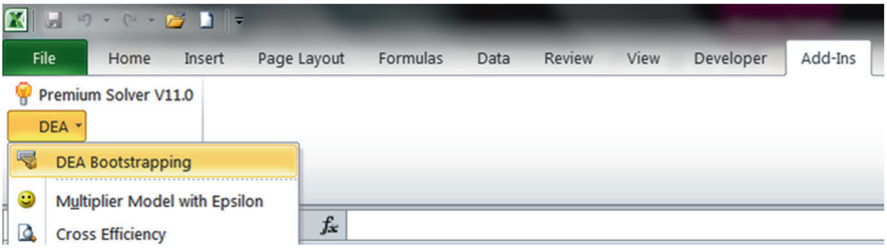
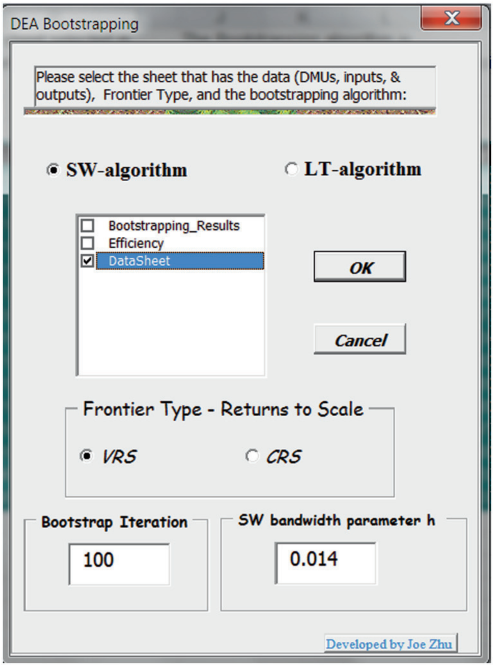


Fig. 19.1 DEAFrontier menu

Fig. 19.2 DEA bootstrapping



efficiency scores and the scores from the bootstrapping runs. The confidence interval can be transformed into a bias-correct interval by subtracting 2*Bias from the interval bounds.

19.3 Free Disposal Hull (FDH)

The free disposal hull (FDH) models are first formulated by Deprins et al. (1984). The input-oriented FDH model can be written as

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	**Inputs**		**Outputs**		0.05		Data sheet			Bootstrapping algorithm:					
2	Assets		Revenue				Data			Simar & Wilson (1998), Management Science, Vol. 44, No. 1					
3	Equity		Profit												
4	Employees														
5			VRS												
6	DMU No.	DMU Name	Efficiency	Bias		Confidence	Interval	Bias-corrected interval can be obtained by subtracting 2*Bias from the interval bounds		[illegible]	Run2	Run3	Run4	Run5	Run6
7	1	Mitsubishi	1	0.231149		1.06145	1.85377			[illegible]	1.06587	1.79068	1.09538	1.07921	1.05784
8	2	Mitsui	1	0.197753		1.06056	1.68882			[illegible]	1.86353	1.05502	1.10803	1.07974	1.06733
9	3	Itochu	1	0.199472		1.04712	1.73079			[illegible]	1.04065	1.06971	1.10426	1.07828	1.05502
10	4	General Motors	1	0.480249		1.05619	1.99243			1.07073	1.04429	1.96931	1.76781	1.93455	2.01134
11	5	Sumitomo	1	0.149099		1.05107	1.42691			1.43531	1.03885	1.08059	1.10928	1.09367	1.06600
12	6	Marubeni	1	0.139867		1.04874	1.33974			1.22500	1.03422	1.06649	1.09259	1.08728	1.06759
13	7	Ford Motor	0.737556	0.079905		0.77927	0.93029			0.79227	0.77252	0.78962	0.80232	0.79020	0.78753
14	8	Totota Motor	0.603245	0.06102		0.63568	0.73944			0.64517	0.63094	0.64151	0.65736	0.65775	0.79358
15	9	Exxon	1	0.196554		1.06147	1.66505			1.06993	1.06491	1.05865	1.08758	1.09957	1.85432
16	10	Royal Dutch/Shell Group	1	0.484923		1.05987	2.26667			1.07254	1.05348	1.35281	2.31089	1.10536	1.75146
17	11	Wal-Mart	1	0.22511		1.06752	1.80898			1.07114	1.70589	1.07003	1.89334	1.07661	1.06547
18	12	Hitachi	0.557596	0.048535		0.58360	0.65164			0.59900	0.57894	0.59460	0.68659	0.60524	0.58930
19	13	Nippon Life Insurance	1	0.405343		1.04723	2.08329			2.20203	1.03682	1.10139	1.11806	1.68873	1.05994
20	14	Nippon Telegraph & Telephone	0.470611	0.041547		0.49684	0.53493			0.52898	0.49257	0.50205	0.53980	0.50985	0.50348
21	15	AT&T	0.533544	0.055099		0.55963	0.65180			0.57615	0.55616	0.57187	0.62736	0.58141	0.56387

Fig. 19.3 DEA bootstrapping results

$$\min \theta^{FDH}$$

subject to

$$\sum_{j=1}^{n} \lambda_j x_{ij} \leq \theta^{FDH} x_{io} \qquad i = 1, 2, \ldots, m;$$

$$\sum_{j=1}^{n} \lambda_j y_{rj} \geq y_{ro} \qquad r = 1, 2, \ldots, s;$$

$$\sum_{j=1}^{n} \lambda_j = 1 \qquad \lambda_j \text{ binary}$$

The output oriented FDH model can be written as

$$\max \phi^{FDH}$$

subject to

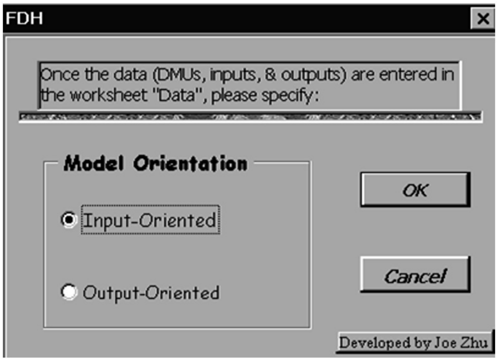
$$\sum_{j=1}^{n} \lambda_j x_{ij} \leq x_{io} \qquad i = 1, 2, \ldots, m;$$

$$\sum_{j=1}^{n} \lambda_j y_{rj} \geq \phi^{FDH}_{y_{ro}} \qquad r = 1, 2, \ldots, s;$$

$$\sum_{j=1}^{n} \lambda_j = 1 \qquad \lambda_j \text{ binary}$$

To run the above two FDH models, select the FDH menu item. You will be asked to select the model orientation (see Fig. 19.4). The results are reported in the "FDH" sheet.

Fig. 19.4 FDH models



19.4 Mamquist Approach

Malmquist (1953) first suggests comparing the input of a firm at two different points in time in terms of the maximum factor by which the input in one period could be decreased such that the firm could still produce the same output level of the other time period. This idea lead to the Malmquist input index. Caves et al. (1982) extend the Malmquist input index to define a Malmquist productivity index. Färe et al. (1994) develop DEA-based Malmquist productivity measures.

Suppose each DMU_j ($j=1,2,\ldots,n$) produces a vector of outputs $y_j^t=\left(y_{1j}^t,\ldots,y_{sj}^t\right)$ by using a vector of inputs $x_j^t=\left(x_{1j}^t,\ldots,x_{mj}^t\right)$ at each time period t, $t=1,\ldots,T$. From t to $t+1$, DMU_o's efficiency may change or (and) the frontier may shift. Malmquist productivity index is calculated via

I. Comparing x_o^t to the frontier at time t, i.e., calculating $\theta_o^t\left(x_o^t, y_o^t\right)$ in the following input-oriented CRS envelopment model

$$
\begin{aligned}
&\theta_o^t(x_o^t, y_o^t) = \min \theta_o \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_j^t \le \theta_o x_o^t \\
&\sum_{j=1}^{n} \lambda_j y_j^t \ge y_o^t \\
&\lambda_j \ge 0,\ j=1,\ldots,n
\end{aligned}
\tag{19.1}
$$

where $x_o^t=\left(x_{1o}^t,\ldots,x_{mo}^t\right)$ and $y_o^t=\left(y_{1o}^t,\ldots,y_{so}^t\right)$ are the input and output vectors of DMU_o among others.

II. Comparing x_o^{t+1} to the frontier at time $t+1$, i.e., calculating $\theta_o^{t+1}(x_o^{t+1}, y_o^{t+1})$

$$\theta_o^{t+1}(x_o^{t+1}, y_o^{t+1}) = \min \theta_o$$
$$\text{subject to}$$
$$\sum_{j=1}^{n} \lambda_j x_j^{t+1} \le \theta_o x_o^{t+1}$$
$$\sum_{j=1}^{n} \lambda_j y_j^{t+1} \ge y_o^{t+1}$$
$$\lambda_j \ge 0, j = 1, \dots, n \tag{19.2}$$

III. Comparing x_o^t to the frontier at time $t+1$, i.e., calculating $\theta_o^{t+1}\left(x_o^t, y_o^t\right)$ via the following linear program

$$\theta_o^{t+1}(x_o^t, y_o^t) = \min \theta_o$$
$$\text{subject to}$$
$$\sum_{j=1}^{n} \lambda_j x_j^{t+1} \le \theta_o x_o^t$$
$$\sum_{j=1}^{n} \lambda_j y_j^{t+1} \ge y_o^t$$
$$\lambda_j \ge 0, j = 1, \dots, n \tag{19.3}$$

IV. Comparing x_o^{t+1} to the frontier at time t, i.e., calculating $\theta_o^t(x_o^{t+1}, y_o^{t+1})$ via the following linear program

$$\theta_o^t(x_o^{t+1}, y_o^{t+1}) = \min \theta_o$$
$$\text{subject to}$$
$$\sum_{j=1}^{n} \lambda_j x_j^t \le \theta_o x_o^{t+1}$$
$$\sum_{j=1}^{n} \lambda_j y_j^t \ge y_o^{t+1}$$
$$\lambda_j \ge 0, j = 1, \dots, n \tag{19.4}$$

The Malmquist productivity index is defined as:

$$M_o = \left[\frac{\theta_o^t(x_o^t, y_o^t)}{\theta_o^t(x_o^{t+1}, y_o^{t+1})} \frac{\theta_o^{t+1}(x_o^t, y_o^t)}{\theta_o^{t+1}(x_o^{t+1}, y_o^{t+1})} \right]^{\frac{1}{2}}$$

M_o measures the productivity change between periods t and $t+1$. Productivity declines if $M_o > 1$, remains unchanged if $M_o = 1$ and improves if $M_o < 1$.

The following modification of M_o makes it possible to measure the change of technical efficiency and the movement of the frontier in terms of a specific DMU_o.

$$M_o = \frac{\theta_o^t(x_o^t, y_o^t)}{\theta_o^{t+1}(x_o^{t+1}, y_o^{t+1})} \cdot \left[\frac{\theta_o^{t+1}(x_o^{t+1}, y_o^{t+1})}{\theta_o^t(x_o^{t+1}, y_o^{t+1})} \frac{\theta_o^{t+1}(x_o^t, y_o^t)}{\theta_o^t(x_o^t, y_o^t)}\right]^{\frac{1}{2}}$$

The first term on the right hand side measures the magnitude of technical efficiency change between periods t and $t+1$. Obviously, $\frac{\theta_o^t(x_o^t, y_o^t)}{\theta_o^{t+1}(x_o^{t+1}, y_o^{t+1})} \stackrel{<}{\underset{>}{=}} 1$ indicating that technical efficiency improves, remains or declines. The second term measures the shift in the EPF between periods t and $t+1$.

Similarly, we can develop the Malmquist index based upon the output-oriented DEA models.

To calculate the Malmquist, we can use the envelopment models and the variable benchmarking models. (This allows us to calculate the Malmquist index under non-CRS conditions with an additional constraint on $\sum_{j=1}^{n} \lambda_j (= 1, \le 1, \text{or} \ge 1)$ even if models (19.3) and (19.4) may be infeasible.

The DEA Excel Solver provides a menu item that calculate the Malmquist index based upon models (19.1)–(19.4).

The data for DMUs in each period should be placed in a sheet with a name starting with “Period”. For example, “Period1”, Period-1”, or “Period A”. The software will first look for the Period sheets once you select the Malmquist menu item. Select two periods to perform the Malmquist calculation (see Fig. 19.5)

The results are reported in the “Malmquist Index” sheet reporting M_o, along with four worksheets related to the results from models (19.1) to (19.4). The names of these four worksheets depend on the periods selected. Suppose “Period A” and “Period2” are selected. Then the name for the four worksheets are (i) “M Period A” (model 19.1); (ii) “M Period2” (model 19.2); (iii) “M Period2-Period A” (model 19.3); and (iv) “M Period A-Period2” (model 19.4). For the latter two, the left side of the name after “M” represents the reference set and the right side the period under evaluation.

19.5 DEAFrontier Models

Figure 19.6 shows various DEA models included in the DEAFroniter. The following list shows where each DEA model/approach is discussed.

DEA Bootstrapping: Chapter 19
Multiplier Model with Epsilon: Chapter 3
Restricted Multiplier: Chapter 3

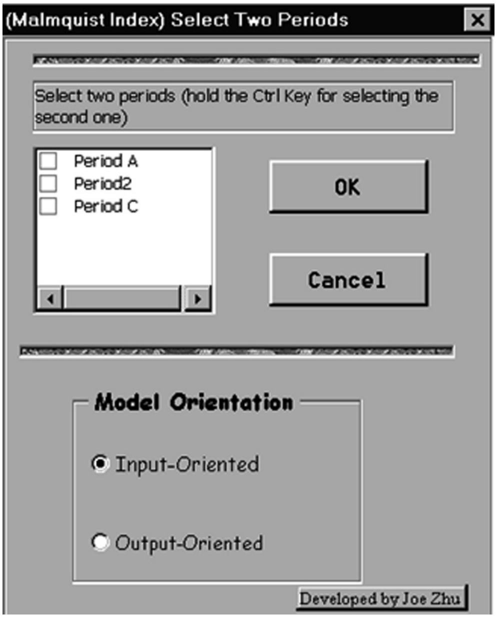
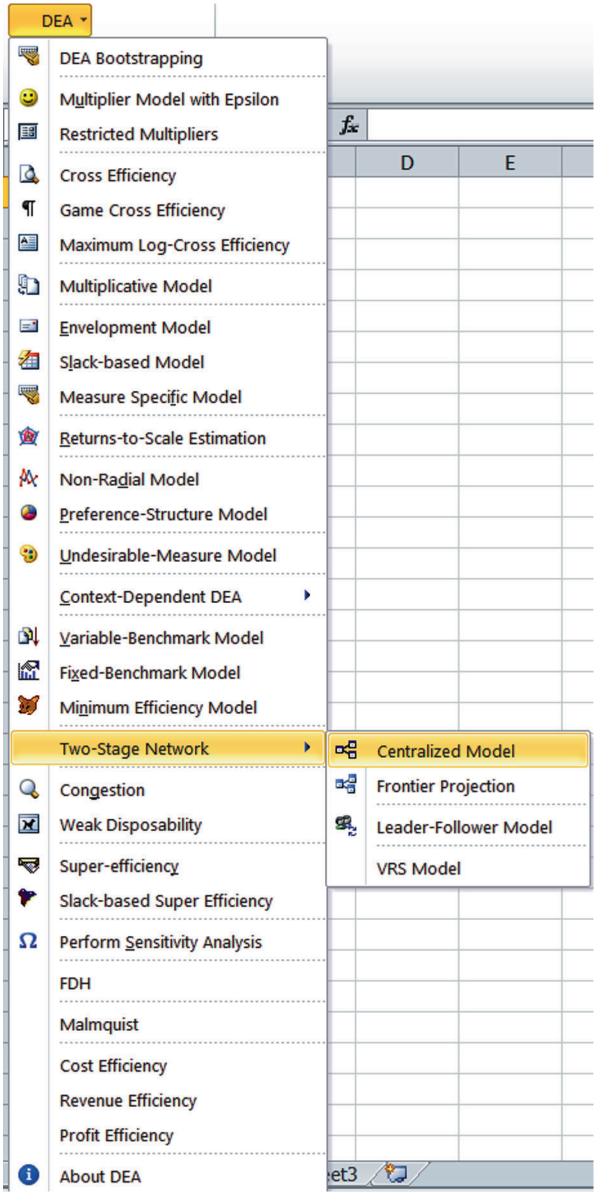


Fig. 19.5 Malmquist

Cross Efficiency: Chapter 4
Game Cross Efficiency: Chapter 4
Maximum Log-Cross Efficiency: Chapter 4
Multiplicative Model: Chapter 4
Envelopment Model: Chapter 2
Slack-based Model: Chapter 5
Measure Specific Model: Chapter 6
Returns-to-Scale Estimation: Chapter 13
Non-Radial Model: Chapter 7
Preference-Structure Model: Chapter 7
Undesirable Measure Model: Chapter 8
Context-dependent DEA: Chapter 9
Variable-benchmark Model: Chapter 12
Fixed-benchmark Model: Chapter 12
Minimum Efficiency Model: Chapter 12
Two stage Network includes a series models: Chapter 14
Congestion: Chapter 16
Weak Disposability: Chapter 16
Super-efficiency: Chapter 10
Slack-based super efficiency: Chapter 10
Perform Sensitivity Analysis: Chapter 11
FDH: Chapter 19
Malmquist: Chapter 19
Cost Efficiency, Revenue Efficiency, and Profit Efficiency: Chapter 7

Fig. 19.6 DEAFrontier



Electronic Supplementary Material

The online version of this chapter (doi:10.1007/978-3-319-06647-9_19) contains supplementary material, which is available to authorized users.

References

Deprins, D., Simar, L., & Tulkens, H. (1984). Measuring labor-efficiency in post offices. In M. Marchand, P. Pestieau, & H. Tulkens (Eds.), *The performance of public enterprises: Concepts and measurement* (pp. 243–267). North-Holland: Elsevier Science Publishers B.V.

Malmquist, S. (1953). Index numbers and indifference surfaces. *Trabajos de Estatistica, 4,* 209–242.

Caves, D. W., Christensen, L. R., & Diewert, W. E. (1982). The economic theory of index numbers and the measurement of input, output, and productivity. *Econometric, 50*(6), 1939–1414.

Färe, R., Grosskopf, S., & Lovell, C. A. K. (1994). *Production frontiers.* Cambridge: Cambridge University Press.

Lothgren, M., & Tambour, M. (1999). Bootstrapping the data envelopment analysis Malmquist productivity index. *Applied Economics, 31,* 417–425.

Simar, L., & Wilson, P. (1998). Sensitivity analysis of efficiency scores: How to bootstrap in non-parametric frontier models. *Management Science, 44*(1), 49–61.

Licensing and Warranty Agreement

READ THIS: Do not install the software until you have read and agreed to this agreement. By opening the accompanying software, you acknowledge that you have read and accepted the following terms and conditions. If you do not agree and do not want to be bounded by such terms and conditions, do not install or use the software "DEAFrontier".

License: The copyright to the software (the "DEAFrontier") is owned by the author. The "DEAFontier" is protected by the United States copyright laws and international treaty provisions. No part of the DEAFrontier may be reproduced, stored in a retrieval systems, distributed (including but not limited to over the Internet), modified, decompiled, reverse engineered, reconfigured, transmitted, or transcribed, in any form or by any means without the permission of the author. The DEAFontier may not, under any circumstances, be reproduced for sale. This license allows you to use the DEAFontier for educational and research purposes only, not for commercial purposes, including consulting. You may only (i) make one copy of the DEAFontier for backup or archival purposes, or (ii) transfer the DEAFontier to a single hard disk, provided that you keep the original for backup or archival purposes. You may not (i) rent or lease the DEAFontier, (ii) copy or reproduce the DEAFontier through a LAN or other network system or through any computer subscriber system, or (iii) modify, adapt, or create derivative works based upon the DEAFontier. You may be held legally responsible for any copying or copyright infringement which is caused by your failure to abide by the above terms and conditions.

Limited Warranty: The Publisher and the author provide no other warranties, expressed or implied, including without limitation implied warranties of merchantability and fitness for a particular purpose, and shall not be liable for any damages, including direct, special, indirect, incidental, consequential, or otherwise. The Publisher does not warrant that the functions contained in the DEAFontier will meet your requirements or that the operation of the DEAFontier will be error-free.

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*,
International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9, © Springer International Publishing Switzerland 2014

Index

J. Zhu, *Quantitative Models for Performance Evaluation and Benchmarking*, International Series in Operations Research & Management Science 213,
DOI 10.1007/978-3-319-06647-9,

V

W