Preface

This book *Algebraic Modeling Systems—Modeling and Solving Real World Optimization Problems*—deals with aspects of modeling and solving real-world optimization problems in a unique combination. It treats systematically the major algebraic modeling languages (AMLs) and algebraic modeling systems (AMSs) used to solve mathematical optimization problems. AMLs helped significantly to increase the usage of mathematical optimization in industry. Therefore, it is a logical consequence that the GOR (Gesellschaft für Operations Research) Working Group *Mathematical Optimization in Real Life* had a second meeting devoted to AMLs, which, after 7 years, followed the one held under the title *Modeling Languages in Mathematical Optimization* during April 23–25, 2003, in the German Physics Society Conference Building in Bad Honnef, Germany. Similar to the first meeting which resulted in the book *Modeling Languages in Mathematical Optimization*, this book is an offspring of the 86th Meeting of the GOR working group again held in Bad Honnef under the title *Modeling Languages in Mathematical Optimization—Overview, Opportunities & Challenges in Application Development*—during November 18–19, 2010. We note that some modeling languages have been sold from the original creators or developers and found a new haven. This time, the modeling language providers

AMPL Robert Fourer, Northwestern Univ.; David M. Gay, AMPL Optimization LLC., NJ,
CPLEX Studio Ferenc Katai, IBM ILOG SWG AIM, Valbonne, France,
GAMS Alexander Meeraus & Jan H. Jagla, GAMS Development Corp., Washington D.C.,
Mosel Susanne Heipcke & Oliver Bastert, FICO (previously, Dash Optimization),

gave deep insight into their motivations and conceptual software design features, highlighted their advantages, and also critically discussed their limits. These lectures were enhanced by presentations of practitioners sharing their experience with the audience, a talk about formal mathematical languages, and outlining the concept of BoFiT, a Graphical Modeling Framework for Mixed Integer Programs. The participants benefited greatly from this symposium with its useful overview and orientation on today's modeling languages in optimization and their capabilities.

Roughly speaking, a modeling language serves the need to pass data and a mathematical model description to a solver in the same way that people, especially mathematicians, describe those problems to each other. Of course, in reality this is not done in exactly the same way, but the resemblance has to be close enough to spare the user any significant translation. As in this book we focus on modeling languages used in mathematical optimization, let us give an example from that discipline. When practitioners or researchers describe large-scale mixed integer linear programs (MILPs) to each other, they use summations and subscripting as well as domain specifications of the variables. To give a negative definition first: Probably one would not consider a language lacking these features to be a modeling language for large-scale MILP. This can be turned into a positive definition: A modeling language in mathematical optimization needs to support the expressions and symbols used in the mathematical optimization community. Therefore, it is natural that algebraic modeling languages support the concepts of data, variables, constraints, and objective functions. Those entities are connected not only by algebraic operations $(+, -, \cdot, /)$ but often also by nonlinear functions. Algebraic models are embedded in a larger class of differential-algebraic models including ordinary or partial differential equations. They may also be extended toward relationships appearing in constraint programming. Examples are membership relations or all-different constraints.

The earliest algebraic modeling languages appeared in the late 1970s and early 1980s. They were already very useful, supporting analysts to input their problems to solvers. In the first middle of the 1980s, when, for instance, `AMPL`, `GAMS`, `MPL`, and `mp-model` appeared, software developers were already trying to improve on previous designs, by taking advantage of faster computers and better computing environments.

Modeling languages supporting differential equations in addition to algebraic terms have their roots in the process industry and are covered by Kallrath [6]. Examples are `MINOPT`, `PCOMP`, and `gPROMS`.

While a more precise definition of modeling languages is given by Schichl [8], it is appropriate at this place to focus for a moment on the terms *modeling language* and *modeling systems*. Some people use these expressions synonymously, and this is also reflected in the acronyms `AMPL` and `GAMS`, where the former translates into *A Mathematical Programming Language* while the latter stands for *General Algebraic Modeling System*. In this book we rather keep the following meaning. In its purest sense, a modeling language in mathematical optimization is a means to give a declarative representation[1] of an optimization problem; `AMPL`, `GAMS`, and

[1] As the book, in the sense of this definition, focuses on algebraic declarative modeling languages, the reader should not be surprised to find not too much on mathematical modeling systems such as Mathematica, MathCad, MAPLE, or MATLAB. These systems are procedural tools.

mp-model are good examples. A modeling system is rather a complete support system to solve real-world problems. It usually contains a modeling language but offers many other features supporting the solution process, e.g., passing commands to the solver, analyzing infeasibilities, incorporating the solver's output back into the model or its next steps, or visualizing the branch&bound tree or the structure of the matrix; many more features could be listed. AIMMS, Mosel, MPL, and OPL Studio (now, CPLEX-Studio) are good examples of modeling systems. Of course, one might argue that the existing modeling languages are usually somewhat in between those extreme definitions. The early versions of AMPL were almost completely declarative, while GAMS from the beginning had also procedural features supporting, e.g., to input the output of one model into subsequent models.

As outlined by Kallrath (2011) in Chap. 1 of this book, AMLs have played and still play an important role in the mathematical optimization community and optimization used in industry. In the early 1980s, they found an initial market niche by enabling the user to formulate NLP problems; they supported automatic differentiation, i.e., they symbolically generated the first and second derivative information. AMLs and their broad distribution were triggered by the advent of personal computers (PCs). Dash Optimization with their solver XPRESS-OPTIMIZER and their modeling language mp-model provided a tool to PC users rather than mainframes. Thus, after a while, AMLs also became superior in implementing LP models and succeeded IBM's accepted industrial standard, MPS. Nowadays, the AMLs ensure the robustness, stability, and data checks needed in industrially stable software. Furthermore, AMLs accelerate the development and improvement of solvers ranging from Linear Programming to Mixed Integer Nonlinear Programming and even Global Optimization techniques. On one hand, users can easily switch from one solver to another one. On the other hand, the solver developers can count on a much larger market when their solver is embedded into an AML. Last but not least, the development of Microsoft Windows and improved hardware technology leads to graphical user interfaces such as Xpress-IVE for Mosel, GAMSIDE in GAMS, or systems such as AIMMS and MPL. This increases the efficiency of working with AMLs and contributes greatly to the fact that AMLs reduce the project time, make maintenance easier and extend the lifetime of optimization software.

This book is aimed at researchers of mathematical programming and operation research, scientists in various disciplines modeling and solving optimization problems, supply chain management consultants, and practitioners in the energy industry. It is beneficial to decision makers in the area of tool selection for optimization tasks as well as students and graduates in mathematics, physics, operations research, and businesses with interest in modeling and solving real optimization problems. Often application software has implemented an optimization model without an algebraic modeling language. The people responsible for maintaining or further developing such applications might be looking for improvements to put their software on a safer footing. They will definitely benefit from this book. Assuming some background in mathematics and optimization or at least a certain willingness to acquire the skills necessary to understand the described algorithms and models, this book provides a sound overview on model formulation and solving as well as implementation of solvers in various modeling language packages. It demonstrates the strengths and characteristic features of such languages. May it provide a bridge for researchers, practitioners, and students into a new world of excitement: solving real-world optimization problems with the most advanced modeling systems.

Structure of this Book

This book benefits from contributions of experienced practitioners and developers of modeling languages and modeling system, respectively. The languages presented during the symposium as well as `ZIMPL` are covered in great detail in chapters of their own. We have structured the book in three parts:

- Introduction and Foundations
- Selected Algebraic Modeling Systems
- Aspects of Modeling and Solving Real World Problems

The book's first part contains introductory material. Chapter 1 starts with an introduction into AMLs, illuminates the meaning of the term model, and includes a brief overview on classes of optimization problems. The main entities in optimization—variables, constraints, and the objective function—are explained. The contrast between procedural and declarative languages is worked out in detail, followed by the importance of teaching and learning modeling. Chapter 2 discusses theoretical aspects and conceptual properties of semantic representation of mathematical specifications, an interesting aspect in the vicinity of declarative languages.

The second and main part of the book illuminates selected algebraic modeling systems. It covers `CPLEX Studio` (formerly, `OPL Studio`), `GAMS`, `Mosel`, and `ZIMPL` in great detail. Those chapters reflect the personal views and focus of the authors. It is left to the reader to compare the languages and see which one serves his personal needs best. The `GAMS` contribution focuses on the purposes of modeling languages with respect to different groups of models or customers, respectively, and describes a few special features of this modeling language. More on language elements related to constraint programming is found in the `CPLEX Studio` and `Mosel` chapters. A useful property in the modeling language `Mosel`is its concept of *modularity*, which makes it possible to extend this language according to one's needs and to provide new functionality, in particular to deal with other types of problems and solvers. There are a few languages or systems not covered in this book because their developers or authors were not able to contribute to this book: Microsoft's solver suite and Excel plug-ins. Others have not seen an updated contribution; for those we refer the reader to the individual chapters in Kallrath [6]. In this group we find:

- `AIMMS`, which is quite different from all others due to its object-oriented design and provides agent-based simulation techniques [1].
- `AMPL` has a dominant position in universities, is often used for prototyping, and has many innovative ideas in its pipeline [3, 4].

- `LPL` with some neat features not found in any other language [5].
- `LINGO` with general global optimization algorithms which partition nonconvex models into a number of smaller convex models and then use a B&B manager to find the global optimal point over the entire variable domain [2].
- `NOP-2`, a special language by Schichl and Neumaier [9] for dealing with nonconvex nonlinear problems to be solved to global optimality.
- `MPL` [7] which has its strength on computer science and covers topics such as speed, scalability, data management, and deployment aspects of optimization.

The third part contains discussions of various aspects of modeling and solving real world problems among them reasons for selecting a procedural programming or a declarative modeling language (Chap. 8), data preparation (Chap. 9), visualization of data structures (Chap. 10), and a feature wish list and reflections about the current and possible future role of AMLs (Chaps. 11 and 12).

By showing the strengths and characteristic features of algebraic modeling languages as well as indicating trends, we hope to give novices and practitioners in mathematical optimization, operations research, supply chain management, energy industry, financial industry, and other areas of industry a useful overview. It may help decision makers to select the best modeling system satisfying their needs.

Weisenheim am Berg *Josef Kallrath*

References

1. Bisschop, J., Roelofs, M.: The modeling language AIMMS. In: Kallrath, J. (ed.) Modeling Languages in Mathematical Optimization, pp. 71–104. Kluwer Academic, Norwell (2004)
2. Cunningham, K., Schrage, L.: The LINGO algebraic modeling language. In: Kallrath, J. (ed.) Modeling Languages in Mathematical Optimization, pp. 159–171. Kluwer Academic, Norwell (2004)
3. Fourer, R., Gay, D.M., Kernighan, B.W.: AMPL: a modeling language for mathematical programming. Duxbury Press, Brooks/Cole Publishing Company, Monterey (1993)
4. Fourer, R., Gay, D.M., Kernighan, B.W.: Design principles and new developments in the the AMPL modeling language. In: Kallrath, J. (ed.) Modeling languages in mathematical optimization, pp. 105–135. Kluwer Academic, Norwell (2004)
5. Hürlimann, T.: The LPL modeling language. In: Kallrath, J. (ed.) Modeling languages in mathematical optimization, pp. 173–183. Kluwer Academic, Norwell (2004)
6. Kallrath, J. (ed.): Modeling languages in mathematical optimization. Kluwer Academic, Norwell (2004)
7. Kristjansson, B., Lee, D.: The MPL modeling system. In: Kallrath, J. (ed.) Modeling languages in mathematical optimization, pp. 239–266. Kluwer Academic, Norwell (2004)
8. Schichl, H.: Theoretical concepts and design of modeling languages. In: Kallrath, J. (ed.) Modeling languages in mathematical optimization, pp. 45–62. Kluwer Academic, Norwell (2004)
9. Schichl, H., Neumaier, A.: The NOP-2 modeling language. In: Kallrath, J. (ed.) Modeling languages in mathematical optimization, pp. 279–291. Kluwer Academic, Norwell (2004)

http://www.springer.com/978-3-642-23591-7

Algebraic Modeling Systems
Modeling and Solving Real World Optimization Problems
Kallrath, J. (Ed.)
2012, XXVI, 238 p., Hardcover
ISBN: 978-3-642-23591-7