12. Robust Optimization in Quality Engineering

Quality engineers often face the job of identifying process or product design parameters that optimize performance response. The first step is to construct a model, using historical or experimental data, that relates the design parameters to the response measures. The next step is to identify the best design parameters based on the model. Clearly, the model itself is only an approximation of the true relationship between the design parameters and the responses. The advances in optimization theory and computer technology have enabled quality engineers to obtain a good solution more efficiently by taking into account the inherent uncertainty in these empirically based models.

Two widely used techniques for parameter optimization, described with examples in this chapter, are the response surface methodology (RSM) and Taguchi loss function. In both methods, the response model is assumed to be fully correct at each step. In this chapter we show how to enhance both methods by using robust optimization tools that acknowledge the uncertainty in the models to find even better solutions. We develop a family of models from the confidence region of the model parameters and show how to use sophistical optimization techniques to find better design parameters over the entire family of approximate models.

Section 12.1 of the chapter gives an introduction to the design parameter selection problem and motivates the need for robust optimization. Section 12.2 presents the robust optimization approach to address the problem of optimizing empirically based response functions by developing a family of models from the confidence region of the model parameters. In Sect. 12.2 robust optimization is compared to traditional optimization approaches where the empirical model is assumed to be true and the optimization is conducted without considering the uncertainty in the parameter estimates. Simulation is used to make the comparison in the context of response surface methodology, a widely used method to optimize products and processes that is briefly described in the section. Section 12.3 introduces a refined technique, called weighted robust optimization, where more-likely points in the confidence region of the empirically determined parameters are given heavier weight than less-likely points. We show that this method provides even more effective solutions compared to robust optimization without weights. Section 12.4 discusses Taguchi's loss function and how to leverage robust optimization methods to obtain better solutions when the loss function is estimated from empirical experimental data.

One of the central themes in quality engineering is the identification of optimal values for the design parameters to make a process or product function in the best possible way to maximize its performance. The advances in optimization theory and computing technology in the last half century have greatly stimulated the progress in quality improvement—optimization methodology has provided a systematic framework to guide today's quality engineers to identify optimal levels in design parameters efficiently, while the same task would have taken many iterations of experiments for engineers one generation ago without the aid of modern optimization techniques.

Many quality engineering problems arising in today's complex manufacturing processes can be reduced to some optimization problem. For example, in process control problems, we are interested in selecting a best possible set of values for process settings to maximize the output of the final products that satisfy the specifications in the shortest amount of time. In the context of product design problems, the purpose is to choose an optimal mix of design parameters to maximize the performance measures of the new products.

The iteration process in applying optimization techniques to solve quality improvement problems includes the following steps:

1. Convert the quality requirements and specifications to an optimization model; (12.1)
2. Solve the optimization problems and identify the optimal values for the decision variables, i. e., the process settings or design parameters;
3. Apply the optimal solution identified in step 2 to the actual process control or product design environment, validate the effectiveness of the optimal solution and revise the optimization model if necessary.

There exists a large volume of literature advocating the use of optimization techniques to improve process and product quality; see *Box* et al. [12.1], *Box* and *Draper* [12.2], *Myers* and *Montgomery* [12.3], *Khuri* and *Cornell* [12.4], among many others.

The most critical step in the above procedure is to construct the optimization model using the historical or experimental data collected in the process control or product design stage. Usually we tend to regard a model constructed on empirical data as a true physical law. Thus we assume that the model accurately describes the underlying process or product and that the optimal solution to the model is better than any other choice.

However there is much uncertainty involved in the model construction process. First, the most common uncertainty comes from the measurement error and noise effect. The devices used to capture the readings are more or less subject to measurement errors. Noise factors, such as environmental conditions and material properties, will sometimes severely distort the values of the true performance measure. Second, the failure to identify and record every possible main factor that contributes to the final performance measure will certainly degrade the quality of the model since it cannot incorporate all of the major predictors. Finally the model selection process adds another layer of uncertainty in the final model we will reach. There are numerous forms of models we can choose from. For example, should we develop a linear model or a nonlinear one? If it is a nonlinear model, should we try a higher-order polynomial function or a logistic function, or something else?

The uncertainty in the model construction process poses huge challenges to the statistical sciences, which have provided numerous methods to identify effective models to represent the true relationship between the design parameters and process/product performance as closely as possible. However, although statistics is highly useful in reducing the uncertainty in a response model, it does not eliminate all of the sources of the uncertainty. Therefore the resulting optimization model, constructed from the empirical data through careful adjustment and calibration using statistical methods, is not a perfect mirror of the true relationship; it is an approximation of the true mechanism in the underlying process or product. We have an opportunity in the optimization stage to address the uncertainty inherent to the statistical model to enhance the optimal solution.

In the context of quality engineering, response surface methodology (RSM) is a set of statistical and optimization techniques that are used sequentially to identify the optimal solution to a quality improvement problem. The iterative procedure in RSM includes performing an experiment in the region of the best known solution, fitting a response model to the experimental data, and optimizing the estimated response model. RSM has been widely used in quality engineering since the seminal work of George Box in the 1950s; for more details see *Box* and *Wilson* [12.5]. We give a brief introduction to RSM in Sect. 12.2 of this chapter.

In RSM, the optimization procedure is performed directly on the estimated response model, so it does not deliver a solution that minimizes the uncertainty in the model estimation process. This chapter is motivated by the work in *Xu* and *Albin* [12.6] and provides

an introduction into how we can use robust optimization methods to identify good solutions that maximize the performance of a process or product and, in the meantime, address the uncertainty inherent to a response model that is an approximation of the true relationship between the design parameters and the process/product performance. To make this idea clearer, consider the following function $y = f(\boldsymbol{x}, \boldsymbol{\beta})$, where y is the true process/product performance, $\boldsymbol{x}$ includes a set of design parameters and the function $f(x, \cdot)$ describes the true relationship between the design parameters in x and the process/product performance y. The vector $\boldsymbol{\beta}$ captures the important parameters in the function $f(x, \cdot)$.

If the function is a first-order polynomial

$$y = \sum_{i=1}^{n} \beta_i x_i \ ,$$

then $\boldsymbol{\beta}$ is a vector including all the coefficients $(\beta_1, \beta_2, \cdots \beta_n)'$. If the function is a second-order polynomial

$$y = \sum_{1 \le i \le j \le n} \beta_{ij} x_i x_j + \sum_{i=1}^{n} \beta_i x_i \ ,$$

then the vector $\boldsymbol{\beta}$ can be written as

$$(\beta_{11}, \beta_{12}, \cdots \beta_{nn}, \beta_1, \beta_2, \cdots \beta_n)' \ .$$

We note that the function $f(\boldsymbol{x}, \boldsymbol{\beta})$ is linear in the coefficients in $\boldsymbol{\beta}$ when $f(\boldsymbol{x}, \boldsymbol{\beta})$ is a polynomial of x. This property plays an important role in the robust method introduced in this chapter.

The parameters in $\boldsymbol{\beta}$ are important in that they characterize how a process or product behaves. For example, let us consider a mixture design problem on glass/epoxy composites. We are interested in choosing the optimal mix of glass and epoxy to maximize the strength of the composites. Assume the relationship between the strength y and the fraction of glass (x_1) and epoxy (x_2) can be described by the response function $y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$. The parameter β_1 (β_2) measures how the composite strength changes in response to the change in the fraction of glass (epoxy) while the parameter β_3 measures the glass–epoxy interaction effect on the composite strength.

Although the parameters in $\boldsymbol{\beta}$ are crucial in determining the behavior of a process or product, the true values for $\boldsymbol{\beta}$ are usually unknown to quality engineers. The only way to derive the values for $\boldsymbol{\beta}$ is by fitting a statistical model to the experimental data. Since the coefficients in $\boldsymbol{\beta}$ are estimated values, instead of writing $y = f(\boldsymbol{x}, \boldsymbol{\beta})$, we will use the notation $y = f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$, where $\hat{\boldsymbol{\beta}}$ is estimated from historical or experimental data.

In quality engineering problems, we usually use the canonical optimization approach to determine the optimal solution. We first estimate the model $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$ from the experimental data and treat it as a true characterization of the underlying model. Then we solve for the optimal solution to the model $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$. In the canonical approach, the point estimates in $\hat{\boldsymbol{\beta}}$ are regarded as a single representation of the true parameters in $\boldsymbol{\beta}$ and thus the optimization steps do not take into account the uncertainty about the estimate $\hat{\boldsymbol{\beta}}$. Although the canonical approach provides a simple, practical way to optimize the process/product performance, the solution obtained from the canonical approach may be far from optimal under the true performance response model.

Figure 12.1 illustrates the potential danger of the canonical approach when the performance response model is a second-order model. The dashed curve on the right represents the true, but unknown, model and the solid curve on the left the fitted model. If the goal is to minimize the performance response, the optimal value of the design variable is D_0 and the optimal performance response is 0. The canonical approach selects the value D_1 for the design variable, which results in the performance response P_1, well above the true optimal. Thus, even a slight deviation of the fitted model from the true model might result in unacceptable performance.

Section 12.2 in this chapter presents the robust optimization approach to address the pitfall illustrated in the above example. In contrast to the canonical approach, where uncertainty about the estimates $\hat{\boldsymbol{\beta}}$ is not explicitly addressed and only a single model $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$ is optimized, the robust approach considers a family of models and each model in the family is a possible representation of the true model. Robust and canonical optimization

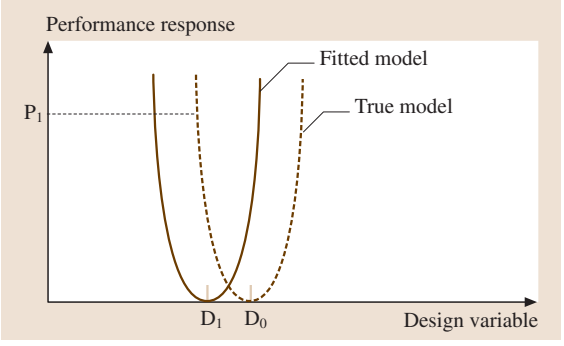

Fig. 12.1 Optimizing the estimated model yields performance response P_1, significantly higher than the true minimum 0

are compared using a Monte Carlo simulation example in Sect. 12.2, in the context of response surface methodology (RSM), a widely used method to optimize product/process.

The single estimated model $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$ in the canonical approach is the most likely representation of the true model $f(\boldsymbol{x}, \boldsymbol{\beta})$, while the robust approach incorporates more information by considering a family of models. Section 12.3 takes this a step further by combining each individual model in this family with a likelihood measure of how close it is to the true $f(\boldsymbol{x}, \boldsymbol{\beta})$. The improved approach presented in Sect. 12.3 is called the weighted robust optimization method and we prove that it provides a more effective solution to the estimated optimization model.

One of the greatest achievements in quality engineering in the last century is the robust design method that Taguchi proposed to minimize the expectation of Taguchi's loss function. The loss function is a measure of the deviation of the product performance from the desired target. The quality of a product can be measured by the loss function so a robust design problem can be reduced to an optimization problem whose objective is to minimize the expectation of the loss function by choosing optimal levels in design parameters.

To obtain the objective function in the robust design problem, designed experiments must be conducted, experimental data be collected and the loss function be fitted from the data. We are confronted with the same problems as discussed earlier on the uncertainty associated with the inference from the experimental data. Therefore robust optimization approach can be applied to identify a robust set of values for the design parameters. Section 12.4 discusses how we can leverage the robust optimization methods to better address Taguchi's robust design problems.

12.1 An Introduction to Response Surface Methodology

Response surface methodology (RSM) is a sequential approach and comprises iterative steps to conduct designed experiments, estimate the response model and derive the optimal solution in sequence. This section introduces the most essential steps in RSM and we refer the reader to *Box* and *Draper* [12.2], *Myers* and *Montgomery* [12.3], *Khuri* and *Cornell* [12.4] for a more comprehensive introduction to RSM.

We assume that, prior to running RSM, we have selected a list of significant factors that are the most important contributors to the response. Screening experiments such as fractional factorial designs and Plackett–Burman designs can be used to identify the important factors; see *Wu* and *Hamada* [12.7].

Let $\boldsymbol{x} = (x_1, x_2, \cdots x_k)$ denote the factors we have selected. We first run a first-order experiment such as 2^k factorial designs and fit a linear model to the experimental data. The next step is to choose the path of steepest ascent or steepest descent, run several experiments along the path and choose the one with the best performance response. We move the experimental region to the new location identified on the steepest ascent (descent) path and run the first-order experiments using the same steps above. We continue this process until no improvement is possible using first-order experiments. A second-order experiment, such as central composite designs, is conducted in order to describe the response surface better. We then solve a quadratic optimization model, obtain the solution and run confirmatory experiments to validate the optimal solution.

We use a paper helicopter example to illustrate the steps described above. The purpose of this exercise is to design a paper helicopter by choosing the optimal levels for rotor length/width, body length/width and other factors to maximize the flight time of the helicopter. Due to the convenience of the design, this exercise has been used in several institutions to teach design of experiments and RSM. We use the results presented in *Erhardt* and *Mai* [12.8] to demonstrate the basic steps in RSM. Another good source for the design of paper helicopter using RSM can be found in *Box* and *Liu* [12.9].

In *Erhardt* and *Mai* [12.8], there are eight factors that are likely to contribute to the flight time of the paper helicopter: rotor length, rotor width, body length, foot length, fold length, fold width, paper weight, and direction of fold. Screening experiments were conducted and the investigators found that two of the eight variables, rotor length and rotor width, are important in determining the flight time.

Erhardt and *Mai* [12.8] conducted a 2^2 factorial experiment with replicated runs and center points. The experimental data is shown in Table 12.1. The coded level 1 and -1 for rotor length stands for 11.5 and 5.5 cm, respectively. The coded level 1 and -1 for rotor width stands for 5 and 3 cm, respectively.

The first-order model fitted to the data in Table 12.1 is

$$\text{Flight time} = 11.1163 + 1.2881 \times \text{Rotor length} - 1.5081 \times \text{Rotor width}\,.$$

Therefore the path of steepest ascent is (1.2881, −1.5081) in coded level; in other words, for every one centimeter of increase in rotor length, rotor width should be decreased by

$$\frac{1.5081}{1.2881} \times \frac{1}{3} = 0.39\,\text{cm}\,.$$

The investigators conducted five experiments along the steepest ascent path and the experimental data is recorded in Table 12.2. The combination of rotor length and width that gives the longest flight time is 11.5 and 2.83 cm.

The investigators then conduct a central composite design (CCD) by adding experimental runs at axial points. Table 12.3 below contains the data from the CCD experiment. The center point of the CCD design is (11.5, 2.83), which is the solution obtained from the experimental runs on the steepest ascent path. One coded unit stands for 1 cm for rotor length and 0.39 cm for rotor width.

Table 12.1 2^2 factorial design for paper helicopter example

Coded level		Actual level (cm)		Flight time (seconds)			
Rotor length	Rotor width	Rotor length	Rotor width	Replicate 1	Replicate 2	Replicate 3	Replicate 4
1	1	11.5	5	10.02	9.94	9.95	9.93
1	-1	11.5	3	16.52	16.99	12.58	13.86
-1	1	5.5	5	10.20	9.26	8.20	9.92
-1	-1	5.5	3	10.24	9.11	11.31	10.94
0	0	8.5	4	11.67	10.74	9.83	

Table 12.2 Experiments along the path of steepest ascent

	Rotor length (cm)	Rotor width (cm)	Flight time (s)
Base	8.5	4	12.99
Path of steepest ascent Δ	1	-0.39	
Base + 1×Δ	9.5	3.61	15.22
Base + 2×Δ	10.5	3.22	16.34
Base + 3×Δ	11.5	2.83	18.78
Base + 4×Δ	12.5	2.44	17.39
Base + 5×Δ	13.5	2.05	7.24

Table 12.3 Central composite design for paper helicopter example

Coded level		Actual level (cm)		Flight time (s)
Rotor length	Rotor width	Rotor length	Rotor width	
1	1	12.5	3.22	13.53
1	-1	12.5	2.44	13.74
-1	1	10.5	3.22	15.48
-1	-1	10.5	2.44	13.65
$\sqrt{2}$	0	12.91	2.83	12.51
$-\sqrt{2}$	0	10.08	2.83	15.17
0	$\sqrt{2}$	11.5	3.38	14.86
0	$-\sqrt{2}$	11.5	2.28	11.85
0	0	11.5	2.83	17.38
0	0	11.5	2.83	16.35
0	0	11.5	2.83	16.41

The second-order model fitted to the data in Table 12.3 is given below:

$$\text{Flight time} = 16.713 - 0.702x_1 + 0.735x_2 - 1.311x_1^2 - 0.510x_1x_2 - 1.554x_2^2 ,$$

where x_1 stands for rotor length and $x_2$stands for rotor width. The optimal solution by maximizing this quadratic model is $(-0.32,\ 0.29)$ in coded units, or 11.18 cm for rotor length and 2.94 cm for rotor width.

The paper helicopter example presented in this section is a simplified version of how response surface methodology works to address quality improvement. A complicated real-world problem may require many more iterations in order to find an optimal solution and many of the technical details can be found in the references given in the beginning of this section.

12.2 Minimax Deviation Method to Derive Robust Optimal Solution

As we discussed in the introduction, the estimated model $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$ is a single representation of the true relationship between the response y and the predictor variables in $\boldsymbol{x}$, where $\hat{\boldsymbol{\beta}}$ is a point estimate and is derived from the sample data. The solution obtained by optimizing a single estimated model $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$ may not work well for the true model $f(\boldsymbol{x}, \boldsymbol{\beta})$. This section introduces the minimax deviation method to derive the robust solution when the experimental or historical data is available to estimate the optimization model.

One assumption we make here is that the vector $\boldsymbol{\beta}$ in $f(\boldsymbol{x}, \boldsymbol{\beta})$ contains the coefficients in the model and we assume that $f(\boldsymbol{x}, \boldsymbol{\beta})$ is linear in the coefficients in $\boldsymbol{\beta}$. This assumption covers a wide range of applications since most of the models considered in quality engineering are derived using regression and the hypothetical model $f(\boldsymbol{x}, \boldsymbol{\beta})$ is always linear in regression coefficients even if the model itself is nonlinear in $\boldsymbol{x}$. For example, consider $f(\boldsymbol{x}, \boldsymbol{\beta}) = \beta_1 x^2 + \beta_2 x + \beta_3 \frac{1}{x}$, clearly $f(\boldsymbol{x}, \boldsymbol{\beta})$ is linear in $(\beta_1, \beta_2, \beta_3)$, although it is not linear in $\boldsymbol{x}$.

12.2.1 Motivation of the Minimax Deviation Method

Consider two models in Fig. 12.2 where model 1 is $y = f(\boldsymbol{x}, \boldsymbol{\beta}^{(1)})$ and model 2 is $y = f(\boldsymbol{x}, \boldsymbol{\beta}^{(12.1)})$. If we assume that model 1 and model 2 are equally likely to be the true one, then how do we choose the value for x to minimize the response y in the true model? If the value at point A is chosen, there is a 50% chance that the response value y reaches its minimum if model 1 is the true model, while we are facing another 50% chance that the response value y is much worse when model 2 is the true model. A similar conclusion can be made if point B is chosen. Thus a rational decision maker will probably

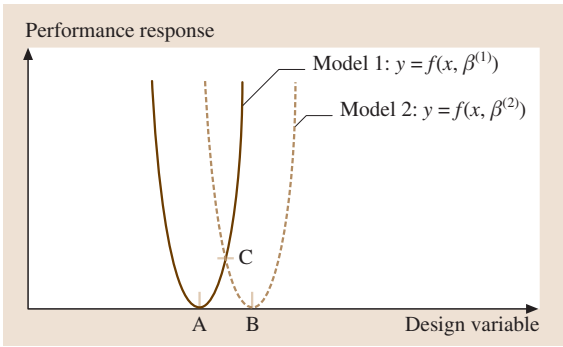

Fig. 12.2 Point C makes the response value close to the minimum whether model 1 or model 2 is the true model

choose point C such that the response value y will not be too far off from the minimum 0 whether model 1 or model 2 is the true one.

To formalize the reasoning, we use the following notation: let g_1 be the minimum value of $f(\boldsymbol{x}, \boldsymbol{\beta}^{(1)})$, and g_2 be the minimum value of $f(\boldsymbol{x}, \boldsymbol{\beta}^{(12.1)})$. For the example in Fig. 12.2, $g_1$and g_2 are both zeros. Given that model 1 and model 2 are equally likely to be the true model, a rational decision maker wants to find an x such that, when the true model is model 1, the response value at $\boldsymbol{x}$, or $f(\boldsymbol{x}, \boldsymbol{\beta}^{(1)})$, is not too far from g_1; and when the true model is model 2, the response value at x, or $f(\boldsymbol{x}, \boldsymbol{\beta}^{(12.1)})$, is not too far from g_2. In other words, we want to select x such that both $f(\boldsymbol{x}, \boldsymbol{\beta}^{(1)}) - g_1$ and $f(\boldsymbol{x}, \boldsymbol{\beta}^{(12.1)}) - g_2$ are as small as possible. Mathematically this is equivalent to the following problem.

Choose $\boldsymbol{x}$ to minimize

$$\text{Max}\left[f(\boldsymbol{x}, \boldsymbol{\beta}^{(1)}) - g_1,\ f(\boldsymbol{x}, \boldsymbol{\beta}^{(2)}) - g_2 \right] . \qquad (12.1)$$

The difference $f(\boldsymbol{x}, \boldsymbol{\beta}^{(1)}) - g_1$ can be understood as the *regret* a rational decision maker will have if he

chooses this particular x when the true model is model 1, since g_1 is the minimum value the response can reach under model 1. Similarly the difference $f(\boldsymbol{x}, \boldsymbol{\beta}^{(2)})$ - g_2 is the *regret* a rational decision maker will have when the true model is model 2. Thus the aim of (12.1) is to choose an $\boldsymbol{x}$ to minimize the maximum regret over the two likely models.

12.2.2 Minimax Deviation Method when the Response Model Is Estimated from Data

Given the motivation in the previous section where the true model has two likely forms, we now consider the case where the response model is estimated from sample data; thus there are infinitely many forms that are likely the true model. Let the experimental data be $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2) \cdots (\boldsymbol{x}_n, y_n)$, where $\boldsymbol{x}_i$ contains predictor variables for the i^{th} observation and y_i is the corresponding response value. Suppose the true model is $y = f(\boldsymbol{x}, \boldsymbol{\beta})$where $\boldsymbol{\beta}$ contains the parameters we will fit using the experimental data. The estimate for $\boldsymbol{\beta}$, denoted by $\hat{\boldsymbol{\beta}}$, can be derived using the MLE or LSapproach. The estimated model using the point estimate $\hat{\boldsymbol{\beta}}$, or $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$, is only one of the many possible forms for the true model $f(\boldsymbol{x}, \boldsymbol{\beta})$. Statistical inference provides ways to construct a confidence region, rather than a single-point estimate, to cover the possible value for the true $\boldsymbol{\beta}$. Let us denote a confidence region for $\boldsymbol{\beta}$ by B; thus any model $f(\boldsymbol{x}, \boldsymbol{\beta})$, where $\boldsymbol{\beta} \in B$, represents a likely true model.

Figure 12.3 illustrates how robust optimization works by incorporating all of the estimates in the confidence region. The rectangle in Fig. 12.3 is the confidence region for $\boldsymbol{\beta}$ derived from the sample data, and the center point of the rectangle is the point estimate $\hat{\boldsymbol{\beta}}$. The usual canonical approach optimizes only a single model corresponding to the point estimate, or $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$. In contrast, robust optimization considers all of the possible estimates in the confidence region, so it optimizes all of the likely models $f(\boldsymbol{x}, \boldsymbol{\beta})$ whose $\boldsymbol{\beta}$ is in the rectangle.

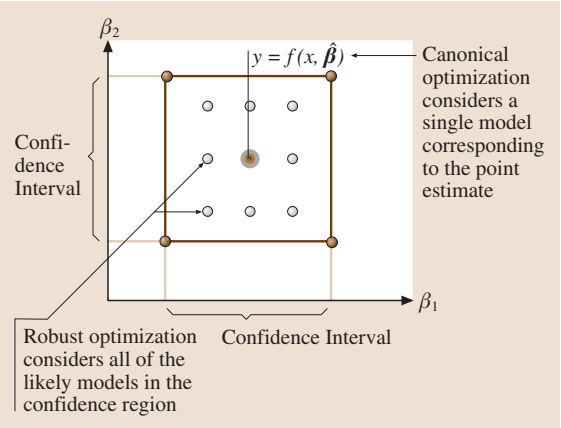

Fig. 12.3 Canonical optimization considers a single model while robust optimization considers all of the models with estimates in the confidence region

We now use the minimax deviation method in Sect. 12.2.1 to derive the robust solution where all of the likely models with estimates in the confidence region are considered. Suppose our goal is to minimize $f(\boldsymbol{x}, \boldsymbol{\beta})$ and the confidence region for $\boldsymbol{\beta}$ is B. The minimax deviation method can be formulated in the following equations:

$$\text{Min}_{\boldsymbol{x}} \text{Max}_{\boldsymbol{\beta} \in B} \left[f(\boldsymbol{x}, \boldsymbol{\beta}) - g(\boldsymbol{\beta}) \right] , \tag{12.2}$$

where

$$g(\boldsymbol{\beta}) = \text{Min}_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{\beta}), \text{ for any} \boldsymbol{\beta} \in B .$$

The interpretation of the minimax deviation method in (12.2) is similar to that given in Sect. 12.2.1. The difference $f(\boldsymbol{x}, \boldsymbol{\beta}) - g(\boldsymbol{\beta})$ is the regret incurred by choosing a particular $\boldsymbol{x}$ if the true coefficients in the model are $\boldsymbol{\beta}$. However the true values for $\boldsymbol{\beta}$ are unknown and they are likely at any point in the confidence region B. So $\text{Max}_{\boldsymbol{\beta} \in B} [f(\boldsymbol{x}, \boldsymbol{\beta}) - g(\boldsymbol{\beta})]$ stands for the maximum regret over the confidence region. We solve for the robust solution for $\boldsymbol{x}$ by minimizing the maximum regret over B.

The minimax deviation model in (12.2) is equivalent to the following mathematical program as in reference [12.10]

$$\begin{aligned} &\text{Min}(z) , \\ &f(\boldsymbol{x}, \boldsymbol{\beta}) - g(\boldsymbol{\beta}) \le z, \quad \forall \boldsymbol{\beta} \in B , \\ &g(\boldsymbol{\beta}) = \text{Min}_{\boldsymbol{x}} \left[f(\boldsymbol{x}, \boldsymbol{\beta}) \right] . \end{aligned} \tag{12.3}$$

The number of decision variables in this statement is finite while the number of constraints is infinite because every constraint corresponds to a point in the confidence region, or the rectangle in Fig. 12.3. Therefore the program in (12.3) is semi-infinite.

As illustrated in Fig. 12.3, we assume the confidence region can be constructed as a polytope. With this assumption, we have the following reduction theorem.

Reduction theorem. If B is a polytope and $f(\boldsymbol{x}, \boldsymbol{\beta})$ is linear in $\boldsymbol{\beta}$ then

$$\begin{aligned} &\text{Min}_{\boldsymbol{x}} \text{Max}_{\boldsymbol{\beta} \in B} \left[f(\boldsymbol{x}, \boldsymbol{\beta}) - g(\boldsymbol{\beta}) \right] \\ &= \text{Min}_{\boldsymbol{x}} \text{Max}_i \left[f(\boldsymbol{x}, \boldsymbol{\beta}^{\text{i}}) - g(\boldsymbol{\beta}^{\text{i}}) \right] , \end{aligned}$$

where $\boldsymbol{\beta}^1, \boldsymbol{\beta}^2 \cdots \boldsymbol{\beta}^m$ are the extreme points of B.

The reduction theorem says that the minimization of the maximum regret over the entire confidence region is equivalent to the minimization of the maximum regret over the extreme points of the confidence region. Figure 12.4 illustrates the use of the reduction theorem that reduces the original semi-infinite program in (12.3) to a finite program. The proof of the reduction theorem can be found in *Xu* and *Albin* [12.6].

12.2.3 Construction of the Confidence Region

One of the assumptions of the reduction theorem is that the confidence region for $\boldsymbol{\beta}$ is a polytope. This section introduces how we can construct a confidence region as a polytope.

A simple and straightforward way to construct a confidence polytope is to use simultaneous confidence intervals (*Miller* [12.11]). Suppose $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_p)$ and we want to construct a confidence polytope with a confidence level of $(1-\alpha)\times 100\%$ or more. First we construct a $(1-\alpha/p)\times 100\%$ confidence interval for each of the p coefficients in $\boldsymbol{\beta}$. Specifically, let I_i be the $(1-\alpha/p)\times 100\%$ confidence interval for β_i, or equivalently, $P(\beta_i \in I_i) = 1-\alpha/p$. Thus the simultaneous confidence intervals is the Cartesian product $B = I_1 \times I_2 \times \cdots \times I_p$. Using Bonferroni's inequality, we have

$$\begin{aligned} P(\boldsymbol{\beta} \in B) &= P\left(\beta_1 \in I_1, \beta_2 \in I_2, \cdots \beta_p \in I_p\right) \\ &\geq 1 - \sum_{i=1}^{p} P(\beta_i \notin I_i) \\ &= 1 - p \times \alpha/p = 1-\alpha\,. \end{aligned}$$

Therefore the confidence level of the simultaneous confidence intervals B is at least $(1-\alpha)\times 100\%$. Figure 12.5 illustrates the simultaneous confidence intervals in a two-dimensional space. Suppose the ellipsoid in the left panel of Fig. 12.5 is a 90% confidence region for (β_1, β_2). To construct simultaneous confidence intervals, we first identify the 95% confidence interval I_1 for β_1, and the 95% confidence interval I_2 for β_2; thus the rectangle $I_1 \times I_2$ is a confidence polytope for (β_1, β_2) with a confidence level of at least 90%. However, we know from Fig. 12.5 that the rectangle does not cover the 90% confidence ellipsoid very tightly, so the simultaneous confidence intervals are not the smallest confidence polytope at a certain confidence level. Clearly a better way to construct a more efficient confidence polytope is to find a rectangle that circumscribes the ellipsoid, such as that in the right panel of Fig. 12.5.

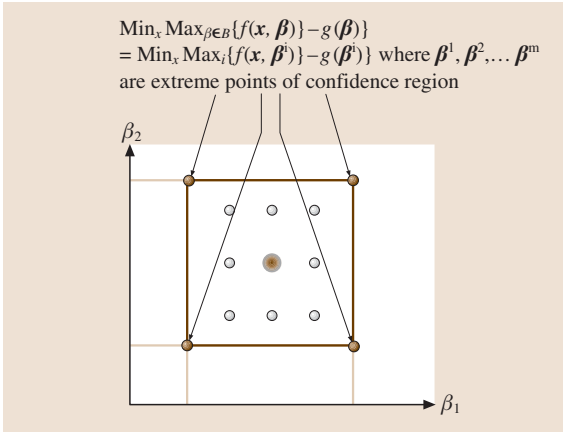

Fig. 12.4 The reduction theorem reduces the semi-infinite program over the entire confidence region to a finite program over the set of extreme points of the confidence region

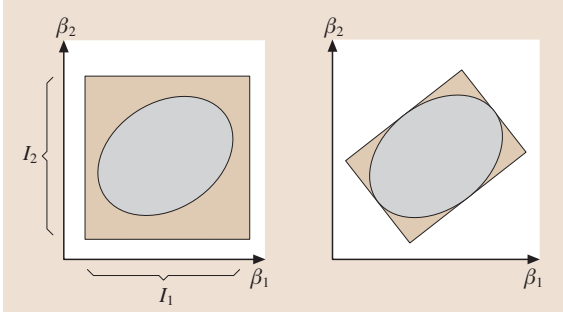

Fig. 12.5 Simultaneous confidence intervals are not the most efficient confidence polytope

We now present a transformation method to construct a tighter confidence polytope, which proves very effective to enhance robust optimization performance. Let $\boldsymbol{X}$ be a matrix with each row representing the observed values for the predictor variables in $\boldsymbol{x}$, and let $\boldsymbol{Y}$ be a vector with each element being the observed response value y.

From regression analysis, the $(1-\alpha)\times 100\%$ confidence region for $\boldsymbol{\beta}$ is an ellipsoid described as

$$\begin{aligned} &(1-\alpha)\times 100\% \text{ confidence region} \\ &= \left(\boldsymbol{\beta} \middle| \frac{(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})'(\mathbf{X}'\mathbf{X})(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{p\mathrm{MSE}} \leq F_{p,n-p,\alpha}\right), \end{aligned} \quad (12.4)$$

where $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ is the point estimator, p is the number of parameters we need to estimate in the response model, n is the total number of observations

we have in the sample data, $\text{MSE} = (\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})/n - p$ is the mean squared error, and $F_{p,n-p,\alpha}$ is the $(1-\alpha)\times 100$ percentile point for the F distribution with p and $(n-p)$ degrees of freedom. Details about (12.4) can be found in *Myers* [12.12].

We use Fig. 12.6 to illustrate the motivation for the transformation method to construct the confidence polytope in two dimensions. The ellipsoid in the left-hand picture of Fig. 12.6 is the $(1-\alpha)\times 100\%$ confidence region in (12.4). We want to find a polytope to cover the confidence ellipsoid more tightly. One such choice is to identify a rectangle with sides parallel to the major and minor axes of the ellipsoid, such as the one with vertices $\boldsymbol{\beta}^1$, $\boldsymbol{\beta}^2$, $\boldsymbol{\beta}^3$ and $\boldsymbol{\beta}^4$ in Fig. 12.6.

It is hard to identify these extreme points $\boldsymbol{\beta}^1$, $\boldsymbol{\beta}^2$, $\boldsymbol{\beta}^3$ and $\boldsymbol{\beta}^4$ directly in the original coordinate system $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$. However, by choosing appropriate algebraic transformation, the coordinate system $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$ can be transformed into the coordinate system (z_1, z_2), where the ellipsoid is converted to a unit ball in the right-hand picture of Fig. 12.6. In the coordinate system (z_1, z_2), it is easy to find a hypercube, with extreme points z^1, z^2, z^3 and z^4, to cover this ball tightly. We then map these extreme points back to the extreme points in $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$ to obtain $\boldsymbol{\beta}^1$, $\boldsymbol{\beta}^2$, $\boldsymbol{\beta}^3$ and $\boldsymbol{\beta}^4$.

To achieve this idea, we define the following transformation $\boldsymbol{\beta} \mapsto \mathrm{z}$:

$$\mathbf{z} = \boldsymbol{\Gamma}\,(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})\,, \text{ where } \Gamma = \frac{(\mathbf{X}'\mathbf{X})^{1/2}}{\sqrt{p \times \text{MSE} \times F_{p,n-p,\alpha}}}\,.$$

Through this transformation, the confidence ellipsoid in (12.4) in the coordinate system $\boldsymbol{\beta}$ can be converted into a unit ball in the coordinate system z: $\left(\mathrm{z}|\mathrm{z}'\mathrm{z} \le 1\right)$. It is easy to know that the hypercube covering the unit ball has extreme points $\mathrm{z}^i = (z_1, z_2, \cdots, z_p)$, where $z_j = \pm 1$, $j = 1, 2, \cdots, p$. By mapping these points back to the coordinate system $\boldsymbol{\beta}$, we can construct a confidence polytope with extreme points as follows:

$$\boldsymbol{\beta}^i = \hat{\boldsymbol{\beta}} + \boldsymbol{\Gamma}^{-1}\mathbf{z}^i, \text{ where } \Gamma = \frac{(\mathbf{X}'\mathbf{X})^{1/2}}{\sqrt{p \times \text{MSE} \times F_{p,n-p,\alpha}}}\,. \tag{12.5}$$

Thus the robust optimization model in (12.3) can be written as

$$\begin{aligned} &\text{Min}(z)\,, \\ &f(\boldsymbol{x}, \boldsymbol{\beta}^{\mathrm{i}}) - g(\boldsymbol{\beta}^{\mathrm{i}}) \le z\,, \\ &g(\boldsymbol{\beta}^{\mathrm{i}}) = \text{Min}_{\boldsymbol{x}}\, f(\boldsymbol{x}, \boldsymbol{\beta}^{\mathrm{i}})\,, \end{aligned} \tag{12.6}$$

where $\boldsymbol{\beta}^{\mathrm{i}}$ is given in (12.5)

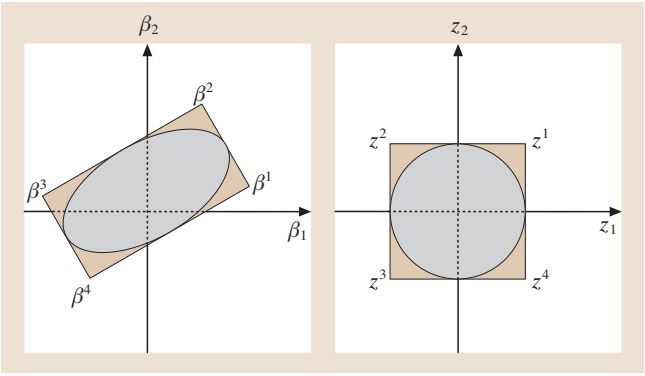

Fig. 12.6 Illustration of the transformation method to construct a confidence polytope

12.2.4 Monte Carlo Simulation to Compare Robust and Canonical Optimization

This section compares the performance of robust optimization and canonical optimization using Monte Carlo simulation on a hypothetical response model. Much of the material is from *Xu* and *Albin* [12.6] and *Xu* [12.13]. Suppose the true function relating performance response $y$and design variables x_1 and x_2 is the quadratic function

$$y = 0.5x_1^2 - x_1x_2 + x_2^2 - 2x_1 - 6x_2\,. \tag{12.7}$$

The objective is to identify x_1 and x_2 to minimize y with the constraints that $x_1 + x_2 \le 6$, $x_1 \ge 0$, and $x_2 \ge 0$. If the response model in (12.7) is known, the true optimal solution can be easily identified: $x_1 = 2.8$, $x_2 = 3.2$, yielding the optimal value $y = -19.6$.

Now suppose that the objective function is not known. We could perform a designed experiment to estimate the performance response function. Since we seek a second-order function we would perform a 3^2 factorial design with three levels for x_1 and three levels for x_2, resulting in a total of nine different combinations of $x_1$and x_2. The possible experimental values are -1, 0 and 1 for $x_1$and -1, 0, and 1 for x_2.

Instead of performing the experiment in a laboratory, we use Monte Carlo simulation, where the response y is produced by generating responses equal to the underlying response function in (12.7) plus noise ε:

$$\begin{aligned} &y = 0.5x_1^2 - x_1x_2 + x_2^2 - 2x_1 - 6x_2 + \varepsilon \\ &\quad \text{and } \varepsilon \sim N(0, \sigma^2)\,. \end{aligned} \tag{12.8}$$

Once the experiment has been run, we fit coefficients to the data by ordinary least-square regression and then optimize using the robust and canonical approaches, respectively.

The solutions obtained from the two approaches are inserted into (12.7) to determine the resulting performance response values and we compare these to determine which is closer to the true optimal.

We perform the above experiment and subsequent optimizations 100 times for each of the following degrees of experimental noise; that is, the noise term ε in (12.8) has standard deviation, σ, equal to 0.5, 1, 2, 3, or 4.

Thus we have 100 objective values for the canonical approach and 100 objective values for the robust approach for each value of σ. Table 12.4 gives the means and standard deviations of these performance responses using the canonical approach, the robust approach with simultaneous confidence intervals, and the robust approach with transformation method.

Table 12.4 shows that, when the experimental noise is small (σ= 0.5), yielding a relatively accurate point estimate of $\boldsymbol{\beta}$, the objective values given by the canonical approach are slightly closer to those given by the robust optimization approach. However, when the experimental noise is large (σ= 1,2,3,4), yielding a relatively inaccurate point estimate of $\boldsymbol{\beta}$, the robust approach yields results much closer to the true optimal than the canonical approach. We also notice that the robust approach using transformation method to construct the confidence polytope gives better results than the method using the simultaneous confidence intervals.

12.3 Weighted Robust Optimization

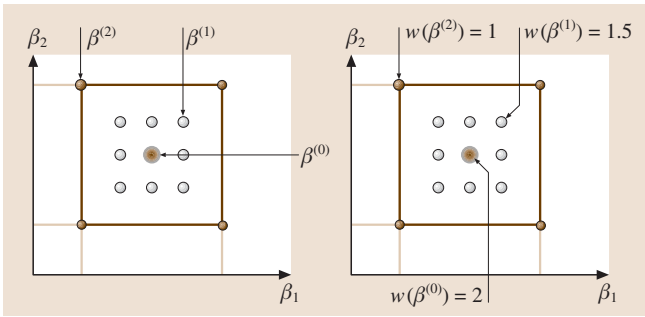

Fig. 12.7 Weighted robust optimization assigns weights to every point in the confidence region to reflect the likelihood of that point being close to the true $\boldsymbol{\beta}$

As we discussed earlier, robust optimization minimizes the maximum regret over a confidence region for the coefficients in the response model. Recall that the robust optimization is written as follows:

$$\mathrm{Min}_{\boldsymbol{x}}\mathrm{Max}_{\boldsymbol{\beta}\in B}\left[f(\boldsymbol{x},\boldsymbol{\beta})-g(\boldsymbol{\beta})\right]\ ,$$

where $g(\boldsymbol{\beta}) = \mathrm{Min}_{\boldsymbol{x}} f(\boldsymbol{x},\boldsymbol{\beta})$, for any $\boldsymbol{\beta}\in B$.

An implicit assumption in the minimax regret equation above is that all of the points in the confidence region B are treated with equal importance. For example, consider the three points $\boldsymbol{\beta}^{(0)}$, $\boldsymbol{\beta}^{(1)}$, and $\boldsymbol{\beta}^{(2)}$ in the left-hand picture of Fig. 12.7, the *regrets* we have at the three points by choosing $\boldsymbol{x}$ are $f(\boldsymbol{x},\boldsymbol{\beta}^{(0)})-g(\boldsymbol{\beta}^{(0)})$, $f(\boldsymbol{x},\boldsymbol{\beta}^{(1)})-g(\boldsymbol{\beta}^{(1)})$ and $f(\boldsymbol{x},\boldsymbol{\beta}^{(2)})-g(\boldsymbol{\beta}^{(2)})$, respectively. However, we know from statistical inference that the center point $\boldsymbol{\beta}^{(0)}$ is more likely close to be the true $\boldsymbol{\beta}$ than $\boldsymbol{\beta}^{(1)}$, and $\boldsymbol{\beta}^{(1)}$ is more likely to be close to the true $\boldsymbol{\beta}$ than$\boldsymbol{\beta}^{(2)}$, so in the *regret* calculation, weights can be assigned to each point in the confidence region to measure how likely that point is to be close to the true $\boldsymbol{\beta}$.

In the right-hand picture of Fig. 12.7, the weights for the three points are $w\left(\boldsymbol{\beta}^{0}\right)=2$, $w\left(\boldsymbol{\beta}^{(1)}\right)=1.5$, and $w\left(\boldsymbol{\beta}^{(2)}\right)=1$, so the *regrets* at these three points can be defined as $2\left[f\left(\boldsymbol{x},\boldsymbol{\beta}^{(0)}\right)-g\left(\boldsymbol{\beta}^{(0)}\right)\right]$, $1.5\left[f\left(\boldsymbol{x},\boldsymbol{\beta}^{(1)}\right)-g\left(\boldsymbol{\beta}^{(1)}\right)\right]$ and $\left[f\left(\boldsymbol{x},\boldsymbol{\beta}^{(2)}\right)-g\left(\boldsymbol{\beta}^{(2)}\right)\right]$. In general, the weighted robust optimization can be written as

$$\mathrm{Min}_{\boldsymbol{x}}\mathrm{Max}_{\boldsymbol{\beta}\in B}\left[f(\boldsymbol{x},\boldsymbol{\beta})-g(\boldsymbol{\beta})\right]w(\boldsymbol{\beta})\ , \qquad (12.9)$$

where $w(\boldsymbol{\beta})$ is the weight assigned to the point $\boldsymbol{\beta}$ in the confidence region. So the aim of the weighted robust optimization in (12.9) is to minimize the maximum weighted regret over the confidence region. The center point of the confidence region should be assigned the largest weight since it is most likely to be close to the true $\boldsymbol{\beta}$. On the other hand, the extreme points of the confidence region should be assigned the smallest weights.

We now consider two choices of the weight function $w(\boldsymbol{\beta})$. Let $\boldsymbol{\beta}^{(0)}$ be the center point of the confidence region; let $\boldsymbol{\beta}^{(+)}$ be an extreme point with the largest distance to $\boldsymbol{\beta}^{(0)}$. In the first version of weight function, we treat the point $\boldsymbol{\beta}^{(0)}$ as twice as important as $\boldsymbol{\beta}^{(+)}$. In other words, we assign weight 1 to the extreme point $\boldsymbol{\beta}^{(+)}$ and the weight for the center point $\boldsymbol{\beta}^{(0)}$ is 2. The weight for any other point $\boldsymbol{\beta}$ is between 1 and 2 and decreases linearly with its distance from the center point $\boldsymbol{\beta}^{(0)}$. This linear-distance-based weight function can be

written in the following way:

$$w(\boldsymbol{\beta}) = 2 - \frac{||\boldsymbol{\beta} - \boldsymbol{\beta}^{(0)}||}{||\boldsymbol{\beta}^{(+)} - \boldsymbol{\beta}^{(0)}||} . \tag{12.10}$$

We now discuss the second version of weight function. Let $\boldsymbol{x}^{(i)}$and y_i, $i = 1, 2, \cdots, n$, be the observation for the predictors and response value. For any estimator $\boldsymbol{\beta}$ in the confidence region, the sum of squared errors (SSE) $\sum_{i=1}^{n}[\,\boldsymbol{y}_i - f(\boldsymbol{x}^{(i)}, \boldsymbol{\beta})]^2$ can be viewed as an indirect measure of how close the estimator $\boldsymbol{\beta}$ is to the true coefficients. So we take the reciprocal of the SSE as the weight function, or

$$w(\boldsymbol{\beta}) = \frac{1}{\sum_{i=1}^{n}[\,\boldsymbol{y}_i - f(\boldsymbol{x}^{(i)}, \boldsymbol{\beta})]^2} . \tag{12.11}$$

To compare the performance of robust optimization and weighted robust optimization, we use the same underlying response model in (12.7) to generate the experimental data and then apply the two approaches to derive the robust solutions. Table 12.5 contains the means and standard deviations of the performance responses obtained by the canonical, robust and weighted robust optimization approaches.

It is clear that the performance of the weighted robust optimization dominates that of the standard robust optimization. We further note that the weight function in (12.10) performs better than the weight function in (12.11) when the experimental noise is large ($\sigma = 1, 2, 3, 4$).

Although the weighted robust optimization gives better results, computationally it is much harder and more challenging. Unfortunately, weighting the points in the confidence region, using weight functions $w(\boldsymbol{\beta})$ in (12.10) or (12.11), results in an optimization problem in (12.9) with an objective function that is not linear in $\boldsymbol{\beta}$. Consequently, the reduction theorem is no longer applicable to (12.9) to reduce the optimization problem to a finite program. Therefore a numerical algorithm has to be designed to solve the weighted robust optimization problem in (12.9).

For simplicity, let $F(\boldsymbol{x}, \boldsymbol{\beta}) = [\,f(\boldsymbol{x}, \boldsymbol{\beta}) - g(\boldsymbol{\beta})]\, w(\boldsymbol{\beta})$. Thus we can write the weighted robust optimization problem as follows

$$\text{Min}_{\boldsymbol{x}}\text{Max}_{\boldsymbol{\beta}\in B} F(\boldsymbol{x}, \boldsymbol{\beta}) \tag{12.12}$$

or equivalently,

$$\begin{aligned} &\text{Min}_{\boldsymbol{x}}\{\xi\} , \\ &\text{s.t. x} \in X , \\ &F(\boldsymbol{x}, \boldsymbol{\beta}) \le \xi, \forall \boldsymbol{\beta} \in B . \end{aligned} \tag{12.13}$$

We use the Shimizu–Aiyoshi relaxation algorithm to solve (12.13). For a rigorous treatment of this relaxation algorithm, see *Shimizu* and *Aiyoshi* [12.10]. The main steps in this algorithm are given as follows:

Step 1: choose any initial point $\boldsymbol{\beta}^{(1)}$. Set $k = 1$.

Step 2: solve the following relaxed problem of (12.13):

$$\begin{aligned} &\text{Min}_{\boldsymbol{x}}\{\xi\} \\ &\text{s.t.}\, \boldsymbol{x} \in X \\ &F(\boldsymbol{x}, \boldsymbol{\beta}^{(i)}) \le \xi, i = 1, 2, \cdots, k \end{aligned} \tag{12.14}$$

Obtain an optimal solution $(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^{(k)})$ for (12.14). The $\boldsymbol{\xi}^{(k)}$ is also the optimal value for (12.14). We note that $\boldsymbol{\xi}^{(k)}$ is a lower bound on the optimal value for (12.13).

Step 3: solve the maximization problem:

$$\text{Max}_{\boldsymbol{\beta}\in B} F(\boldsymbol{x}^{(k)}, \boldsymbol{\beta}) . \tag{12.15}$$

Obtain an optimal solution $\boldsymbol{\beta}^{(k+1)}$ and the maximal value $\phi(\boldsymbol{x}^{(k)}) = F(\boldsymbol{x}^{(k)}, \boldsymbol{\beta}^{(k+1)})$. We note that $\phi(\boldsymbol{x}^{(k)})$ is an upper bound on the optimal value of (12.12) or (12.13).

Step 4: If $\phi(\boldsymbol{x}^{(k)}) - \boldsymbol{\xi}^{(k)} < \varepsilon$, terminate and report the solution $\boldsymbol{x}^{(k)}$; otherwise, set $k = k+1$ and go back to step 2.

We now introduce the method for solving the optimization problems (12.14) and (12.15). First, we address (12.14). If the response model $f(\boldsymbol{x}, \boldsymbol{\beta})$ is linear in $\boldsymbol{x}$, then $F(\boldsymbol{x}, \boldsymbol{\beta})$ is also linear in $\boldsymbol{x}$; thus (12.14) is a linear programming problem if we assume that the feasible region $\boldsymbol{X}$ contains only linear constraints. If the response model $f(\boldsymbol{x}, \boldsymbol{\beta})$ is quadratic in $\boldsymbol{x}$, then $F(\boldsymbol{x}, \boldsymbol{\beta})$ is also quadratic in $\boldsymbol{x}$; thus (12.14) is a quadratically constrained quadratic programming problem.

Quadratically constrained quadratic programming (QCQP) is a very challenging problem. One efficient way to solve QCQP is to approximate it by a semidefinite program (SDP) and the solution for SDP usually provides a very tight bound on the optimal value of the QCQP. There exist very efficient and powerful methods to solve SDP and numerous software packages have been developed. After an approximate solution is obtained from SDP, we then use a randomized algorithm to search for a good solution for the original QCQP. For a comprehensive introduction to SDP, see *Vandenberghe* and *Boyd* [12.14]; for the connection between QCQP and SDP, see *Alizadeh* and *Schmieta* [12.15], and *Frazzoli* [12.16]; for software packages to solve SDP, see *Alizadeh* et al. [12.17], and *Sturm* [12.18].

We finally comment on the optimization problem (12.15). The objective function in (12.15) is

$F(\boldsymbol{x}^k, \boldsymbol{\beta}) = \{f(\boldsymbol{x}^k, \boldsymbol{\beta}) - g(\boldsymbol{\beta})\}w(\boldsymbol{\beta})$. We note that the function $g(\boldsymbol{\beta})$ has no closed-form expression since it is the minimal value of $f(\boldsymbol{x}, \boldsymbol{\beta})$. Here we use a powerful global optimization algorithm, called DIRECT, to solve (12.15). The DIRECT algorithm was proposed by *Jones* et al. [12.19]. There are several advantages of using DIRECT: it is a global optimization algorithm and has a very good balance between global searching and local searching, and it converges quite quickly; it does not need derivative information on the objective function. For software on DIRECT, see *Bjorkman* and *Holmstrom* [12.20].

12.4 The Application of Robust Optimization in Parameter Design

This section applies robust optimization to solve Taguchi's parameter design problem. The aim of parameter design is to choose optimal levels for the control factors to reduce the performance variation as well as to make the response close to the target. Section 12.4.1 introduces both traditional and more recent approaches to handling parameter design problems. Section 12.4.2 discusses how to use robust optimization to identify a robust solution for control factors when the response model is estimated from experimental data. Section 12.4.3 presents the robust optimization method to solve parameter design problem when the experimental data is from a fractional factorial design and some effects are aliased.

12.4.1 Response Model Approach to Parameter Design Problems

Parameter design was promoted by Genichi Taguchi in the 1950s and has since been widely used in quality engineering (*Taguchi* [12.21]). In parameter design, there are two sets of variables: control factors and noise variables. Control factors are those variables that can be set at fixed levels in the production stage; noise variables are those variables that we cannot control such as environmental conditions and material properties, and are hence assumed random in the production stage. The performance response is affected by both control factors and noise variables.

The contribution of Taguchi, among many others, is to recognize that interaction often exists between control factors and noise variables. Hence, appropriate levels of control factors can be selected to reduce the impact of noise variables on the performance response.

Taguchi proposed a set of methodologies, including inner–outer array design and signal-to-noise ratio (SNR), to identify optimal levels of control factors. *Welch* et al. [12.22] and *Shoemaker* et al. [12.23] have proposed the response model formulation, a more statistically sound method, to deal with parameter design problems.

In the response model approach, we first conduct experiments at appropriate levels of control factors and noise variables. Then we can fit the following model to relate the performance response to both control factors and noise variables:

$$\begin{aligned} y &= f(\boldsymbol{x}; \alpha, \gamma, \mu, A, \Delta) \\ &= \mu + \tfrac{1}{2}\mathrm{x}'A\mathrm{x} + \alpha'x + x'\Delta\mathrm{z} + \gamma'z + \varepsilon\,, \end{aligned} \tag{12.16}$$

where $\boldsymbol{x}$ represents the control factors and $\boldsymbol{z}$ the noise variables. Equation (12.16) includes first- and second-order terms in control factors, a first-order term in noise variables and an interaction term between control factors and noise variables. The noise variables $\boldsymbol{z}$ have a normal distribution with mean $\boldsymbol{0}$ and variance Σ_z, or $z \sim N(0, \Sigma_z)$. The ε term incorporates unidentified noise other than $\boldsymbol{z}$.

The difference between the response model in (12.16) and the response model $f(\boldsymbol{x}, \boldsymbol{\beta})$ in the previous sections is that the former divides the noise into $\boldsymbol{z}$ and ε and introduces a first-order term and an interaction term related to $\boldsymbol{z}$ while the latter has the noise only in ε. We further note that the coefficients $(\alpha, \gamma, \mu, A, \Delta)$ in (12.16) are estimated from designed experiments. More details about (12.16) can be found in *Myers* and *Montgomery* [12.3].

From (12.16), it is easy to derive the expected value and standard deviation of the response value y

$$\begin{aligned} E(y) &= \mu + \tfrac{1}{2}x'Ax + \alpha'x\,, \\ \mathrm{Var}(y) &= x'\Delta\Sigma_z\Delta'x + \gamma'\Sigma_z\gamma + \sigma_\varepsilon^2\,. \end{aligned}$$

Suppose our goal is to choose control factors $\boldsymbol{x}$ such that the response y is as close as possible to a target t. In Taguchi's parameter design problem, the criterion to identify optimal levels for control factors $\boldsymbol{x}$ is to minimize the following expected squared loss:

$$\begin{aligned} L(\boldsymbol{x}; \alpha, \gamma, \mu, A, \Delta) &= [E(y) - t]^2 + \mathrm{Var}(y) \\ &= \left(\mu + \tfrac{1}{2}x'Ax + \alpha'x - t\right)^2 \\ &\quad + x'\Delta\Sigma_z\Delta'x + \gamma'\Sigma_z\gamma + \sigma_\varepsilon^2\,. \end{aligned}$$

12.4.2 Identification of Control Factors in Parameter Design by Robust Optimization

Since the true values for the coefficients $(\alpha, \gamma, \mu, A, \Delta)$ in (12.16) are unknown and they are estimated from data, we use robust optimization to derive a robust solution $\boldsymbol{x}$ that is resistant to the estimation error. First we use the same method as in Sect. 12.2.3 to construct a confidence region B for the coefficients $(\alpha, \gamma, \mu, A, \Delta)$ and then we solve the following minimax deviation model:

$$\begin{aligned}
&\mathrm{Min}_{\boldsymbol{x}}\mathrm{Max}_{(\alpha,\gamma,\mu,A,\Delta)\in B} L(\boldsymbol{x};\alpha,\gamma,\mu,A,\Delta)\\
&-g(\alpha,\gamma,\mu,A,\Delta)\,,\\
&g(\alpha,\gamma,\mu,A,\Delta)=\mathrm{Min}_{\boldsymbol{x}} L(\boldsymbol{x};\alpha,\gamma,\mu,A,\Delta)\,,\\
&L(\boldsymbol{x};\alpha,\gamma,\mu,A,\Delta)=\left(\mu+\tfrac{1}{2}x'Ax+\alpha'x-t\right)^2\\
&+x'\Delta\Sigma_z\Delta'x+\gamma'\Sigma_z\gamma+\sigma_\varepsilon^2\,. \qquad (12.17)
\end{aligned}$$

Note that the model (12.17) is not linear in the coefficients $(\alpha, \gamma, \mu, A, \Delta)$, so we have to resort to a numerical optimization algorithm to solve it.

We use the following example from *Xu* [12.13] to show the application of robust optimization to the parameter design problem. Suppose there are two control factors A and B and one noise variable C. The underlying relationship between the performance response $y$and control/noise factors A, B and C is

$$y=3A+2B+0.15C+0.5AC-BC+6+\varepsilon\,, \qquad (12.18)$$

where $\varepsilon\sim N(0,1)$. We further assume that the variance of the noise factor C is $\sigma_C=1$. Our goal is to choose the optimal levels of (A, B) over the feasible region $\{(A, B)|-1\le A, B\le 1\}$ to make the response y close to the target $t=7.5$. We notice that the response values have a minimum squared loss of 1 when the control factors $(A, B)=(0.3, 0.3)$.

Assume we do not know the true model (12.18), so we have first to fit a response model $y=\mu+\alpha_1A+\alpha_2B+\gamma_1C+\delta_1AC+\delta_2BC$, where $(\mu, \alpha_1, \alpha_2, \gamma_1, \delta_1, \delta_2)$ are the coefficients we will estimate. Suppose we perform a full 2^3 factorial design with the design matrix and the observed responses as follows.

Using the experimental data from the factorial design in Table 12.3, we first construct the confidence region B for the coefficients $(\mu, \alpha_1, \alpha_2, \gamma_1, \delta_1, \delta_2)$ in the response model. The squared loss for the response value y is $L=[E(y)-t]^2+\mathrm{Var}(y)=(\mu+\alpha_1A+\alpha_2B-t)^2+(\gamma_1+\delta_1A+\delta_2B)^2\sigma_C^2+\sigma_\varepsilon^2$. By substituting $t=7.5$ and $\sigma_C=\sigma_\varepsilon=1$, we have $L=(\mu+\alpha_1A+\alpha_2B-7.5)^2+(\gamma_1+\delta_1A+\delta_2B)^2+1$. Therefore we can write the robust optimization model as follows:

$$\begin{aligned}
&\mathrm{Min}_{\{-1\le A,B\le 1\}}\mathrm{Max}_{(\mu,\alpha_1,\alpha_2,\gamma_1,\delta_1,\delta_2)\in B}\\
&L(A,B;\mu,\alpha_1,\alpha_2,\gamma_1,\delta_1,\delta_2)\\
&-g(\mu,\alpha_1,\alpha_2,\gamma_1,\delta_1,\delta_2)\,,\\
&g(\mu,\alpha_1,\alpha_2,\gamma_1,\delta_1,\delta_2)\\
&=\mathrm{Min}_{(A,B)}L(A,B;\mu,\alpha_1,\alpha_2,\gamma_1,\delta_1,\delta_2)\,,\\
&L(A,B;\mu,\alpha_1,\alpha_2,\gamma_1,\delta_1,\delta_2)\\
&=(\mu+\alpha_1A+\alpha_2B-7.5)^2\\
&+(\gamma_1+\delta_1A+\delta_2B)^2+1\,. \qquad (12.19)
\end{aligned}$$

By solving the optimization problem in (12.19), we can obtain the robust solution $(A, B) = (0.35, 0.22)$. If this solution is applied to the underlying model (12.18), the response values would have an expected squared loss 1.01, which is quite close to the true minimum 1. To be complete, we also present the results obtained by canonical optimization. The canonical solution is $(A, B) = (0.29, 0.95)$ and if this solution is applied to the true model, the expected squared loss would be 3.08, which is much worse than the true optimum.

12.4.3 Identification of Control Factors when the Response Model Contains Alias Terms

Fractional factorial design is a widely used tool to reduce the number of runs in experimental design. The downside of fractional factorial design is that the main effects and higher-order interactions are confounded. For example, in a fractional factorial design with resolution III, the main effects are aliased with the two-factor interaction in the response model. A usual way to address this question is to assume that the interaction is zero and attribute all effects to the main factors. However if the interaction term is important to determine the process/product performance, the loss of this information may be critical. If in the parameter design we cannot differentiate between the effects from the main factors and those from the interaction terms, there is no easy way to identify the optimal levels for the control factors to minimize the variance in the final performance response.

Fractional factorial design usually is used for factor-screening purposes, however if we can use the data from fractional design to make a preliminary assessment of where the optimal levels for control factors may be located, this can help move the design more quickly to the region where the final performance response is most

likely to be optimal and start the full factorial design or other sophisticated designs sooner. So this poses the challenge of how we can solve a parameter design problem if two effects are aliased due to the nature of the data from a fractional factorial design.

Robust optimization provides a useful methodology to address the above challenge if we can include prior information on the alias terms. For example, the prior information can be that both the main factor and interaction term contribute positively to the response value, etc.. To be clear, let us consider the same response model as in (12.18), but assume that, instead of the full factorial design in Table 12.3, only the data from a fractional factorial design is available. The 2^{3-1} design is shown in Table 12.4 where we retain the observations 1, 4, 6, 7 from Table 12.3. At each design point in Table 12.4, replicate 1 is the response value we observed from the design in Table 12.3, in addition, we perform one more run of the experiments and replicate 2 contains the corresponding response value.

We note that the design in Table 12.4 has the defining relation $ABC = I$, so the effects of the main factor A and the interaction BC cannot be differentiated using the data in Table 12.4; similarly the effects of the main factor B and the interaction AC are confounded too. Hence instead of estimating the response model in (12.18), we can only use the data in Table 12.4 to estimate the following model:

$$y = \mu + \boldsymbol{\beta}_1 \tilde{A} + \boldsymbol{\beta}_2 \tilde{B} + \gamma_1 C \,, \tag{12.20}$$

where $\tilde{A} = A + BC$, $\boldsymbol{\beta}_1$ measures the combined effect of the factors A and BC, $\tilde{B} = B + AC$, $\boldsymbol{\beta}_2$ measures the combined effect of the factors B and AC.

Using the same notation as in Sect. 12.4.2, let

$\alpha_1 =$ denote the effect of the main factor A,
$\alpha_2 =$ denote the effect of the main factor B,
$\delta_1 =$ denote the effect of the interaction term AC,
$\delta_2 =$ denote the effect of the interaction term BC.

Given the values for $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$, if there is no other information, α_1 and δ_2 can be any values as long as they satisfy $\alpha_1 + \delta_2 = \boldsymbol{\beta}_1$; similarly, α_2 and δ_1 can be any values as long as they satisfy $\alpha_2 + \delta_1 = \boldsymbol{\beta}_2$. However we assume here that quality engineers already know the prior information that: (1) the effects of the main factor A and the interaction BC are in the same direction; and (2) the effects of the main factor B and the interaction AC are in the opposite direction. We can describe the prior information in (1) and (2) in the following constraints:

$$\alpha_1 = \lambda_1 \boldsymbol{\beta}_1, \quad \delta_2 = (1 - \lambda_1)\boldsymbol{\beta}_1, \quad 0 \le \lambda_1 \le 1 \,, \tag{12.21}$$

$$\alpha_2 = \lambda_2 \boldsymbol{\beta}_2, \quad \delta_1 = (1 - \lambda_2)\boldsymbol{\beta}_2, \quad \lambda_2 \ge 1 \,. \tag{12.22}$$

Table 12.4 Comparison of performance responses using canonical and robust optimization approaches (true optimal performance: -19.6)

Dist. of ε	Canonical approach		Robust approach with simultaneous confidence intervals		Robust approach with transformation method	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
$N(0, 0.5)$	-18.7	1.2	-18.2	1.5	-18.4	1.6
$N(0, 1)$	-15.4	6.0	-15.2	3.3	-17.0	3.5
$N(0, 2)$	-9.9	8.7	-10.8	4.9	-15.0	5.4
$N(0, 3)$	-6.3	9.3	-9.0	5.4	-13.2	6.3
$N(0, 4)$	-4.6	9.0	-7.8	5.7	-11.4	6.9

Table 12.5 Comparison of performance responses using canonical, robust, and weighted robust optimization (adapted from [12.13])

ε	Canonical optimization		Robust optimization		Weighted robust opt. with weights (12.10)		Weighted robust opt. with weights (12.11)	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
$N(0, 0.5)$	-18.7	1.2	-18.4	1.6	-18.4	1.4	-18.8	1.0
$N(0, 1)$	-15.4	6.0	-17.0	3.5	-18.0	1.9	-17.8	2.1
$N(0, 2)$	-9.9	8.7	-15.0	5.4	-17.4	2.8	-16.4	3.7
$N(0, 3)$	-6.3	9.3	-13.2	6.3	-17.2	3.0	-15.3	4.8
$N(0, 4)$	-4.6	9.0	-11.4	6.9	-17.0	3.7	-14.7	5.4

We first construct the confidence region B for $(\mu, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \gamma_1)$, the parameters in the response model (12.20). By substituting (12.21) and (12.22) into the optimization problem in (12.19), we have the following equations:

$$\begin{aligned}
&\mathrm{Min}_{\{-1\le A,B\le 1\}}\mathrm{Max}_{(\mu,\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\gamma_1)\in B}\\
&L(A,B;\mu,\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\gamma_1)-g(\mu,\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\gamma_1)\,,\\
&g(\mu,\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\gamma_1)\\
&=\mathrm{Min}_{(A,B)}L(A,B;\mu,\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\gamma_1)\,,\\
&L(A,B;\mu,\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\gamma_1)\\
&=[\mu+\lambda_1\boldsymbol{\beta}_1A+\lambda_2\boldsymbol{\beta}_2B-7.5]^2\\
&\quad+[\gamma_1+(1-\lambda_2)\boldsymbol{\beta}_2A+(1-\lambda_1)\boldsymbol{\beta}_1B]^2+1\,,\\
&0\le\lambda_1\le 1,\ \lambda_2\ge 1\,. \qquad (12.23)
\end{aligned}$$

By solving the optimization problem in (12.23), we will get the solution $(A, B) = (0.17, 0.36)$ with the expected squared loss 1.089. Although this solution seems a little off from the true optimal solution (0.3, 0.3), it still provides valuable information and can guide the design to move quickly to the region closer to the true optimal solution even in the early stage that only the data from the fractional factorial design is available.

We finally comment on the use of the prior information on the main factor effect and the higher-order interaction effect in the formulation of the robust optimization model in (12.23). This information is usually available based on the qualitative knowledge and reasonable judgment of quality engineers. If this information is not available, that is, the values for λ_1 and λ_2 in (12.23) can take any real numbers, we believe robust optimization will not be able to yield a good solution.

References

12.1 G. E. P. Box, W. G. Hunter, J. S. Hunter: *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building* (Wiley, New York 1978)

12.2 G. E. P. Box, N. R. Draper: *Empirical Model-Building and Response Surfaces* (Wiley, New York 1987)

12.3 R. H. Myers, D. C. Montgomery: *Response Surface Methodology: Process and Product Optimization Using Designed Experiments* (Wiley, New York 1995)

12.4 A. I. Khuri, J. A. Cornell: *Response Surfaces: Designs and Analyses* (Marcel Dekker, New York 1996)

12.5 G. E. P. Box, K. B. Wilson: On the experimental attainment of optimum conditions, J. R. Stat. Soc. Ser. B **13**, 1–45 (1951)

12.6 D. Xu, S. L. Albin: Robust optimization of experimentally derived objective functions, IIE Trans. **35**, 793–802 (2003)

12.7 C. F. Wu, M. Hamada: *Experiments: Planning, Analysis, And Parameter Design Optimization* (Wiley, New York 2000)

12.8 E. Erhardt, H. Mai: The search for the optimal paper helicopter, personal communication, (2002) http://www.stat.unm.edu/~erike/projects/Erhardt_Erik_rsmproj.pdf

12.9 G. E. P. Box, P. Y. T. Liu: Statistics as a catalyst to learning by scientific method, J. Qual. Technol. **31**, 1–15 (1999)

12.10 K. Shimizu, E. Aiyoshi: Necessary conditions for Min-Max problems and algorithms by a relaxation procedure, IEEE Trans. Autom. Control **25**, 62–66 (1980)

12.11 R. G. Miller: *Simultaneous Statistical Inference* (McGraw-Hill, New York 1966)

12.12 R. H. Myers: *Classical and Modern Regression with Applications* (PWS-Kent, Boston 1990)

12.13 D. Xu: Multivariate statistical Modeling and robust optimization in quality engineering, Doctoral Dissertation, Department of Industrial and Systems Engineering, Rutgers University (2001)

12.14 L. Vandenberghe, S. Boyd: Semidefinite programming, SIAM Rev. **38**, 49–95 (1996)

12.15 F. Alizadeh, S. Schmieta: Optimization with semidefinite, quadratic and linear constraints, Rutcor Res. Rep., Rutgers University , 23–97 (1997)

12.16 E. Frazzoli, Z. H. Mao, J. H. Oh, E. Feron: Resolution of conflicts involving many aircraft via semidefinite programming, MIT Research Report , MIT–ICAT 99–5 (1999)

12.17 F. Alizadeh, J. P. Haeberly, M. Nayakkankuppam, M. Overton, S. Schmieta: SDPPACK User's Guide. NYU Computer Science Department Technical Report (1997)

12.18 J. Sturm: Using SeDuMi 1.0x, a Matlab toolbox for optimization over symmetric cones, Optim. Methods Softw. **11**, 625–663 (1999)

12.19 D. Jones, C. Perttunen, B. Stuckman: Lipschitzian optimization without the Lipschitz constant, J. Opt. Theory Appl. **79**, 157–181 (1993)

12.20 M. Bjorkman, K. Holmstrom: Global optimization using the DIRECT algorithm in Matlab, Adv. Model. Opt. **1**, 17–37 (1999)

12.21 G. Taguchi: *System of Experimental Design* (Unipub/Kraus, White Plains 1987)

12.22 W. J. Welch, T. K. Yu, S. M. Kang, J. Sacks: Computer experiments for quality control by robust design, J. Qual. Technol. **22**, 15–22 (1990)

12.23 A. C. Shoemaker, K. L. Tsui, C. F. J. Wu: Economical experimentation methods for robust parameter design, Technometrics **33**, 415–428 (1991)