UNIVERSITÀ DEGLI STUDI DI BARI
Dottorato di Ricerca in Matematica
XXI Ciclo – A.A. 2007/2008
Settore Scientifico-Disciplinare:
MAT/09 – Ricerca Operativa

Tesi di Dottorato

Robust Simulation-Optimization Methods using Kriging Metamodels

Candidata:
Gabriella Dellino

Supervisori della tesi:
Prof. Carlo Meloni
Prof. Tiziano Politi
Prof. Jack P.C. Kleijnen

Coordinatore del Dottorato di Ricerca:
Prof. Luciano Lopez

Acknowledgements

Now that I become closer and closer to the end of this adventure, I would like to deeply thank some people who helped me — in different ways and at different stages of my PhD — to achieve this goal.

First of all, I want to thank my advisors: Carlo Meloni, for having introduced me to the fascinating world of research, for his being always so supportive to me, and for his willingness to discuss some research issues, going on until a suitable solution was found. Tiziano Politi, for having provided me the important background I needed to tackle the topic that, afterwards, evolved into the present dissertation. Jack Kleijnen, for having given me the opportunity to visit Tilburg University and personally collaborate with him; thanks to our intensive and productive meetings, I learned so much during the year I spent working there.

I would also like to thank my former room-mate, Wim Van Beers, with whom I shared the office during my stay in Tilburg: he was always available to help me, no matter whether he was busy or not; I will never forget what he told me one day I felt a bit frustrated for my research having temporarily come to a stop: "Keep smiling!". I will...
Many thanks also to Dick den Hertog and Inneke Van Nieuwenhuyse for the stimulating and thorough discussions which definitely improved the outcome of my work.

Besides the results discussed in this thesis, I cooperated in some research projects at the Politecnico di Bari: therefore, I wish to thank Alessandro Rizzo e Paolo Lino, who — together with Carlo Meloni — have been my first research group and initiated me into my PhD, always making me feel at ease.

I am grateful to the referees of my dissertation, Alessandro Agnetis, Bill Biles and Renato De Leone, who gave me valuable remarks to improve the quality of my PhD thesis; their comments really encouraged me to keep on working on this topic.

Finally, I want to thank my parents, for their discreet support and for giving me all I needed to concentrate on work whenever I had to. Thanks to my friends for their perseverance in inviting me to have fun together, despite all the times I answered 'no'. In the end, I heartily thank my fiancé, Giulio, for his infinite love, his support and encouragement and for always being by my side every time I need.

To Giulio,
the love of my life

Contents

Introduction

Simulation-optimization aims to identify the setting of the input parameters of a simulated system leading to optimal system performances. In practice, however, some of these parameters cannot be perfectly controlled — due to measurement errors or other implementation issues, and because of the inherent uncertainty caused by fluctuations in the environment (e.g. temperature or pressure in physical and chemical processes or demand in inventory problems). Consequently, the classic optimal solution — derived ignoring these sources of uncertainty — may turn out to be sub-optimal or even infeasible. Robust optimization (RO) offers a way to tackle this class of problems, with the purpose of deriving solutions that are relatively insensitive to perturbations caused by the so-called noise factors.

According to the 2006 report of the National Science Foundation, "the development of reliable methodologies – algorithms, data acquisition and management procedures, software, and theory – for quantifying uncertainty in computer predictions stands as one of the most important and daunting challenges in advancing simulation-based engineering science" (Oden, 2006). In practical engineering and management applications, there exist two different types of uncertainties: aleatory and epistemic uncertainties (see Swiler and Giunta, 2007). *Aleatory uncertainty* — also referred to as variability, stochastic uncertainty, irreducible uncertainty, or Type A uncertainty — is caused by the inherent randomness in the behaviour of the system under study; it can be caused by incomplete model input information and is irreducible except through design modifications. Methods to account for aleatory uncertainty into simulations deal with uncertain input variables that can be specified through a probability distribution. *Epistemic uncertainty* — also called state of knowledge uncertainty, subjective uncertainty, reducible uncertainty, and Type B uncertainty — characterizes the lack of knowledge about the appropriate value to use for a quantity that is assumed to have a fixed value in the context of a specific application; caused by incomplete model structure information, epistemic uncertainties are reducible through further research to gain more insight in the model, or through collecting more data or more relevant data. The epistemic uncertain input variables are often modeled as sets of intervals, where each variable may be defined by one or more intervals. Notice that this does not imply that any value within that interval is equally likely — as it would be the case with a uniform distribution —; rather, the interpretation is that any value within that interval is a possible realization of that variable. Most frequently, real (or

realistic) applications have to deal with both types of uncertainty at the same time. A common situation is where the form of the probability distribution for an uncertain variable is supposed to be known (e.g., it is normally or lognormally distributed): if the parameters governing the distribution are supposed to be known as well, then we have to deal with aleatory variables according to some methods properly chosen. On the other hand, if the parameters characterizing the distribution remain unknown, then two steps have to be performed: first, the epistemic variables are specified, which means — in our example — that we specify some intervals on parameter values such as means or standard deviations of uncertain variables. A particular value is selected within the specified intervals, according to a given distribution, usually called a *prior distribution*. Then, this value represents a particular realization of the epistemic variable and allows to completely define the distribution on the corresponding aleatory variable.

In the literature, several methods have been proposed for achieving robust optimization in simulation; see Beyer and Sendhoff (2007). The thesis proposes a novel robust optimization methodology that is applicable to both deterministic and stochastic simulation models, and combines the Taguchian view of the uncertain world with a metamodeling technique, namely Kriging.
Genichi Taguchi is an engineer and a statistician; Taguchi (1987) introduced an innovative approach for reducing variations in the product/process design. The author identifies two different types of factors that can be varied in designing a product or process: *controllable* or *decision* factors and *noise* or *environmental* factors; noise factors cannot be changed or varied while the process operates or the product is used. His basic idea in dealing with Robust Parameter Design is to involve noise factors in the experimental design and seek for the most insensitive, or robust, system configuration in the controllable factors w.r.t. the noise factors variation.
Kriging takes the name from a South African mining engineer, Daniel G. Krige, who developed this method for geostatistical applications (Krige, 1951). Kriging provides an interpolating approximation model, which can be seen as a combination of a global model plus a local deviation, usually modelled as $y(x) = f(x) + Z(x)$, where $f(x)$ is a regression model (whose structure is selected by the user), and $Z(x)$ is a Gaussian random function with zero mean and non-zero variance. The construction of the Kriging model is based on an estimation of the unknown parameters by maximizing a likelihood function using numerical optimization techniques to determine the vector of the correlation parameters used to fit the model. See also Kleijnen (2008).

The method proposed considers at least two metamodels, namely one for the mean of the main performance function indicator and one for its variance, caused by the noise factors. During the iterative robust optimization process, the updating and validation of these metamodels needs to be carefully considered. In the simplest case, the developed framework solves a constrained stochastic optimization problem, namely optimizing the expected value of the objective function, such that the variance does not exceed a user-defined threshold. By moving this threshold within a given interval, we are able to estimate a set of Pareto optimal *robust* solutions for the specified problem. Further analyses based on the bootstrap method are also performed

to account for the variability of the metamodelling estimations, thus evaluating the influence this may have on the robust solutions computed.

The resulting methodology has been applied in the context of Supply Chain Management, starting from some building blocks such as the Economic Order Quantity (EOQ) and the (s,S) inventory models.

The dissertation consists of two parts, respectively dealing with the proposed methodology and the computational experiments. Part I first presents a literature review (Chapter 1) on robust optimization methods, paying more attention to those techniques that are applicable to simulation-based applications, eventually using surrogate functions to approximate the simulation model, always treated as a black-box. Then, Chapter 2 introduces two examples from Inventory Management we will refer to throughout the thesis: the Economic Order Quantity (EOQ) model and the (s, S) inventory model are described, starting from the classic formulations and then moving to the robust optimization problems, which will be discussed in details in Part II; some possible extensions of both the two models are outlined, to be used for further research and applications. Chapter 3 discusses a robust approach for simulation-optimization, combining Taguchi's method and Response Surface Methodology: the classic method is briefly recalled, then the robust formulation proposed by Myers and Montgomery (2002) is described; finally a contribution for extending the method is proposed, through removing some restrictive assumptions, thus resulting in a more general approach (Dellino et al., 2008a). Chapter 4 proposes a novel methodology to deal with robust simulation-optimization problems, through integrating Taguchi's method with Kriging metamodels (Dellino et al., 2008b), which often allows a more precise estimation of the system response. Part II extensively discuss the experiments performed on the EOQ and (s, S) inventory model and the results obtained by applying the approaches presented in Chapters 3-4. Chapter 7 is devoted to a final discussion over the methods proposed and the results achieved in the applications considered, comparing the performances provided by the two adopted metamodelling techniques — namely, regression and Kriging — and the alternative approaches examined in previous chapters.

Part I

Methodology

Chapter 1

Robust Optimization in Simulation-based Applications

The simulation-optimization process aims to identify the setting of input parameters leading to optimal system performance, evaluated through a simulation model of the system itself. The factors involved in the simulation model are often noisy and cannot be controlled or varied during the design process, due to measurement errors or other implementation issues; moreover the inherent uncertainty residing in the model — mainly caused by fluctuations in the environmental parameters — is in many cases ignored. Therefore, the presumed optimal solution may turn out to be sub-optimal or even infeasible. Robust optimization tackles problems affected by uncertainty, providing solutions that are almost insensitive to perturbations in the model parameters.

Several alternative methods have been proposed for achieving robustness in simulation-based optimization problems, adopting different experimental designs and/or metamodeling techniques. The present chapter aims at reviewing the current state of the art on robust optimization approaches, especially devoted to simulated systems. The remainder of the chapter is structured as follows: Section 1.1 introduces the "classical" robust optimization approach, originally proposed by Ben-Tal and Nemirovski in 1997; Section 1.2 discusses an alternative approach to the one proposed by Ben-Tal and Nemirovski, which was introduced by Taguchi in 1970s. Section 1.3 presents a method to tackle robustness based on Response Surface Methodology, with some applications to process and design optimization; Section 1.4 discusses robust optimization through the use of Kriging metamodels. Finally, Section 1.5 introduces the robustness issue in the context of Supply Chain Management.

1.1 Robust Counterpart formulation through Uncertainty Sets

The robust optimization methodology developed by Ben-Tal and Nemirovski (2008) investigates different choices of uncertainty sets to model data uncertainty, in order to characterize the structure of the resulting robust counterparts. In particular, their research focuses on robust formulations for Linear Programming (LP), Mixed Integer Programming (MIP), Second Order Cone Programming (SOCP) and Semidefinite Programming (SDP) problems. For these family of problems a fundamental issue is related to the feasibility of the solutions with respect to the classical optima; in particular, the challange is to guarantee that the constraints will be satisfied for any possible value of the parameters in a given uncertainty set. The computational complexity of the deterministic problem and its robust counterpart is also investigated, aiming at insuring that the problem remains tractable.

Although the approach suggested by Ben-Tal has a strong theoretical background, there are several practical problems to which it cannot be applied, due to many reasons (see Beyer and Sendhoff, 2007): the main disadvantage is related to the necessity to model a real-world problem through a linear model with (at most) conic or quadratic constraints; moreover, in order to satisfy all the assumptions under which the method is applicable, the approximate model might become very complex and difficult to manage. Finally, if the objective function is not defined through a mathematical expression but can only be evaluated through simulations, the methodology presented cannot be applied.

Zhang (2004) deals with some of the aforementioned cases, proposing a mathematical formulation to extend Ben-Tal's approach to parameterized nonlinear programming, with both equality and inequality constraints; the inequality constraints are supposed to be strictly satisfiable and are referred to as *safety constraints*. Zhang points out that the approach he proposes is especially suitable for applications where the satisfaction of safety constraints is of crucial importance. However, the formulation of the robust problem assumes that a reasonable estimate for the uncertain parameters is available and the magnitude of the variations in the uncertain parameters is relatively small. He proved that his formulation reduces to the Ben-Tal's when the objective function and the inequality constraints are linear and there is no uncertainty in the equality constraints. Anyway, further research is needed to develop algorithms able to effectively solve the proposed formulation.

Mainly based on Ben-Tal's approach, Bertsimas proposes a formulation for stochastic and dynamic optimization problems using uncertainty sets, in contrast to the stochastic programming approach, which assumes full knowledge of the underlying probability distributions. Bertsimas and Sim (2004) propose a robust optimization methodology — based on linear and mixed-integer programming — to find an optimal supply chain control strategy, assuming stochastic demand. Their approach incorporates demand randomness in a deterministic manner, without making any specific assumption on the demand distribution. First, a robust formulation is given for the

simple uncapacitated single-station case; then, capacity constraints are introduced, both on the orders and on the inventory level; finally, the network case is considered. The numerical experiments showed that, if only the mean and the variance of the demand distribution are known, the robust policy often outperforms the nominal policy, as well as policies computed assuming full but erroneous knowledge of the demand distribution. The authors also prove that the nominal problem and its robust counterpart belong to the same complexity class and the robust formulation does not suffer from the curse of dimensionality. The method can guarantee the robust solution to be feasible if less than a prespecified number of coefficients change; moreover, if the coefficient of variation affects a bigger number of factors, they provide a probabilistic guarantee that the solution will be feasible with high probability. The method has been applied by Bertsimas and Thiele (2004, 2006).

In a recent paper, Bertsimas et al. (2007) propose an approach to solve Robust Optimization problems in which the objective function is not explicitly available, being derived from simulation models. They implement an iterative local search method, moving along descent directions of the worst-case cost function. The first step of the proposed algorithm consist of exploring a (properly defined) neighborhood of the current point; then, a descent direction can be found by solving a Second Order Cone Programming (SOCP) problem. The robust local search is designed to terminate at a robust local minimum, i.e. a point where no improving directions are available for the algorithm.

1.2 Robust Product Optimization: Taguchi's Approach

In the 1980s, Genichi Taguchi, a textile engineer, introduced new ideas on quality improvement, revealing an innovative parameter design approach for reducing variation in products and processes (see Taguchi, 1987). His methodology has been successfully applied in many important industries in America, such as Ford Motor Company and Xerox.

Taguchi identifies three stages in the design process:

- *System Design* is a general approach to design a process that includes defining its objectives and goals.
- *Parameter Design* involves defining responses of interest and optimizing them w.r.t. their mean and variation.
- *Tolerance Design* corresponds to fine-tuning the variables that have been optimized in the previous stage by controlling the factors that affect them.

Notice that the last two stages may appear quite similar to each other and thus it may be difficult to keep them distinct Beyer and Sendhoff (2007). In fact, from a mathematical point of view, parameter and tolerance design differ only in the *granularity*

by which design parameters are treated. On the other hand, from a practical point of view, it is important to distinguish between the two phases, because they can occur under very different restraints, e.g. design time versus operation time. As Figure 1.1 illustrates, Taguchi — focusing on Parameter Design — distinguishes between two different types of factors that can be varied in designing a product or process:

- *control* or *decision* factors (denoted by $d_j, j = 1, \ldots, n_d$) are under the control of the users; e.g., in inventory management, the order quantity may be controllable.
- *noise* or *environmental* factors (denoted by $e_k, k = 1, \ldots, n_e$) cannot be changed or varied while the process operates or the product is used; e.g. the demand rate in inventory problems.

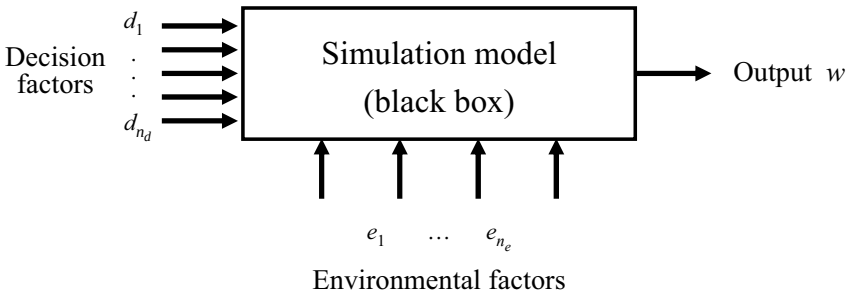


Figure 1.1: Taguchi's view

Notice that, in practice, the controllability of a factor depends on the specific situation; e.g., the users may change the demand rate through an advertising campaign.

Other authors distinguish between environmental uncertainty (e.g., demand uncertainty) and system uncertainty (e.g., yield uncertainty); see Mula et al. (2006) and also Beyer and Sendhoff (2007). Implementation errors may also be a source of uncertainty. These errors occur whenever recommended (optimal) values of control factors have to be realized in practice; see Stinstra and den Hertog (2008). Continuous values are hard to realize in practice, because only limited accuracy is then possible; e.g., the optimal solution in the Economic Order Quantity model (EOQ) turns out to be the square root of some expression, but in practice only a discrete number of units can be ordered. Besides implementation errors, there are validation errors of the simulation model — compared with the real system — and the metamodel — compared with the simulation model —; see Kleijnen and Sargent (2000).

Taguchi's basic idea in dealing with Robust Parameter Design is to take account of the noise factors in the experimental design and seek for the most insensitive, or robust, system configuration in the controllable factors w.r.t. the noise factors

variation.
As far as the experimental strategy is concerned, Taguchi adopted crossed arrays, resulting from the product of two Design of Experiments, one in which control factors $\mathbf{d}$ are varied (obtaining the "inner array", as Taguchi calls it) and one in which noise factors $\mathbf{e}$ are varied (thus obtaining the "outer array"): combining them together is equivalent to consider variations in the uncontrollable factors at different locations in the space of the controllable factors.

Taguchi further distinguishes between factors that have a *location* effect, changing the mean of the response or objective function, and factors that have a *dispersion* effect, since they affect the variance of the process. Therefore, in the optimization process, Taguchi takes into account the first two moments of the distribution of the objective function, and combines them using the signal-to-noise ratio (SNR). Taguchi suggests to consider three types of problems, which correspond to minimization, maximization, and the case the response has a target value:

1. "Smaller the better". Here it is suggested to select the solution as the factor combination in the inner array that maximizes

$$SNR_S = -10 \log \sum_{i=1}^{n_0} \frac{w_i^2}{n_0} \tag{1.2.1}$$

   where $w_i = w(\mathbf{d}, e_i)$ and n_0 is the number of runs in the outer array.

2. "Larger the better". Here the solution is given by the inner array point maximizing

$$SNR_L = -10 \log \frac{1}{n_0} \sum_{i=1}^{n_0} \frac{1}{w_i^2} \tag{1.2.2}$$

3. "Target is best". Here Taguchi proposes a two-step approach, suggesting two cases:

   (a) μ_w (mean of w) is not related to σ_w (standard deviation of w). In this case the steps are the following:

      i - Select some control factors that maximize

$$SNR_{T1} = -10 \log s^2 \tag{1.2.3}$$

      where s^2 is the sample variance of the outer array observations.

      ii - Select some other inner array factor (not varied before) to make

$$\overline{w} \approx T \tag{1.2.4}$$

      where $\overline{w}$ is the average of the outer array observations and T is the target of the quality characteristic.

   (b) If σ_w is proportional to μ_w, a case likely to occur in practice, then

i - Select some control factors to maximize

$$SNR_{T2} = -10 \log \frac{\overline{w}^2}{s^2} \tag{1.2.5}$$

ii - Select some other control factor, not varied before, to make

$$\overline{w} \approx T.$$

Since the standard deviation is proportional to the mean, the idea is that the controllable factors will change the mean but will not change the ratio $\overline{w}^2/s^2$ much.

Some aspect of the Taguchian approach have been strongly criticized; see, among others, Myers et al. (1992), Myers and Montgomery (2002) and del Castillo (2007). The mostly debated issues were the following:

- A data set with no outer array variability and one with considerable outer array variability may result in the same SNR; therefore, SNR would result inefficient for Robust Parameter Design.
- No attention is paid to the computational costs required by the experimental design: in fact, using a crossed array design often requires a large number of runs, which can be prohibitive in some industrial processes.
- The method lacks of flexibility in modeling the design variables, not taking into account the interactions either among control factors or between control and noise factors. Standard ANOVA techniques can be used to identify the control factors that impact SNR (see Myers and Montgomery, 2002; Robinson et al., 2004).
- Factors may have both location and dispersion effects, so the proposed two-step approach may be infeasible to do in practice. Moreover, the adoption of the SNR as performance characteristic appears to be too restrictive (Park et al., 2006) and risks to confound the mean and variance contributions; keeping them separately, instead, can provide further insight into the process behaviour.

1.2.1 Definition of a Loss Function

Although based on Taguchi's view of modeling uncertainty in a design process, some authors (Trosset, 1997) have suggested to directly model the response as a function of both control and noise factors, instead of using SNRs. Suppose to measure q performance indicators, $w_1, \ldots, w_q$; let $w_i(\mathbf{d}, \mathbf{e})$ denote the value of the i-th performance indicator when control and noise factors assume values $(\mathbf{d}, \mathbf{e})$ and let $l[w_1(\mathbf{d}, \mathbf{e}), \ldots, w_q(\mathbf{d}, \mathbf{e})]$ denote the corresponding loss. A robust design approach will seek a combination of control factors that minimizes the expected loss, computed

w.r.t. the random vector $\mathbf{e}$. If the distribution of $\mathbf{e}$ does not depend on $\mathbf{d}$, then the objective function

$$L(\mathbf{d}) = \int l[w_1(\mathbf{d}, \mathbf{e}), \ldots, w_q(\mathbf{d}, \mathbf{e})]\, p(\mathbf{e})\, d\mathbf{e}, \tag{1.2.6}$$

is obtained, where $p(\mathbf{e})$ denotes the probability density function of $\mathbf{e}$.
Then a question arises: how have statisticians sought to minimize Eq. 1.2.6? A numerical optimizer would answer to this question in the following manner:

1. A design is chosen, that specifies the (d_j, e_j) at which the y_i have to be evaluated; this approach results in a single "combined" array, instead of using inner and outer arrays — as Taguchi suggested to do.

2. The $w_i(d_j, e_j)$ are used to construct cheap-to-compute surrogate models $\widehat{y}_i$.

3. Optimization is carried out using the surrogate objective function

$$\hat{L}(\mathbf{d}) = \int l[\hat{y}_1(\mathbf{d}, \mathbf{e}), \ldots, \hat{y}_q(\mathbf{d}, \mathbf{e})]\, w(\mathbf{e})\, d\mathbf{e}. \tag{1.2.7}$$

A similar approach is suggested by Sanchez (2000), who proposes a robust methodology, starting from Taguchi's approach and combining it with metamodeling techniques. Focusing on discrete-event simulation models, she identifies some performance characteristic, denoted by $w(\mathbf{d})$, $\mathbf{d}$ being the vector of decision factors, and an associated target value T. The goal would be to select the control factors to keep the objective function on target, with zero variance. However, this would be an ideal situation — hard to realize in practice. Therefore, aiming at finding a trade-off between performance mean and variability, Sanchez proposes to use a quadratic loss function, defined as follows: assuming that no loss occurs when $w(\mathbf{d})$ achieves the target T, the quadratic loss function can be written as

$$l(w(\mathbf{d})) = c[w(\mathbf{d}) - T]^2, \tag{1.2.8}$$

where c is a scaling factor, introduced to take account of possible units conversions. It follows from Eq. 1.2.8 that the expected loss associated with configuration $\mathbf{d}$ is

$$E[l(w(\mathbf{d}))] = c[\sigma^2 + (\mu - T)^2]\,, \tag{1.2.9}$$

where μ and σ^2 denote the true mean and variance of the output function y, respectively.
As far as the robust design is concerned, Sanchez tries to characterize the system behaviour as a function of the control factors only. First, an appropriate experimental design is planned, for both control and noise factors. Then, for every combination of control factor configuration i and noise factor configuration j, the sample average $\overline{w}_{ij}$

and sample variance s_{ij}^2 are computed — after suitable truncation to remove initialization bias. Finally, summary measures across the noise space for each control factor configuration i are computed:

$$\overline{w}_{i\cdot} = \frac{1}{n_e} \sum_{j=1}^{n_e} \overline{w}_{ij} \tag{1.2.10}$$

$$\overline{V}_{i\cdot} = \frac{1}{n_e - 1} \sum_{j=1}^{n_e} (\overline{w}_{ij} - \overline{w}_{i\cdot})^2 + \frac{1}{n_e} \sum_{j=1}^{n_e} s_{ij}^2 \tag{1.2.11}$$

where n_e is the number of points in the noise factor plan.
Two initial metamodels are then built, using regression polynomials: one for the performance mean, and one for the performance variability; for discrete-event simulation experiments, Sanchez recommends a design which allows for fitting at least a quadratic effect. Robust configurations are identified by combining information resulting from the mean and variance metamodels, using Eq. 1.2.9, where the true mean and variance are replaced by the estimate given in Eqs. 1.2.10-3.2.24. If the configurations suggested by the robust design were not among those initially tested, further experimentation could be needed: in this case, however, computational time could be saved, by screening among the decision factors involved in the experiment.

Al-Aomar (2002) presents an iterative scheme to solve simulation-based optimization problems. His work considers a discrete-event simulation model; the (controllable) design parameters characterizing the system are $d_1, \ldots, d_n$, and its performances are evaluated through some metrics $w_1, \ldots, w_q$. Then, an overall utility function U is defined combining multiple performance measures at each point into a single function. The general formulation of the system design problem can be defined as follows:

$$\begin{aligned} &\max\ U(w_1, \ldots, w_q) \\ &s.t.\ w_i = f_i(d_1, \ldots, d_n), \quad 1 \leq i \leq q \\ &d_j \in S, \quad 1 \leq j \leq n \end{aligned} \tag{1.2.12}$$

S being the feasible space for the control variable $\mathbf{d}$. The methodology proposed by Al-Aomar consists of four modules: i) in the Simulation Modelling (SM) module, a discrete-event simulation model is utilized to evaluate the set of performance metrics w_i associated with each solution alternative $\mathbf{d}$, in terms of means and variances; ii) the Robustness Module (RM) transforms the mean and variance of each performance measure into a Signal-to-Noise Ratio — thus adopting a Taguchian approach; iii) the Entropy Method (EM) module is used to build the utility function U by linearly combining the performance criteria, through a proper choice of the weights, dynamically updated at each iteration; iv) the Genetic Algorithm (GA) module is utilized as a global optimizer, working on a set of possible solutions that are selected basing on the overall utility function value at each point. A convergence test is conducted at the end of each step to control whether any stopping criterion is met (maximum number

of generations reached or convergence rate achieved). For a detailed discussion, see also El-Haik and Al-Aomar (2006).

1.3 Dual Response Surface Approach

Moving from Taguchi's approach to Robust Optimization and on the basis of the criticism on using SNRs, some authors like Myers and Montgomery (2002) suggest to build separate models for the mean and variance of the system performance, adopting the so-called Dual Response Surface approach. This methodology has some advantages:

- It provides an estimate of mean and standard deviation at any location of interest in the control design variables.
- Some insight can be gained regarding the roles of these variables in controlling process mean and variance.
- It could be easily integrated into process optimization based on a squared error loss criterion, $\widehat{E}_{\mathbf{e}}(w-T)^2 = [\widehat{E}_{\mathbf{e}}(w)-T]^2 + \widehat{\sigma}^2_{\mathbf{e}}(w)w$ or the maximization of an estimated quantile $\hat{E}_{\mathbf{e}}(w) - 2\hat{\sigma}_{\mathbf{e}}(w)$ in the Taguchian "larger the better" case, or the minimization of $\widehat{E}_{\mathbf{e}}(w) + 2\widehat{\sigma}_{\mathbf{e}}(w)$ in the Taguchian "smaller the better" case.
- It allows the use of constrained optimization; that is, choosing a target value of $\widehat{\mu}_{\mathbf{e}}[w(\mathbf{d},\mathbf{e})]$ or, better to say, a threshold T, below which one cannot accept. Therefore, the following problem has to be solved:

$$\min_{\mathbf{d}} \widehat{\sigma}^2_{\mathbf{e}}[w(\mathbf{d},\mathbf{e})] \quad s.t. \quad \widehat{\mu}_{\mathbf{e}}[w(\mathbf{d},\mathbf{e})] \leq T \tag{1.3.1}$$

  The choice of several values of T may be made to provide several alternatives for the user.

1.3.1 Robust Process Optimization

The dual response surface approach has been successfully applied to robust process optimization, as discussed in del Castillo (2007). Quite often, in fact, the purpose is to reach a desired performance for the process that manufactures some products — e.g. by minimizing the cost of operation in a production process, or the variability of a quality characteristic, or by maximizing the throughput of the manufacturing process. Evidently, multiple — and sometimes conflicting — responses are usually considered in practical problems. However, due to noisy data and/or to uncertainty affecting some parameters of the model, achieving robust performances is of interest.

Besides what discussed so far about possible sources of uncertainty, Miró-Quesada and del Castillo (2004) point out that the classical Dual Response Surface approach takes into account only the uncertainty due to the noise factors; they identify an additional component as due to the uncertainty in the parameter estimates. Therefore,

they propose an extension of the Dual Response Surface approach, by introducing the additional variance of the parameters estimates into an objective function that combines it with the noise factor variance. Optimizing such a function will achieve a process that is robust with respect to both noise factor variation and to uncertainty in the parameter estimates.

One such function is the variance of the predicted response, where the variance is now taken w.r.t. the parameter estimates of the model and w.r.t. the noise factors, thus resulting in $Var_{\mathbf{e},\widehat{\boldsymbol{\beta}}}(\widehat{y}(\mathbf{d},\mathbf{e}))$.

1.3.2 Robust Design Optimization

Robustness is also a central issue in design optimization. Many engineering applications have to deal with the uncertainty which affects the components of the system under design; ignoring the source of uncertainty and assuming some parameters to be *exactly* known and constant might cause the designed system not to be adequate whenever the environmental setting changes.

Bates et al. (2006) compare different methods to perform RO, applying them to solve (robust) design optimization of a piston: the objective is to achieve a given mean cycle time while minimizing the standard deviation of the response. The authors discuss the Taguchian approach, based on crossed-array experimental design and maximization of the SNR; the Response Model analysis, where the design involves both decision and environmental factors, taking account not only of main effects but also of factor interactions, which can influence the variance of the system response; and the Dual Response Surface approach. The framework proposed, which is called Stochastic Emulator Strategy, consists of the following building blocks: (i) DoE, using an array that includes both design and noise factors; preferable designs are space-filling designs (such as LHS) or lattice designs rather than orthogonal arrays or fractional factorials, to achieve more uniform coverage of the input space. (ii) Metamodel (or emulator, as they called it) fitting, to represent the relationship between all factors — disregarding whether they are decision or environmental factors — and the chosen response. (iii) Metamodel prediction, to estimate the piston cycle time for a given set of factor values and evaluate the effect of noise on the output by studying how it behaves when subject to small changes in factor values. (iv) Optimization process, minimizing the output variance, with a target value for the mean cycle time.

1.4 Robust Optimization using Kriging Metamodels

Lee and Park (2006) present a methodology — based on Kriging metamodels — to tackle robust optimization in (deterministic) simulation-based systems. Simulated annealing is adopted to solve the optimization problem. The approach is basically the one proposed by Taguchi, employing mean and variance as statistics to study the insensitivity of the response to possible variations in the noise factors.

The use of Kriging as approximation technique is justified by the fact that Kriging provides reliable approximation models of highly nonlinear functions, and this feature is even more useful in RO than it is in classical optimization, since in general the nonlinearity of the response variance is higher than that of the function itself. The choice is also supported by Jin et al. (2003), who compare some metamodelling techniques and — based on the results of some tests performed on both mathematical functions and a more complex case study — they conclude that Kriging models provide better accuracy than the other selected alternatives. However, Allen et al. (2003) notice that regression modeling should not be quickly discarded for cases in which the number of runs is comparable to the number of terms in a quadratic polynomial model. Besides that, Kriging-based approaches offer potentially important accuracy advantages as the number of runs increases from those minimally needed to estimate a quadratic polynomial model.

1.4.1 Problem formulation and algorithm description

We denote the response we are interested in by w; it depends on both control factors ($\mathbf{d}$) and noise factors ($\mathbf{e}$).

The robust optimization problem aims at determining a design point $\mathbf{d}$ (we are not able to control the noise factors $\mathbf{e}$) providing a target response value $\overline{\mu}_w$, with the smallest variation, σ_w^2. The corresponding formulation is given by:

$$\min \quad \sigma_w^2 \tag{1.4.1}$$

$$s.t. \quad \mu_w \leq \overline{\mu}_w \tag{1.4.2}$$

Since the analytical computation of both mean and variance of a given response w is not always viable, being time expensive or too difficult to be solved, one might think to approximate the two statistics by means of the first-order Taylor expansion, which provides the following expressions:

$$\mu_w \approx w(\mathbf{d}, \mathbf{e})_{\bar{\mathbf{d}}, \bar{\mathbf{e}}}\,; \tag{1.4.3}$$

$$\sigma_w \approx \sum_{i=1}^{n_d} \left(\frac{\partial w}{\partial d_i} \right)_{\bar{\mathbf{d}}}^2 \sigma_{d_i}^2 + \sum_{j=1}^{n_e} \left(\frac{\partial w}{\partial e_j} \right)_{\bar{\mathbf{e}}}^2 \sigma_{e_j}^2\,, \tag{1.4.4}$$

where $\bar{\mathbf{d}}$ and $\bar{\mathbf{e}}$ denote the mean vectors of the control and noise factors, respectively, and $\sigma_{d_i}^2$ and $\sigma_{e_j}^2$ represent the variance of the i-th control variable and the j-th noise variable, respectively.

It should be pointed out, however, that Eqs. 1.4.3-1.4.4 are valid approximations only for monotonic functions, which is usually a property difficult to determine, when working with black-box simulation models.

Fig. 1.2 depicts the main steps of the method proposed by Lee and Park (2006).

Global Robust Optimization (Lee and Park, 2006)

Input: initial design ($\mathbf{D_1}$, $s_1 \times (n_d + n_e)$ in the control-by-noise space $(\mathbf{d}, \mathbf{e})$;
probability distribution of control and noise factors;
design for Monte-Carlo simulations ($\mathbf{D_2}$, $s_2 \times (n_d + n_e)$), $s_2 > s_1$;
begin
for each design point $(\mathbf{d}_k, \mathbf{e}_k)$ in $\mathbf{D_1}$, $k = 1, \ldots, s_1$
Run the simulation model
Compute $w_k = w(\mathbf{d}_k, \mathbf{e}_k)$
Fit a Kriging metamodel of the response, $y(\mathbf{d}, \mathbf{e})$
for each design point in $\mathbf{D_2}$
Predict the response using the metamodel, $\hat{y}(\mathbf{d}, \mathbf{e})$
Compute the sample variance w.r.t. the noise factors $\mathbf{e}$, $\sigma^2_{\hat{y}}(\mathbf{d})$
Solve the optimization problem, according to Eq. 1.4.1,
using the Kriging metamodels for mean and variance
end

Figure 1.2: Lee and Park (2006)'s Global Robust Optimization Scheme

1.4.2 Important issues

Some issues emerge from Lee and Park (2006), which are shortly pointed out in the following:

- The authors fit one *single* metamodel over the control-by-noise factors space. They suggest it to be highly accurate, since it would be used to derive the approximation model for the variance.

- To derive a model for the mean of the response, they use the approximation provided by Eq. 1.4.3, applying it to the metamodel computed.

- To derive a model of the variance, they use Monte-Carlo simulations, performed not on the simulation model but on the (inexpensive) metamodel obtained in the beginning.

- The authors point out that a post-processing may be necessary, due to the nonlinearity both of the response and (even more) of its variance, and to approximation errors coming from fitting the metamodel of the variance based on the metamodel of the response function. It would aim to further refine the robust optimum, but experimental results usually show quite small improvements. The post-processing consists in solving the following optimization problem, restricting the search area to the neighborhood of the optimal solution found so

far:

$$\min \quad \widehat{\sigma}_w^2 = \sum_{i=1}^{n_d} \left(\frac{\partial \hat{y}}{\partial d_i} \right)^2 \sigma_{d_i}^2 + \sum_{j=1}^{n_e} \left(\frac{\partial \hat{y}}{\partial e_j} \right)^2 \sigma_{e_j}^2 \,, \tag{1.4.5}$$

$$s.t. \quad \hat{y}(\mathbf{d}, \mathbf{e})_{\bar{\mathbf{d}}, \bar{\mathbf{e}}} \leq \bar{\mu}_w \tag{1.4.6}$$

- As a further research topic, the authors suggest to adopt two *distinct* meta-models, approximating the true response and the true variance. The authors suggest to adopt this approach for strongly nonlinear models, especially as far as the variance model is concerned.

1.5 Optimization under uncertainty in Supply Chain Management

Supply Chain Management (SCM) is the process of planning, implementing and controlling the operations of a supply chain as efficiently as possible. The importance of supply chain management is increasing over time, because companies have to face the necessity to improve customer service, which is not possible by considering just separate organizations. In fact, a crucial role for improving the performance of the entire system — which usually has a complex structure — is played by the coordination of interactions among all the entities involved. This need has been driven by increasing customer expectations, growing global competition, and technological developments, which have jointly contributed to greater uncertainty and volatility of the management processes.

In general, supply chains operate in uncertain environments. Therefore, operations need to be planned and executed to account for risk and uncertainty, that can be caused by either internal or external factors. One of the primary challenges is represented by customer demand uncertainty, which shows up in multiple ways, such as increasing customer expectations for price, quality and delivery performance, demand for customized products, shortened product life cycle, and unstable demand behavior. These factors are supplemented by the traditional uncertainty concerning demand volume.

As a result of what discussed so far, robustness turns out to be one of the major issues that must be taken into account in SCM. Designing a robust supply chain would imply deriving values for the factors that managers can control when making a strategic decision about the supply chain, accounting for the randomness of the environmental factors. The supply chain literature distinguishes between robustness and flexibility. A flexible supply chain can react to a changing environment by adapting its operations; see Beamon (2000) and Zhang et al. (2003). A robust supply chain keeps its design fixed, but can still accommodate many changes in its environment. So the two concepts focus on operational and strategic decisions respectively. Also see Van Landeghem and Vanmaele (2002).

Inventory planning plays a crucial role in SCM (see Shapiro, 2007): this asks for the design of planning systems using inventory models to select reorder points, replenishment quantities, and safety stocks based on demand forecasts. Inventory planning is strictly related to the decision-making process which affects the whole supply chain; ignoring other supply chain decisions and costs can have negative consequences on the efficiency and effectiveness of the inventory management. On the other hand, supply chain plans ignoring the inventory consequences will also be inefficient and ineffective.

The objective of inventory management has been to keep enough inventory to meet customer demand, being — at the same time — cost effective. However, inventory has not always been perceived as an area to control costs. Traditionally, companies have held "generous" inventory levels to meet long-term customer demand because there were relatively few competitors and products in a generally sheltered market environment. In the current global business environment, with many more competitors and highly diverse markets where new products and new product features are rapidly and continually introduced, the cost of inventory has increased, partially due to faster product obsolescence. At the same time, companies are continuously seeking to lower costs so they can provide a better product at a lower price.

Indeed, in this context, inventory is a natural candidate for cost reduction (Russell and Taylor, 2008). High inventory costs have motivated companies to focus on both efficient supply chain management and quality management. They believe that inventory can be significantly reduced by reducing uncertainty at various points along the supply chain. In many cases, uncertainty is introduced by poor quality on the company or its suppliers or both. These factors can be expressed through variations in delivery times, uncertain production schedules caused by late deliveries, poor forecasts of customer demand, large fluctuations in customer demand and large numbers of defective products requiring levels of production or service higher than necessary.

An efficient supply chain management allows products or services to be moved from one stage of the supply chain to another according to a reliable communication system between customers and suppliers. Items are replaced as they are diminished without maintaining larger buffer stocks of inventory at each stage to compensate for late deliveries, inefficient service, poor quality, or uncertain demand. An efficient and well coordinated supply chain reduces these sources of uncertainty so that this type of system will work adequately; see Shapiro (2007) and Russell and Taylor (2008).

Some companies maintain buffer inventories between production stages to offset irregularities and problems and keep the supply chain flowing smoothly. On the other hand, quality-oriented companies consider large buffer inventories to be a costly crutch that hides problems and inefficiency primarily caused by poor quality. Followers of quality management believe that inventory should be minimized. However, this may primarily work for a production or manufacturing process. For the retailer who sells finished goods directly to the customer or the supplier who sells parts or materials to the manufacturer, inventory still is a necessity. Hence the traditional inventory decisions of how much to order and when to order continue to be important in the supply chains. In addition, the traditional approaches — mainly based on Economic Order Quantity and (s,S) models (see, e.g., Zipkin (2000)) — to inventory

management are still widely used by most companies (Neale et al., 2004).

It is matter of fact that a company employs an inventory strategy for many reasons both in final and intermediate stages of its supply chain. Since demand is usually not known with certainty, it is not possible to produce exactly the amount required. An additional amount of inventory is kept on hand to meet variations in product demand. This uncertainty about demand back upstream in the supply chain and causes stages to increase their level of inventory to compensate. Other stocks of inventory are devoted to cope with seasonal or cyclical demand maintaining a relatively smooth production flow during the year.

At the other end of the logistic chain, a company might keep stocks of parts and raw materials to meet variations in supplier deliveries. Components and materials are kept on hand so that the production process will not be delayed as a result of missing or late deliveries or shortages from a supplier. These motivations often induce companies to use inventories to decouple the production processes of different stages of the supply chain. Moreover, a company will form stock of inventory to take advantage of price discounts or future price variations or even to limit the effect of cost ordering.

The ability to effectively meet internal organizational demand or external customer demand in a timely and efficient manner leads to the concept of level of customer service. A primary objective of today's supply chain management is to provide a level of customer service as high as possible. Customers for finished goods (or internal customers along the supply chain) usually perceive quality service as availability of goods when they want them. To provide this level of quality in the customer service, the tendency is to maintain large stocks of all type of items. However, there is a cost associated with carrying items to inventory, which creates a tradeoff between the quality level of customer service and the cost of that service; see Russell and Taylor (2008).

As the level of inventory increases to provide better customer service, inventory costs increase. The conventional approach to inventory management is to maintain a level of stock to solve the compromise between inventory costs and customer service. The current global view of the production and logistics leads companies to strive to reduce prices also through reduced inventory costs and calls for more sophisticated models and methods able to reduce the uncertainties characterizing the supply chain giving robust inventory solutions; see Shapiro (2007) and Russell and Taylor (2008).

Chapter 2

Two examples from Inventory Management

In this chapter we introduce some examples we will use as benchmarks to test our heuristic method, taken from inventory optimization within the context of Supply Chain Management. Although we have not described the proposed methodology yet, this chapter may help in understanding the procedure more easily, having already in mind the context and some applications.

The chapter is structured as follows: Section 2.1 deals with the Economic Order Quantity (EOQ) model, Section 2.2 discusses the (s, S) inventory model and Section 2.2.1 extends the previous model, by taking into account possible output constraints (e.g. on the service level). Each section includes the description of the classic version of the problem — where no uncertainty is assumed —, then it introduces the robust formulation, highlighting the motivation behind the study, the variables involved (distinguishing them between decision and environmental factors) and the objectives we want to optimize.

2.1 Economic Order Quantity (EOQ) model

2.1.1 Problem description

The well-known Economic Order Quantity (EOQ) model is based on the following assumptions (see Zipkin, 2000):

i. The demand is known and constant, equal to a items per unit time.

ii. Q items are ordered at a time, to replenish the inventory.

iii. Planned shortages are not allowed.

iv. The delivery lead time is 0, i.e. goods arrive into inventory as soon as an order is placed.

v. Continuous review is assumed, i.e. an order is placed as soon as the inventory level falls down to the prescribed reorder point, which is set to zero, due to hypotheses (i) and (iv).

vi. Costs include:

- the setup cost for ordering one batch, K;
- the cost for producing/purchasing each item, c;
- the holding cost per item held in inventory per unit time, h.

The goal is to minimize the average costs per unit time, over an infinite time horizon. Defining a cycle as the period T between two consecutive replenishment of inventory, the cycle length is given by $T = Q/a$. Fig. 2.1 puts into evidence the main variables involved and how the inventory level varies over time.

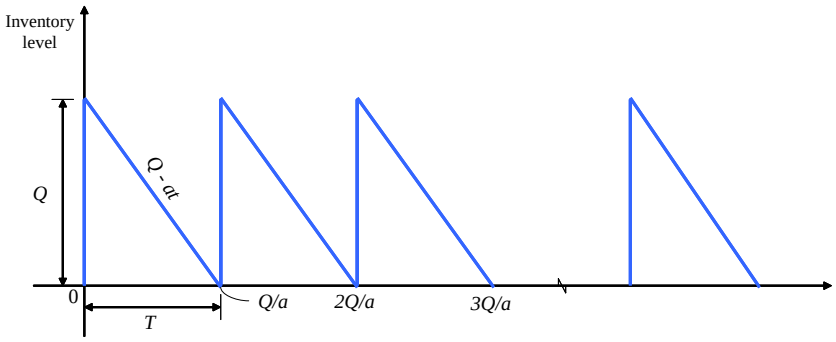


Figure 2.1: Inventory level over time for the basic EOQ model.

Since all cycles are identical — as follows from the definition together with hypotheses (i) and (ii) — the average cost per unit time is simply equal to the total cost incurred in one single cycle divided by the cycle length. Therefore, the experiments we perform focus on a single cycle, whose length is constant.

Let us examine the terms contributing to the total cost per cycle:

- The costs of producing/purchasing one batch in a cycle is given by

$$C_P(Q) = K + c\,Q \tag{2.1.1}$$

- The holding cost per cycle (say, from $t = 0$ to $t = T$) can be calculated as

$$C_H(Q) = h \int_0^T (Q - at)dt = h(QT - \frac{1}{2}aT^2) = \frac{hQ^2}{2a} \tag{2.1.2}$$

  where the integral expresses the inventory available in one cycle.

By adding up the two terms in (2.1.1) and (2.1.2)

$$C_P(Q) + C_H(Q) = K + c\,Q + \frac{hQ^2}{2a} \tag{2.1.3}$$

we obtain the total cost occurring in one cycle; finally, the total cost per unit time is obtained as:

$$C(Q) = \frac{K + c\,Q + \frac{h\,Q^2}{2a}}{Q/a} = \frac{aK}{Q} + ac + \frac{hQ}{2}. \tag{2.1.4}$$

The value of Q (say Q_o) that minimizes the total cost C is found by setting the first derivative of the objective function to zero:

$$\frac{\partial C}{\partial Q} = 0 \Rightarrow -\frac{aK}{Q^2} + \frac{h}{2} = 0 \Rightarrow Q_o = \sqrt{\frac{2aK}{h}}; \tag{2.1.5}$$

of course, the negative solution has been discarded, because the order quantity Q must assume non-negative values. Notice that the second derivative of Q_o is strictly positive, thus guaranteeing that the optimal solution is a minimum, indeed. The corresponding optimal cost is

$$C_o = C(Q_o) = \frac{aK}{Q_o} + ac + \frac{hQ_o}{2} = \sqrt{2aKh} + ac\,; \tag{2.1.6}$$

the optimal cycle time is given by

$$t_o = \frac{Q_o}{a} = \sqrt{\frac{2K}{ah}}\,. \tag{2.1.7}$$

From the EOQ formula in Eq. 2.1.5 some observations can be pointed out, especially concerning the influence of some parameters' variation:

- Being the unit cost of an item c independent on the batch size, it does not appear in the optimal solution for the order quantity, Q_o; therefore, possible fluctuations in the value of this parameter will not influence the position of the optimal point.
- Increasing the setup cost K, both Q_o, C_o and t_o increase.
- When the unit holding cost h increases, both Q_o and t_o decrease, whereas C_o increases.

- As the demand rate a increases, both Q_o and C_o increase but t_o decreases.
- The curve of the total cost is relatively flat around the minimum cost solution, thus resulting in being relatively insensitive to small changes in order quantity w.r.t. the optimal point Q_o.

All these aspects will be further investigated through the sensitivity analysis that will be described later on, in Section 5.1, also supported by a numerical example.

2.1.2 Possible extensions of the basic EOQ model

It is important to point out that the basic EOQ model assumes some simplistic conditions, that are seldom satisfied in practice. So, starting from this model, it is possible to formulate some others, which allow us to relax some assumptions moving towards new models, providing better approximations of real problems.

Holding cost related to the ordering costs

It makes sense to consider the holding cost not as a constant, but as a quantity depending on the ordering cost by means of a coefficient taking into account the capital invested in the product, namely the MARR (Minimum Acceptable Rate of Return); this kind of relationship is expressed through Eq. 2.1.8:

$$h(Q) = r_0 \cdot \bar{Q}\,\frac{p(Q)}{Q} \tag{2.1.8}$$

where r_0 is the MARR and $\bar{Q}$ is the total inventory available in one cycle (see Eq. 2.1.2). So, the holding cost per cycle is given by:

$$h(Q) = r_0\,\frac{Q^2}{2a}\frac{K + c\,Q}{Q} = r_0\frac{Q}{2a}(K + c\,Q)$$

Notice that a more realistic assumption on the holding costs would include also an additive component modelling all the costs for physical handling, warehouse rental, etc. Then, the total cost per unit time will be obtained as:

$$C(Q) = \frac{p(Q) + h(Q)}{Q/a} = r_0\left(\frac{K}{2} + \frac{c\,Q}{2}\right) + \frac{aK}{Q} + ac$$

Following a procedure similar to that adopted in the basic EOQ model, we can easily obtain the expression for the optimal batch size as follows:

$$\frac{\partial C}{\partial Q} = 0 \Rightarrow r_0\frac{c}{2} - \frac{aK}{Q^2} = 0 \Rightarrow Q_o = \sqrt{\frac{2aK}{r_0 c}} \tag{2.1.9}$$

Comparing the EOQ formula just obtained with the former one, obtained for the basic model, it can be noticed that the only difference between the two is in the denominator, where Eq. 2.1.9 replaces the holding cost coefficient h in Eq. 2.1.5 by the product $r_0\,c$.

EOQ model with planned shortages

The inventory shortage or stock-out refers to the demand that cannot be currently satisfied due to the depletion of product stocks. Relaxing the requirement that demand is always met from on hand stock, we now consider a policy where planned shortages are allowed: this implies that, when a shortage occurs, the affected customers will wait for the product to become available again. The backorders are filled immediately when the order quantity arrives to replenish inventory.

We need to introduce some additional notations: the unit shortage cost per time unit when shortages occurs, c_b, and the inventory level just after a batch of Q units is added to the inventory, S. So, in this model there are two decision variables, Q and S; the optimal solution minimizing the total cost per time unit C is found by setting both the partial derivatives of C (w.r.t. Q and S) to zero, which give the following results:

$$Q_o = \sqrt{\frac{2aK}{h}}\sqrt{\frac{c_b + h}{c_b}} \qquad S_o = \sqrt{\frac{2aK}{h}}\sqrt{\frac{c_b}{c_b + h}} \tag{2.1.10}$$

Positive Lead Time

Assuming zero lead time — as the EOQ model does — is clearly unrealistic: in practice, there will be some time $LT > 0$ between the placement of an order and its arrival. Still assuming a constant demand rate a, an order can be placed LT time units before the current inventory level will drop to zero: i.e. we place an order exactly in the lead time it takes for the new order to arrive. Therefore, in this case the reorder point is set to:

$$s = LT \cdot a\,; \tag{2.1.11}$$

therefore s represents the amount of inventory which will be used up during the lead time LT. At the beginning of each cycle, the inventory on hand is equal to Q items. During the cycle, the inventory level decreases at a constant demand rate a — exactly in the same way observed in the classic EOQ model. When the inventory level equals the reorder point s, a new order of Q items is placed. After a lead time LT, that is at the end of the cycle, the new order arrives, just when the inventory runs out of stock. Then, a new cycle begins with the arrival of the new order of Q units.

Finite Production Rate Model

Another extension of the basic EOQ model is the Economic Manufacturing Quantity (EMQ) model: this model assumes that an order of stock does not arrive instantaneously but instead, stock is produced at a finite rate of ψ items per time unit, where $\psi > a$. Here, while production of a batch is underway, stock is accumulating at a rate of $\psi - a$ per time unit. The total time of production in a cycle is Q/ψ, so that the peak level of inventory in a cycle is $(\psi - a)Q/\psi$. The total inventory held in a cycle is thus $(\psi - a)QT/2\psi$.

Proceeding as with the EOQ model, we obtain the optimal batch size Q_o as:

$$Q_o = \sqrt{\frac{2aK\psi}{h(\psi - a)}} \tag{2.1.12}$$

which has almost the same form as the EOQ formula, except that the holding cost coefficient should be defined as

$$h' = h\left(1 - \frac{a}{\psi}\right)$$

As ψ goes to infinity, Eq. 2.1.12 reduces to the EOQ.

EOQ model with Quantity Discount

The models considered so far were based on the assumption that the unit cost of the product is the same regardless of the quantity ordered; as a consequence, the optimal solution turns out to be independent on this parameter. However, this assumption can be removed, considering the unit cost of an item related to the batch size: in fact, it is realistic that suppliers offer lower prices for larger orders, thus providing an incentive to increase the order size. Two kinds of discounts can be considered:

- According to the incremental discount model, we assume that the purchase price changes at some breakpoint χ, such that the (variable) ordering cost is c_0 for any amount up to χ, while — for an order larger that χ — the additional amount over χ is charged by a rate $c_1 < c_0$. Thus, the total ordering cost per cycle is $K + c(Q)$, where

$$c(Q) = \begin{cases} c_0\, Q & 0 < Q \leq \chi \\ c_0\, \chi + c_1(Q - \chi) & Q > \chi \end{cases} \tag{2.1.13}$$

- When adopting an all-units discount policy, if the order size is $Q \geq \chi$, then the entire order is priced at the lower rate c_1, so that Eq. 2.1.13 slightly changes as follows:

$$c(Q) = \begin{cases} c_0\, Q & 0 < Q < \chi \\ c_1\, Q & Q \geq \chi \end{cases} \tag{2.1.14}$$

2.1.3 Robust formulation

Moving back to the classic EOQ model, it is interesting to extend it to the case where some parameters are uncertain: in this case, the goal would be to look for an optimal order quantity, which could guarantee the total cost to be minimum despite possible fluctuations in the parameters affected by uncertainty.

More specifically, we drop Assumption (i) in Section 2.1, which stated that the demand is a known constant a. We still assume that the demand rate is constant, but this constant is unknown. Therefore, according to the terminology introduced by

Taguchi, we consider the order quantity Q as the decision factor, while the demand rate a is the environmental factor; all the other parameters are supposed to be known constants. Many references on inventory management with uncertain parameters are given by Borgonovo and Peccati (2007).

Note: Yu (1997) also assumes an uncertain demand rate, but uses other criteria than we do: he either minimizes the maximum costs or minimizes the maximum percentage deviation from the optimal cost. Moreover he does not assume a probability function for the various scenarios (demand rate values), but uses a 'discrete scenario set'. Altogether, his approach resembles that of Ben-Tal et al., which we discussed in Chapter 1.

The assumption of uncertain constants is often made in deterministic simulation of physical systems; e.g., a nuclear waste-disposal simulation may assume that the permeability of a specific area is constant but unknown; see Kleijnen and Helton (1999). An economic example is the exchange rate between the US dollar and the euro exactly one year from today: that rate is a constant but unknown.

We may collect historical data to infer the probability of the true value of the parameter a. If there is no such data, then we may ask experts for their opinion on the true value of the parameter. This *knowledge elicitation* results in an input distribution (say) $F(a)$. In practice, several distribution types are used, such as normal, lognormal, and uniform; see Kleijnen and Helton (1999) and also Gallego et al. (2007) for inventory problems.

2.1.4 Numerical example

We will apply our methodology to an example from Hillier and Lieberman (2001). We provide a short description of the problem: a television manufacturing company produces its own speakers, then used in the production of its television sets. The television sets are assembled on a continuous production line at a rate of $a = 8000$ per month, with one speaker needed per set. The speakers are produced in batches, incurring a setup cost of $K = 12000$; the unit production cost of a single speaker is $c = 10$, independently of the batch size produced. The speakers are placed into inventory until they are needed for assembly into television sets on the production line; the estimated holding cost of keeping a speaker in stock is $h = 0.30$ per month. The company is interested in determining how many speakers to produce in each batch, and which will be the corresponding total cost for it. The numerical values of the parameters are summarized in Table 2.1.

In our experiments we assume — without loss of generality — that a has a Normal (Gaussian) distribution with mean μ_a and standard deviation σ_a:

$$a \sim \mathcal{N}(\mu_a, \sigma_a). \tag{2.1.15}$$

We assume that μ_a denotes the nominal value used in the classic EOQ model and σ_a quantifies the uncertainty about the true input parameter.

Table 2.1: Nominal values of the parameters, adopted in the numerical example for EOQ.

Parameter	**Symbol**	**Nominal Value**
demand rate	a	8000
setup cost	K	12000
unit ordering cost	c	10
unit holding cost	h	0.30

2.2 (s, S) Inventory model

2.2.1 Problem description

The (s, S) inventory model aims at determining how many items (of a single product) should be ordered to satisfy customers demand, during a specified period of (say) N intervals. The policy states that the inventory is replenished whenever it drops to a value smaller than or equal to the *reorder level* s; the order quantity Q is such that the inventory is raised to the *order-up-to level* S:

$$Q = \begin{cases} S - I & \text{if } I \leq s \\ 0 & \text{if } I > s \end{cases} \tag{2.2.1}$$

where I denotes the inventory level.

Several variations on this model can be found in the literature; some examples concern the following aspects (see Lee and Nahmias, 1993):

- The *review process* could be continuous or periodic: in the first case the level of inventory is known at all times and each demand transaction is recorded as it occurs; otherwise, the inventory level is known only at discrete time instants, therefore reorder decisions can be made only at the predetermined times corresponding either to the beginning or to the end of each period.

- The *demand size* might be deterministic or stochastic.

- The *lead time* could be zero, constant or random: as already discussed for the EOQ model, the simplest assumption is that the lead time is zero, but it is not realistic; therefore, the most common assumption is that the lead time is a fixed constant: this implies that orders are received in the same sequence as they are placed. However, we can also assume the lead time to be a random variable: in such a case, the analysis of the system becomes more complicated, because orders are allowed to cross in time, that is they might arrive in a different sequence with respect to the sequence they have been placed (see Bashyam and Fu, 1998).

- If the demand cannot be fully satisfied by on-hand stock, then *shortages* occur: the excess demand can be either lost or backordered, implying that it will be satisfied by future deliveries.
- When shortages occur, it is possible to consider a *penalty cost* for not being able to satisfy a demand when it occurs: this accounts for the cost of extra record keeping when a backlog exists, as well as loss of customers' goodwill.
- In general, penalty costs are difficult to assess; for this reason, *service level* measures are introduced: it can be evaluated, e.g., through the proportion of demands filled from stock; a lack of fill-rate can also be measured, by computing the fraction of demand on backorder during the period.

Our case study will focus on an infinite horizon periodic review (s, S) inventory system with full backlogging, continuous-valued independent identically distributed (i.i.d.) demands and integer-valued i.i.d. random lead times. The sequence of events that can occur in each period is described as follows:

1. At the beginning of the period orders are received; we emphasize that — given the randomness in the lead time values — the orders are not received according to a FIFO (First In First Out) policy. When an order arrives, it is first used to fulfill the backlogged demand; the remainder of the order (if any) is added to the inventory.
2. During the period customers' demands occur: if supply is at least as large as the demand, then the demand is immediately satisfied; otherwise, the exceeding demand is backlogged.
3. At the end of the period, the company reviews the inventory level and decides how many items to order from its supplier. The number of items to be ordered are determined according to Eq. 2.2.1.

If the company orders Q items, it incurs a cost $C = K + c\,Q$, where K is the setup cost, and c is the incremental cost per item ordered; if $Q = 0$ no cost is incurred. The order lead time follows a Poisson distribution. Apart from the ordering costs, we also take account of holding and shortage costs, which we detail after introducing some additional notation. We define the *inventory level* in each period as the on-hand stock minus backorders; the *inventory position* is the inventory level plus outstanding orders. The former is considered to fulfill demand from customers and is the one determining the holding costs; the latter plays a role in the review process, to determine whether an order is needed or not. Let V_n be the inventory level in period n before demand subtraction and W_n be the inventory level in period n after demand subtraction; then $V_n = W_n + D_n$, where D_n is the demand in period n. $B_n = D_n - min(D_n, V_n)$ denotes the demand in period n not satisfied from on-hand stock, i.e. the backorder. The inventory position in period n is denoted by I_n. We assume that the company incurs a holding cost of h per item per period held in inventory. Similarly, a backlog cost of c_b per item per period in backlog is assumed.

The total cost in each period is given by the sum of the holding cost, the ordering cost (including setup costs) and the shortage cost incurred during the observed time period n:

$$C_n = hW_n^+ + \mathcal{I}_{I_n \leq s}\left[K + c(S - I_n)\right] + c_b\left(D_n - min(D_n, V_n)\right) , \tag{2.2.2}$$

where $W_n^+ = \max(0, W_n)$ and $\mathcal{I}(\cdot)$ denotes the indicator function.

We assume that the initial inventory level is $I(0) = S$ and that no order is outstanding.

The objective of the classical optimization problem is to determine the optimal policy allowing to achieve the minimum total cost over the planning horizon.

Constrained formulation for the simulation-optimization problem

Based on the discussion at the beginning of the section, we can slightly reformulate the problem presented before as follows: we do not consider any shortage costs, since obtaining a "reliable" estimation of this parameter is not an easy task; therefore, the total cost in each period will be determined only by holding and ordering costs. A service level constraint is introduced to account for the 'cost' of losing customers' goodwill; more specifically, we adopt the so-called γ-service level, which is defined such that

$$1 - \gamma = \frac{\sum_{n=1}^{N} B_n}{\sum_{n=1}^{N} D_n} . \tag{2.2.3}$$

This measure is also adopted in Bashyam and Fu (1998), by complementing the fill-rate measure considered in Tijms and Groenevelt (1984). Based on (2.2.3), we solve the problem of minimizing the total cost guaranteeing that the service level is not smaller than a prescribed value.

2.2.2 Possible extensions of the basic (s, S) model

As the assumptions of the basic (s, S) model are frequently satisfied, it is widely used in practice. However, starting from this model, it is easy to formulate some others, which allow us to relax some assumptions moving towards new models in order to give a better representation of a specific real problem. Indeed, a first extension is represented by the introduction of the service level as an additional objective or constraint. In what follows we consider some other possible extensions of the basic (s, S) model.

Inventory record inaccuracy

Inventory record inaccuracy is a quite common problem in the industrial practice (Fleish and Tellkamp, 2005; Kök and Shang, 2007). Although companies have abundantly invested to automate and improve their inventory management processes, inventory records and physical inventory are seldom aligned. There are several reasons

for these discrepancies: stock loss or shrinkage, transaction errors, and product misplacement. Such misalignment may create significant problems and losses for companies. The direct effect of inventory record inaccuracy is losses resulting from ineffective inventory order decisions. In fact, when an out-of-stock item is reported as in stock, an automated replenishment system may not reorder that product, which may result in higher backorder penalties or lost sales. On the other end, if the records show fewer items than the physical inventory level, more items will be ordered of that product, leading to higher inventory costs. A stochastic model of this inaccuracy could be included in the inventory representation based on the (s, S) model. To model such inaccuracy both aleatory and epistemic aspects of the uncertainty could be considered. Of course they will be related to the organization of the inventory management and the available technology (Fleish and Tellkamp, 2005; Kök and Shang, 2007).

Space Constraint

The continuous-review inventory policies are popular in inventory management. In such cases, the system is continuously monitored, and whenever the inventory position drops to or below a given level, a certain amount of units of goods is ordered to replenish the system. To be able to implement the policy, the system needs a storage space capable to store the maximum inventory. In real systems resources such as the storage space are usually limited (Allgor et al., 2005; Gallego and Scheller-Wolf, 2000; Zhao et al., 2007). Then, an issue in planning and operations management is the optimal inventory policy with capacitated storage space. This extension of (s, S) model could be relevant in multi-item and multi-stage inventory systems (Allgor et al., 2005), and in cases characterized by inventory record inaccuracy (Kök and Shang, 2007; Zhao et al., 2007).

Setup Variability

Traditionally, inventory models assume that setup costs are constant. However, in practice, setup costs may vary and they could be another source of uncertainty in the (s, S) model (Darwish, 2008; Sarker and Coates, 1997). As reduced setup costs with the accompanying possibility of smaller lot sizes have numerous organizational and economic benefits, companies are often active in pursuing such reductions. These companies investments and efforts to reduce setups lead to learning-forgetting models for the setup cost (Darwish, 2008; Sarker and Coates, 1997). These models introduce a deterministic or stochastic variability in the setup costs and, at the same time, the need to consider both aleatory and/or epistemic uncertainties.

2.2.3 Robust formulation

In order to deal with the robustness issue in the simulation-optimization of the (s, S) inventory model, we have to tackle some sources of uncertainty: more specifically,

we have to distinguish *aleatory* and *epistemic* uncertainty, as discussed in the Introduction of this thesis. Given the stochasticity of the model under study, accounting for the aleatory uncertainty and ignoring the epistemic uncertainty implies that the demand and lead time random variables belong to some specified distributions, with known parameters. A classic optimization approach like the one described in the previous subsection would deal with such a case.
Robustness, however, may ask for one step further: we consider the epistemic uncertainty affecting the parameters of the distributions for demand and lead time, which are not supposed to be constant values anymore; these parameters, instead, can take any value within a given interval, according to a so-called *prior distribution*. If we assume to have collected historical data on the random variables of interest for a reasonably large period of time, then we can select a normal distribution as a prior distribution for our random variables, by virtue of the Central Limit Theorem.

2.2.4 Numerical example

We will perform our experiments based on a model that was originally studied by Bashyam and Fu (1998). This model assumes that the size of customers' demand per time period is exponentially distributed with mean $\lambda_D = 100$ units; delivery lead times are Poisson distributed with mean $\lambda_L = 6$ time periods.

Initially, $I_0 = S$ items are on hand. The range for the reorder point is $900 \leq s \leq 1250$ and that for the order-up-to level is $901 \leq S \leq 1750$ (where the lower bound for S is fixed in order to satisfy the implicit constraint that $S > s$).

The setup cost is $K = 36$, the incremental ordering cost per item is $c = 2$, the holding cost is $h = 1$ per item per period and the shortage cost is $c_b = 5$ per item per period.

The system is observed for $N = 30000$ periods, as this was proved to be sufficient to achieve steady-state conditions (see Bashyam and Fu, 1998). We make $m = 10$ independent replications of the simulation for all the input combinations (s, S).

The numerical values of the parameters are summarized in Table 2.2.

Table 2.2: Nominal values of the parameters in the numerical example for (s, S) model.

Parameter	**Symbol**	**Nominal Value**
mean demand size	λ_D	100
mean lead time	λ_L	6
setup cost	K	36
unit ordering cost	c	2
unit holding cost	h	1
unit shortage cost	c_b	5

For the 'epistemic' robust formulation, the parameters of the prior normal dis-

tribution are defined such that μ_i denotes the nominal value of the corresponding random variable (demand or lead time) used in the classic version of the (s, S) model and σ_i quantifies the uncertainty about the true input parameter.

Chapter 3

Robustness combining Taguchi's method and RSM

This chapter is devoted to describe a methodology to achieve robustness in simulation-optimization, combining Taguchi's view of the uncertain world and the Response Surface Methodology (RSM) to provide approximation of the (possibly) expensive simulation model.
The chapter is structured as follows: Section 3.1 summarizes the main aspects characterizing the RSM technique — mainly focussing on second-order regression models, since these are the ones we will use in our experiments. Then, a possible scheme to use RSM metamodels within an optimization process is mentioned. Finally, a novel approach is proposed in Section 3.2, basically inspired by Myers and Montgomery (2002) but removing some restrictive assumptions: in this way, we keep the methodology more widely applicable; Dellino et al. (see 2008a). The computational experiments will be described in Chapters 5 and 6. Section 3.3 adds some comments on the normality distribution which is often assumed for the simulation output.

3.1 Response Surface Methodology (RSM)

Response Surface Methodology (RSM) is a method proposed by Box and Wilson (1951), originally applied to real (i.e., non-simulated systems): based on a sequential procedure, it aims at optimizing the output function of the system through a *local* approximation model which is valid over a small region identified by properly designed experiments. Typically, RSM locally fits low order polynomial regression models. Then, for a minimization (resp. maximization) problem a steepest descent (resp. ascent) search algorithm is adopted to explore the experimental region moving towards more promising (sub)areas to find the optimum. Notice that in the following we will not consider the iterative search phase, assuming that it has already been performed leading to the experimental region we will be working on; in the remainder of the

dissertation we will refer to such approach as a *simplified* RSM.

The general expression for a second-order regression model is:

$$y = \beta_0 + \sum_{j=1}^{n} \beta_j x_j + \sum_{j=1}^{n} \sum_{j' \geq j}^{n} \beta_{j;j'} x_j x_{j'} \,, \tag{3.1.1}$$

where n is the size of the input design space, i.e., the number of input variables x_j. Eq. 3.1.1 can be written also in a more compact form:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} \,, \tag{3.1.2}$$

where $\mathbf{X}$ is the $n \times q$ matrix of explanatory variables, with q denoting the number of parameters collected in $\boldsymbol{\beta}$; $\boldsymbol{\beta}$ is the vector of the regression parameters, whose elements should be defined according to the form of the $\mathbf{X}$ matrix. This $\mathbf{X}$ is assumed not to be collinear (otherwise, $(\mathbf{X}'\mathbf{X})^{-1}$ would not exist). So a necessary but not sufficient condition for $\mathbf{X}$ is $n > q$. The matrix $\mathbf{X}$ can be column-partitioned as follows:

$$\begin{bmatrix} \mathbf{1} & \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n & \mathbf{x}_1^2 & \dots & \mathbf{x}_n^2 & \mathbf{x}_1\mathbf{x}_2 & \dots & \mathbf{x}_{n-1}\mathbf{x}_n \end{bmatrix} . \tag{3.1.3}$$

An estimation of the vector $\boldsymbol{\beta}$ is usually provided by the so-called Ordinary Least Squares (OLS) method, which solves the following problem:

$$\min_{\widehat{\boldsymbol{\beta}}} \left\| \mathbf{w} - \mathbf{X}\widehat{\boldsymbol{\beta}} \right\|_2^2 , \tag{3.1.4}$$

where the vector $\mathbf{w}$ consists of the n simulation outputs. Therefore, setting the gradient of the objective function to zero, we obtain an explicit formula for the OLS estimate:

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{w} \,. \tag{3.1.5}$$

In practice, Eq. 3.1.5 is not directly used to solve the problem; instead, the computation of the OLS estimate $\widehat{\boldsymbol{\beta}}$ is usually performed by solving the following linear system of equations:

$$\mathbf{X}^T\mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{X}^T\mathbf{y} \,, \tag{3.1.6}$$

which is referred to as the set of *normal equations* for the problem in Eq. 3.1.4. Different numerical methods can be used to solve Eq. 3.1.6, and the choice among them needs to be carefully evaluated, since some problems related to numerical accuracy might arise (see Nocedal and Wright, 2006). We adopt the QR factorization of the matrix $\mathbf{X}^T\mathbf{X}$: as Nocedal and Wright (2006) point out, this method is usually reliable, because the solution found is affected by a relative error which is proportional to the condition number of $\mathbf{X}^T\mathbf{X}$. However, some situations might require a more robust method, namely the Singular Value Decomposition (SVD) approach, which is computationally expensive, but can provide some useful information for particular problems; see Nocedal and Wright (2006) for more details. In our experiments, we monitor the condition number of the matrix $\mathbf{X}^T\mathbf{X}$, to check the numerical accuracy of the results we obtain.

3.1.1 Classic simulation-optimization using *simplified* RSM

Once a second-order regression model has been fitted, according to Eq. 3.1.1, it can be used in a classic simulation-optimization problem, assuming no uncertainty in the parameters: the procedure usually adopted in such a context is outlined in Fig. 3.1. Notice that step 5 asks for choosing a solver enabling the use of a metamodel as the objective function.

Classic Simulation-Optimization with *simplified* RSM

Input: Design of Experiments $\mathbf{D} = (x_{ij})$, $\mathbf{x}_i \in \mathbb{R}^n$, $(i = 1, \ldots, m)$;
begin
  for each design point $\mathbf{x}_i$ in $\mathbf{D}$, $i = 1, \ldots, m$
    1a. Run the simulation model
    1b. Compute $w_i = f(x_i)$
  2. Fit a regression metamodel for the response, $\widehat{y}$
  3. Validate the metamodel, using leave-one-out cross validation
  4. Test the importance of the estimated effects, using t-statistics
  5. Solve the optimization problem
end

Figure 3.1: Classic simulation-optimization process using regression metamodels.

Besides some steps discussed in the next Section, we will not further analyze this topic, which is extensively examined in Myers and Montgomery (2002). However, it was worth mentioning this simple procedure before moving to the robust approach, where uncertainty on some input factors is taken into account.

3.2 RSM and robust optimization

Based on what discussed in Chapter 1, we adopt the Taguchian view of the uncertain world — without restricting our attention to the statistical techniques he proposed, mainly based on computing SNRs as a measure of robustness and crossed-array designs (see Nair et al., 1992; del Castillo, 2007). Therefore, we distinguish between control or decision factors and noise or environmental factors, according to the definition given in Chapter 1; we recall that we denote the control factors by $\mathbf{d}$ while the environmental factors are denoted by $\mathbf{e}$.

Fig. 3.2 briefly sketches the main steps of our Robust Optimization (RO) procedure, combining Taguchi's method and RSM. Each step is detailed in what follows.

Before building any approximation model, we need to properly design the experiments in the input space according to some sampling techniques based on Design of Experiments (DoE); different types of DoE can be adopted but this discussion lies

Robust Optimization combining Taguchi & *simplified* RSM

Input: design ($\mathbf{D}$, $n_{\mathbf{d}} \times n_{\mathbf{e}}$) in the control-by-noise space $(\mathbf{d}, \mathbf{e})$;
probability distribution of noise factors;

begin
for each design point (d_j, e_j) in $\mathbf{D}$, $j = 1, \ldots, n_{\mathbf{d}} \times n_{\mathbf{e}}$
1a. Run the simulation model
1b. Compute $w_j = f(d_j, e_j)$
2. Fit a regression metamodel for the mean of the response, $\widehat{y}$, and one for its standard deviation, $\widehat{\sigma_y}$
3. Validate the metamodels, using leave-one-out cross validation
4. Test the importance of the estimated effects, using t-statistics
5. Estimate the Pareto frontier, solving constrained optimization problems
for each $b \in \{1, \ldots, B\}$
6. Apply parametric bootstrap on the regression coefficients
Repeat step 5 B times, using the bootstrapped values of the coefficients
end

Figure 3.2: Robust Optimization approach through RSM.

beyond the scope of the present work (see Montgomery, 2009). When planning an experiment, we should distinguish between the following three kinds of regions:

- The *region of operation* is the complete space defined by the input factors $\mathbf{x}_i$, in their original units; it represents the entire area over which the experiments can theoretically be conducted.
- Usually, some physical or safety constraints are defined over the region of operation, thus restricting the area to explore through the experiments to the so-called *region of interest*.
- We further distinguish the region of interest from the *experimental region*, that is defined by the ranges of the input factors once the experiment has been designed. Notice that the experimental region should resemble the region of interest.

We will adopt this terminology throughout the dissertation; for further details, refer to del Castillo (2007).
At this stage, we adopt two separate designs for decision and environmental factors: in particular, we pick n_{d_j} equally spaced values for the decision factor d_j and we sample n_e values for each environmental factor e_k from its corresponding probability distribution — which is assumed to be known. More specifically, we sample the environmental factors values using Latin Hypercube Sampling (LHS), whose algorithm is briefly described in the following:

1. Divide the experimental range of each factor e_k into n_e intervals of equal marginal probability $1/n_e$.

2. Sample one value from each interval, according to the probability density function over that interval; let this sample be denoted as e_{kl}, $l = 1, \ldots, n_e$.

3. The components of the various e_k's are matched together by associating a random permutation of the first n_e integers with each variable.

It is important to notice that sometimes we may have to deal with truncated random variables, that is random variables which follow a given probability distribution, but they are forced to stay above and/or below some specified values. In such a case, we can explicitly compute the truncated distribution by observing that

$$f(x|x_L < x < x_U) = \frac{f(x)}{P(x_L < x < x_U)}\,, \tag{3.2.1}$$

where $f(x)$ is the probability density function (pdf) of the original (i.e., not truncated) random variable, and x_L and x_U are the lower and upper truncation point, respectively. If x_L is replaced by $-\infty$, or x_U by $+\infty$, the distribution is singly truncated from above, or below, respectively; nevertheless, the results discussed in the following remain valid, after some simplifications.
Suppose, for instance, that an environmental factor follows a Normal distribution, $e \sim \mathcal{N}(\mu, \sigma^2)$, but it can only take values between e_L and e_U; therefore:

$$\begin{aligned} P(e_L \le e \le e_U) &= P(e \le e_U) - P(e \le e_L) = \\ &= \Phi\left(\frac{x_U - \mu}{\sigma}\right) - \Phi\left(\frac{x_L - \mu}{\sigma}\right) = \Phi(z_U) - \Phi(z_L)\,, \end{aligned} \tag{3.2.2}$$

where $z_U = (x_U - \mu)/\sigma$, $z_L = (x_L - \mu)/\sigma$ and $\Phi(\cdot)$ is the standard normal cumulative distribution function (cdf). The pdf of the truncated normal distribution is then:

$$\begin{aligned} f_{tr}(e) = f(e|e_L < e < e_U) = \frac{f(e)}{\Phi(z_U) - \Phi(z_L)} = \frac{(2\pi\sigma^2)^{-1/2} e^{-\dfrac{(x-\mu)^2}{2\sigma^2}}}{\Phi(z_U) - \Phi(z_L)} = \\ = \frac{\sigma^{-1}\varphi\left(\dfrac{x - \mu}{\sigma}\right)}{\Phi(z_U) - \Phi(z_L)}\,, \end{aligned} \tag{3.2.3}$$

where $\varphi(\cdot)$ is the standard normal pdf. More specifically, to take into account that the truncated pdf is defined only for $e_L < e < e_U$, we can use the following compact notation:

$$f_{tr}(e) = \frac{\sigma^{-1}\varphi\left(\dfrac{x - \mu}{\sigma}\right)}{\Phi(z_U) - \Phi(z_L)}\, \mathcal{I}_{(e_L, e_U)}(e)\,, \tag{3.2.4}$$

where $\mathcal{I}_{(e_L,e_U)}(x)$ denotes the indicator function.
Further details on truncated distributions can be found in Greene (2003) and Johnson et al. (1994).

After deriving the analytical expression for the truncated distribution, we have to implement the algorithm to sample values from it, according to LHS. The sampling technique is described in Iman et al. (1979) and we follow the approach suggested by Stein (1987), since it results in an efficient algorithm implementation. The main steps for producing a LHS design of size N are described in what follows, assuming that we have K input variables, and F_k denotes the cdf of the k-th variable:

i. Generate the $N \times K$ matrix $P = (p_{nk})$, $n = 1, \ldots, N$, $k = 1, \ldots, K$, where each column of P is an independent random permutation of $\{1, \ldots, N\}$.

ii. Generate the $N \times K$ matrix $\Xi = (\xi_{nk})$, $n = 1, \ldots, N$, $k = 1, \ldots, K$, where ξ_{nk} is a random number, uniformly distributed on $[0, 1]$.

iii. Each sample is obtained as

$$x_{nk} = F_k^{-1}\left(\frac{1}{N}(p_{nk} - 1 + \xi_{nk})\right). \tag{3.2.5}$$

Notice that, despite the description of the algorithm given before, the implementation does not require to explicitly compute the extreme points of each interval. For further details, refer to Iman et al. (1979).
Eq. 3.2.5 implies that we need to derive the inverse cdf for the truncated normal distribution. Based on Eq. 3.2.4 and observing that the denominator is a constant for known truncation point, we can simply rewrite this equation as

$$f_{tr}(x) = \begin{cases} c_1 \cdot f(x) & \text{if } x_L \le x \le x_U \\ 0 & \text{elsewhere} \end{cases} = c_1 \cdot f(x)\mathcal{I}_{(x_L \le x \le x_U)}(x)\,,$$

where $c_1 = 1/(\Phi(z_U) - \Phi(z_L))$. Therefore, it follows that the truncated cdf is easily derived as follows:

$$F_{tr}(x) = \int_{-\infty}^{x} f_{tr}(y)dy = c_1 \int_{x_L}^{x} f(y)dy = c_1[F(x) - F(x_L)] = c_1 F(x) - c_2\,, \tag{3.2.6}$$

where $c_2 = c_1 F(x_L) = \Phi(z_L)/\left(\Phi(z_U) - \Phi(z_L)\right)$.

Now we have to invert $F_{tr}(x)$, given that $F^{-1}(x)$ is known. Using the definition of the inverse function, it follows — after a few algebraic computations — that:

$$F_{tr}^{-1}(p) = F^{-1}\left(\frac{p + c_2}{c_1}\right). \tag{3.2.7}$$

After obtaining the designs for the decision and environmental factors, we "crossed" them, according to the Taguchian terminology: this implies that we merge each combination of the decision factors with each combination of the environmental factors;

the overall design therefore results in a $n_d \times n_e$ matrix of (decision $\times$ environmental) factor combinations.

The factor combinations resulting from the DoE are then used as inputs to the simulation model, which we implement using Rockwell's Arena simulation software (see Kelton et al., 2007). The simulation runs produce a set of input/output (I/O) combinations which are used to build a metamodel.

To analyze our simulation experiments we use RSM following Myers and Montgomery (2002). RSM extends Taguchi's simpler statistical techniques — as discussed in Chapter 1. The simplest RSM metamodel is a polynomial of degree as low as possible:

- Because we wish to estimate the optimal combination(s) of the decision factors d_j ($j = 1, \ldots, n_{\mathbf{d}}$), we fit a *second-order* polynomial for these factors.
- Moreover, we wish to model possible effects of the environmental factors e_k ($k = 1, \ldots, n_{\mathbf{e}}$); we fit a first-order polynomial for these factors.
- Finally, we wish to estimate interactions between the two types of factors, so we fit *'control-by-noise' two-factor interactions*. Notice that interaction between d_j and e_k implies nonparallel response surfaces for d_j — given different values for e_k.

Altogether Myers and Montgomery (2002) propose the following metamodel:

$$\begin{aligned} y &= \beta_0 + \sum_{j=1}^{n_{\mathbf{d}}} \beta_j d_j + \sum_{j=1}^{n_{\mathbf{d}}} \sum_{j' \geq j}^{n_{\mathbf{d}}} \beta_{j;j'} d_j d_{j'} + \sum_{k=1}^{n_{\mathbf{e}}} \gamma_j e_j + \sum_{j=1}^{n_{\mathbf{d}}} \sum_{k=1}^{n_{\mathbf{e}}} \delta_{j;k} d_j e_k + \epsilon \\ &= \beta_0 + \boldsymbol{\beta}'\mathbf{d} + \mathbf{d}'\mathbf{B}\mathbf{d} + \boldsymbol{\gamma}'\mathbf{e} + \mathbf{d}'\boldsymbol{\Delta}\mathbf{e} + \epsilon\,, \end{aligned} \tag{3.2.8}$$

where y denotes the regression predictor of the simulation output w, ϵ denotes the residual with $E(\epsilon) = 0$ if this metamodel has no *lack of fit* (this zero mean should be tested; see cross-validation below) and constant variance σ_ϵ^2 (an unrealistic assumption in simulation experimentation), and the bold symbols are the vectors and matrices that are defined in the obvious way (e.g., $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{n_{\mathbf{d}}})'$ and $\mathbf{B}$ denotes the $n_{\mathbf{d}} \times n_{\mathbf{d}}$ symmetric matrix with main-diagonal elements $\beta_{j;j}$ and off-diagonal elements $\beta_{j;j'}/2$).

It is convenient and traditional in DoE to use *coded* — also called *standardized* or *scaled* — factor values. Let the 'original' factor be denoted by ξ (ξ corresponds with d_j or e_k in (3.2.8)); then the coded variable x use the linear transformation

$$x = \frac{\xi - \xi_{lb}}{\xi_{ub} - \xi_{ub}} (x_{ub} - x_{lb}) + x_{lb}\,, \tag{3.2.9}$$

where $[\xi_{lb}, \xi_{ub}]$ is the original range and $[x_{lb}, x_{ub}]$ is the range we want to map ξ into, i.e. it gives the interval where the coded variable x will range. Being interested in coding the original variables into the interval $[-1, +1]$, the adopted coding scheme can be simplified as follows:

$$x = a + b\xi \text{ with } a = \frac{x_{lb} + x_{ub}}{x_{lb} - x_{ub}} \text{ and } b = \frac{2}{x_{ub} - x_{lb}}\,. \tag{3.2.10}$$

If ξ is a random variable (like e), then this coding implies $var(x) = b^2 var(\xi)$. The numerical accuracy of the estimates may be affected by coding; we focus on the estimated effects of the coded variables. Coding is further discussed by Kleijnen (2008).

Assuming a model like (3.2.8), Myers and Montgomery (2002) derive the mean and the variance of y (the regression predictor of the simulation output w), after averaging over the noise factors — and assuming that the environmental variables $\mathbf{e}$ satisfy

$$E(\mathbf{e}) = \mathbf{0} \text{ and } \mathbf{cov}(\mathbf{e}) = \sigma_e^2 \mathbf{I}. \tag{3.2.11}$$

Given (3.2.11), they derive

$$E(y) = \beta_0 + \boldsymbol{\beta}'\mathbf{d} + \mathbf{d}'\mathbf{B}\mathbf{d} \tag{3.2.12}$$

and

$$var(y) = \sigma_e^2(\boldsymbol{\gamma}' + \mathbf{d}'\boldsymbol{\Delta})(\boldsymbol{\gamma} + \boldsymbol{\Delta}'\mathbf{d}) + \sigma_\epsilon^2 = \sigma_e^2 \mathbf{l}'\mathbf{l} + \sigma_\epsilon^2, \tag{3.2.13}$$

where $\mathbf{l} = (\boldsymbol{\gamma} + \boldsymbol{\Delta}'\mathbf{d}) = (\partial y/\partial e_1, \ldots, \partial y/\partial e_c)'$; i.e., $\mathbf{l}$ is the gradient with respect to the environmental factors — which follows directly from (3.2.8). So, the larger the gradient's elements are, the larger the variance of the predicted simulation output is — which stands to reason. Furthermore, if $\boldsymbol{\Delta} = \mathbf{0}$ (no control-by-noise interactions), then $var(y)$ cannot be controlled through the control variables $\mathbf{d}$.

Equation (3.2.13) implies that the predicted simulation output y has heterogeneous variances — even if σ_e^2 and σ_ϵ^2 were constants — because changing the control factors $\mathbf{d}$ changes $var(y)$. Whereas Myers and Montgomery (2002) present examples with $\sigma_e^2 = \sigma_\epsilon^2/2$, Kleijnen (2008) gives a supply-chain simulation model with $\sigma_e^2 = 10\sigma_\epsilon^2$. Most important is the gradient $\mathbf{l}$, because it shows the key role played by the control-by-noise interactions; i.e., to reduce the predicted output's variance $var(y)$(or σ_y^2) the analysts should take advantage of the interactions $\boldsymbol{\Delta}$; they cannot control the main effects of the noise factors ($\boldsymbol{\gamma}$) and the variances of the noise factors and the residuals (σ_e^2 and σ_ϵ^2). For example, if a particular decision factor (say, d_1) has no effects on the mean output (so $\beta_1 = \beta_{1;1} = \beta_{1;2} = \ldots = \beta_{1;k} = 0$) but has important interactions with the noise factors (e.g., $\delta_{1;2} \gg 0$), then this interaction can be utilized to decrease the output variance (e.g., decrease σ_y^2 by decreasing d_1). If there are multiple decision factors, then the following solution method may be tried:

1. select the values of some decision factors such that $\mathbf{l} = \mathbf{0}$, so $var(y)$ in (3.2.13) is minimized;

2. select the remaining decision factors such that the predicted mean output $E(y)$ in (3.2.12) gets the desired value.

Obviously, assuming zero means and constant variances could be unrealistic, so we replace Myers and Montgomery's assumption formulated in Eq. 3.2.11 by the following:

$$E(\mathbf{e}) = \boldsymbol{\mu}_\mathbf{e} \text{ and } \mathbf{cov}(\mathbf{e}) = \boldsymbol{\Omega}_\mathbf{e}. \tag{3.2.14}$$

Then Eq. 3.2.12 becomes

$$E(y) = \beta_0 + \boldsymbol{\beta}'\mathbf{d} + \mathbf{d}'\mathbf{B}\mathbf{d} + \boldsymbol{\gamma}'\boldsymbol{\mu}_\mathbf{e} + \mathbf{d}'\boldsymbol{\Delta}\boldsymbol{\mu}_\mathbf{e} \tag{3.2.15}$$

and Eq. 3.2.13 becomes

$$var(y) = (\boldsymbol{\gamma}' + \mathbf{d}'\boldsymbol{\Delta})\boldsymbol{\Omega}_\mathbf{e}(\boldsymbol{\gamma} + \boldsymbol{\Delta}'\mathbf{d}) + \sigma_\epsilon^2 = \mathbf{l}'\boldsymbol{\Omega}_\mathbf{e}\mathbf{l} + \sigma_\epsilon^2. \tag{3.2.16}$$

Hereinafter, we will not take account of Eqs. 3.2.12-3.2.13 anymore.

To *estimate* the unknown (regression) parameters in (3.2.8) — which also gives the parameters in the mean and variance equations (3.2.15) and (3.2.16) — we reformulate (3.2.8) as the following *linear regression model*:

$$y = \boldsymbol{\zeta}'\mathbf{x} + \epsilon \tag{3.2.17}$$

with $\boldsymbol{\zeta} = (\beta_0, \boldsymbol{\beta}, \mathbf{b}, \boldsymbol{\gamma}, \boldsymbol{\delta})'$ where $\mathbf{b}$ denotes the vector with the $n_\mathbf{d} \times (n_\mathbf{d} - 1)/2$ interactions between the decision factors plus their $n_\mathbf{d}$ purely quadratic effects, and $\boldsymbol{\delta}$ denotes the $n_\mathbf{d} \times n_\mathbf{e}$ control-by-noise interactions; $\mathbf{x}$ is defined in the obvious way (e.g., the element corresponding with the interaction effect $\beta_{1;2}$ is $d_1 d_2$). The regression coefficients are estimated through the OLS given in Eq. 3.1.5, using the `regress` function from the MATLAB Statistics Toolbox (see The MathWorks Inc., 2005b). Notice that in this case the matrix of explanatory variables, $\mathbf{X}$, involves both decision and environmental factors: thus, here n denotes the number of scenarios (combinations of decision and environmental factors) determined by DOE that are actually simulated.

To estimate the left-hand side of (3.2.15), we simply plug in the estimators for β_0, $\boldsymbol{\beta}$, $\mathbf{B}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\Delta}$ in the right-hand side (the factors $\mathbf{d}$ and $\boldsymbol{\mu}_\mathbf{e}$ are known). To estimate the left-hand side of (3.2.16), we again use plug-in estimators—now for $\boldsymbol{\gamma}$, $\boldsymbol{\Delta}$, and σ_ϵ^2 (the factor $\boldsymbol{\Omega}_\mathbf{e}$ is known because the environmental factors are sampled from a known distribution); see Myers and Montgomery (2002).

In order to apply classic OLS results, we assume that σ_w^2 is constant, the outputs for different scenarios are independent, and the environmental factors are fixed (Myers and Montgomery (2002) do not make these assumptions explicit; in our EOQ application, where we are interested in optimizing mean and variance of the total cost, we can derive the true Pareto optimum corresponding to these two objectives, so we can verify how sensitive our analysis is to these assumptions). Then the classic estimator of σ_ϵ^2 is the Mean Squared Residuals (MSR)

$$MSR = \frac{SSR}{n-q} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-q} = \frac{(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\zeta}})^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\zeta}})}{n-q}, \tag{3.2.18}$$

where $SSR = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ defines the Sum of Squared Residuals, $\mathbf{y}$ is the vector of observed values for the output function and $\widehat{\mathbf{y}} = \mathbf{X}\widehat{\boldsymbol{\zeta}}$ is the vector of estimated values according to the regression model; $(n-q)$ are the degrees of freedom left after fitting the model, with q expressing the number of regression coefficients (including β_0); also see Kleijnen (2008). It should be noticed that the estimate of σ_ε^2 provided

by the MSR in (3.2.18) is model dependent, since it depends on the structure of the model used to fit the data; nevertheless Myers and Montgomery (2002) point out that if replicates are available — as it happens when working on random simulations — then a model independent estimate of σ_ε^2 can be obtained.

We point out that (3.2.16) has products of unknown parameters, so it implies a *nonlinear estimator* $\widehat{\sigma_y^2}$ (we are also interested in $\widehat{\sigma_y} = \sqrt{\widehat{\sigma_y^2}}$, a nonlinear transformation of $\widehat{\sigma_y^2}$) so this plug-in estimator is certainly biased; we ignore this bias when estimating the Pareto frontier that balances $\widehat{y}$ and $\widehat{\sigma_y}$.

The estimation of the mean and variance of the simulation output through (3.2.15) and (3.2.16) raises the following crucial question (also raised by Myers and Montgomery (2002), but assuming a constant output variance): is the underlying RSM model (3.2.8) an *adequate approximation*? The linear regression literature presents several methods for answering this question; see Kleijnen (2008). We focus on a method that is also applied outside linear regression (e.g. in Kriging), namely cross-validation. There are several variations on cross-validation (see Iooss et al. (2007) and Meckesheimer et al. (2002)), but the most popular variant is *leave-one-out cross-validation*. Following Kleijnen (2008), we define this cross-validation as follows:

1. Delete I/O combination i from the complete set of n combinations, to obtain the remaining I/O data set — denoted by $(\mathbf{X}_{-i}, \mathbf{w}_{-i})$. Assume that this step results in an $(n-1) \times q$ noncollinear matrix $\mathbf{X}_{-i}$ $(i = 1, \ldots, n)$; a necessary condition is $n > q$. Obviously, $\mathbf{w}_{-i}$ denotes the $(n-1)$-dimensional vector with the remaining $(n-1)$ simulation outputs.

2. Recompute the OLS estimator of the regression parameters in (3.1.5):

$$\widehat{\boldsymbol{\zeta}_{-i}} = (\mathbf{X}'_{-i}\mathbf{X}_{-i})^{-1}\mathbf{X}'_{-i}\mathbf{w}_{-i}\,. \tag{3.2.19}$$

3. Use $\widehat{\boldsymbol{\zeta}_{-i}}$ (recomputed regression parameters) to compute $\widehat{y_{-i}}$, which denotes the regression predictor of the simulation output generated by $\mathbf{x}_i$ (which corresponds with the simulation input of the combination deleted in step 1):

$$\widehat{y_{-i}(\mathbf{x}_i)} = \mathbf{x}'_i\widehat{\boldsymbol{\zeta}_{-i}}\,. \tag{3.2.20}$$

4. Repeat the preceding three steps, until all n combinations have been processed. This results in n predictions $\widehat{y_{-i}}$ $(i = 1, \ldots, n)$.

5. Use a scatterplot with the n pairs $(w_i, \widehat{y_{-i}})$ to judge whether the metamodel is valid, where w_i denotes the simulation output at $\mathbf{x}_i$.

6. Because the scaling of this scatterplot may give the wrong impression, we also evaluate the relative prediction errors $\widehat{y_{(-i)}}/w_i$.

7. A valid regression model also implies that the estimated regression coefficients do not change much when deleting an I/O combination; i.e., there is not much change in $\widehat{\boldsymbol{\zeta}_{-i}}$ with $(i = 0, 1, \ldots, n)$ where $\widehat{\boldsymbol{\zeta}_{-0}}$ denotes the estimator when zero combinations are deleted, so $\widehat{\boldsymbol{\zeta}_{-0}} = \widehat{\boldsymbol{\zeta}}$.

When the simulation model is stochastic (instead of being deterministic), it is strongly recommended to make several independent runs (i.e. replications) of the simulation for each input combination. Although it is not of interest at this stage to further detail this topic — which will be discussed in Chapter 6 — it is important to mention that, when replications are available, the cross-validation procedure described before remains applicable, after replacing the output w_i corresponding to the i-th input combination $\mathbf{x}_i$ by the average

$$\overline{w(\mathbf{x}_i)} = \frac{1}{m} \sum_{r=1}^{m} w_r(\mathbf{x}_i)\,, \tag{3.2.21}$$

where m denotes the total number of replications and $w_r(\mathbf{x}_i)$ is the simulation output for the i-th input combination $\mathbf{x}_i$ in the r-th replication.
We also point out that having replicated output data allows an alternative evaluation of the metamodel accuracy w.r.t. the one proposed in step 5, based on the *Studentized prediction error*, as proposed by Kleijnen (1983a): for the i-th combination $\mathbf{x}_i$ left out during the cross-validation procedure we compute

$$t_{m-1}^{(i)} = \frac{\overline{w(\mathbf{x}_i)} - \widehat{y_{-i}(\mathbf{x}_i)}}{\sqrt{\widehat{var(\overline{w(\mathbf{x}_i)})} + \widehat{var(y_{-i}(\mathbf{x}_i))}}} \qquad i = 1, \ldots, n \tag{3.2.22}$$

where $\overline{w(\mathbf{x}_i)}$ and $\widehat{y_{-i}(\mathbf{x}_i)}$ are given by (3.2.21) and (3.2.20) respectively,

$$\widehat{var(\overline{w(\mathbf{x}_i)})} = \widehat{var(w(\mathbf{x}_i))}/m\,, \tag{3.2.23}$$

$$\widehat{var(w(\mathbf{x}_i))} = \frac{1}{m-1} \sum_{r=1}^{m} \left(w_r(\mathbf{x}_i) - \overline{w(\mathbf{x}_i)} \right)^2\,, \tag{3.2.24}$$

and

$$\widehat{var(y_{-i}(\mathbf{x}_i))} = \mathbf{x}_i' \widehat{\mathbf{cov}(\boldsymbol{\zeta}_{-i})} \mathbf{x}_i\,, \tag{3.2.25}$$

$$\widehat{\mathbf{cov}(\boldsymbol{\zeta}_{-i})} = \widehat{var(\overline{w(\mathbf{x}_i)})} \left(\mathbf{X}_{-i}' \mathbf{X}_{-i} \right)^{-1}\,. \tag{3.2.26}$$

Computing (3.2.22) for each input combination $\mathbf{d}_i$ $(i = 1, \ldots, n)$, the regression metamodel is rejected if

$$\max_i t_{m-1}^{(i)} > t_{m-1;1-\alpha/(2n)}\,, \tag{3.2.27}$$

where the right-hand side follows from *Bonferroni's inequality*; for further details, refer to Kleijnen (2008).

Besides cross validation, we also perform a test on individual regression coefficients, which Myers and Montgomery (2002) suggest to be useful in determining the value of each of the structure of the regression model. We assume that the simulation outputs $\mathbf{w}$ are *normally* distributed; i.e., we assume that the environmental variables $\mathbf{e}$ and the noise ϵ in (3.2.8) are normally distributed. The OLS estimator $\widehat{\boldsymbol{\zeta}}$ in (3.1.5) is then also normally distributed. Consequently, the individual estimated regression parameters $\widehat{\zeta_j}$ may be tested through the following t-statistic:

$$t_0 = \frac{\widehat{\zeta_j}}{\sqrt{\hat{\sigma}_\varepsilon^2 C_{jj}}} \tag{3.2.28}$$

where $\widehat{\zeta_j}$ is the OLS estimate of the coefficient ζ_j and C_{jj} is the j-th diagonal element of the matrix $(\mathbf{X}^T\mathbf{X})^{-1}$. It can be noted that the denominator in Eq. 3.2.28 represents the square root of the j-th element on the main diagonal of the covariance matrix for $\widehat{\zeta}$ given by

$$\mathbf{cov}(\widehat{\boldsymbol{\zeta}}) = (\mathbf{X}'\mathbf{X})^{-1}\sigma_w^2 \,. \tag{3.2.29}$$

with σ_w^2 assumed to be constant and estimated through the MSR defined in (3.2.18). The null hypothesis $H_0 : \zeta_j = 0$ is rejected if

$$|t_0| > t_{1-\alpha/2,n-q} \,. \tag{3.2.30}$$

It is well-known that the t-statistic is not very sensitive to nonnormality; see Kleijnen (1987).
Using the results from the t-statistics, we also compute the confidence intervals for each regression coefficient, derived as follows:

$$\zeta_j \in \left[\widehat{\zeta_j} - t_{1-\alpha/2,n-q}\sqrt{\hat{\sigma}_\varepsilon^2 C_{jj}}, \widehat{\zeta_j} + t_{1-\alpha/2,n-q}\sqrt{\hat{\sigma}_\varepsilon^2 C_{jj}}\right] \tag{3.2.31}$$

Myers and Montgomery (2002) keep only the significant effects in their response model. We agree that when estimating the robust optimum, we should use the *reduced* metamodel, which eliminates all non-significant effects in the full model — except for those non-significant effects that involve factors that have significant higher-order effects; e.g., if the estimated effect $\widehat{\beta_1}$ is not significant but the estimated quadratic effect $\widehat{\beta_{1;1}}$ is, then $\widehat{\beta_1}$ is not set to zero. We point out that the (possibly non-significant) OLS estimator is the Best Linear Unbiased Estimator (BLUE) so we must have good reasons to replace it by zero (also see the 'strong heredity' assumption in Wu and Hamada (2000)). The reduced metamodel could make the optimization process faster, because it may imply a unique optimum, whereas the full metamodel may suggest (say) a saddlepoint. To find the unimportant effects, Myers and Montgomery (2002) use ANalysis Of VAriance (ANOVA). Note that $t_{n-q}^2 = F_{1;n-q}$; the F statistic is used in ANOVA.

Our final goal in robust optimization is to solve a multiobjective optimization problem, minimizing both the estimated mean $\widehat{y}$ and the estimated standard deviation $\widehat{\sigma_y}$ — assuming that (3.2.15) and (3.2.16) are adequate approximations. More specifically, we adopt the ε-constraint method, a classical technique to solve optimization problems with multiple criteria (see Miettinen, 1999): therefore we formulate the problem as a constrained minimization problem, where we minimize the estimated mean $\widehat{y}$, while keeping the estimated standard deviation $\widehat{\sigma_y}$ below a given threshold T. This gives the values of the 'estimated robust decision variables' (say) $\widehat{\mathbf{d}^+}$ and its corresponding mean $\widehat{y}^+$ and standard deviation $\widehat{\sigma_y^+}$. Next, we vary the threshold value T (say) 100 times, which may give a different solution $\widehat{\mathbf{d}^+}$ with its corresponding $\widehat{y}^+$ and $\widehat{\sigma_y^+}$. Then, we collect the 100 pairs $(\widehat{y^+}, \widehat{\sigma_y^+})$ to estimate the Pareto frontier.

Myers and Montgomery (2002) also discuss *constrained optimization*, which minimizes (e.g.) the variance (3.2.13) subject to a constraint on the mean (3.2.12). Often those authors simply superimpose contour plots for the mean and variance, to select an appropriate compromise or 'robust' solution. We shall use Mathematical Programming, which is more general and flexible.

To construct *confidence intervals* for the robust optimum Myers and Montgomery (2002) assume normality, which results in an F statistic. Myers and Montgomery (2002) notice that the analysis becomes complicated when the noise factors do not have constant variances. We shall therefore use *parametric bootstrapping* of the estimated regression parameters that gave $\widehat{y}$ and $\widehat{\sigma_y}$, in order to estimate the variability of the Pareto frontier. By definition, parametric bootstrapping assumes that the distribution of the relevant random variable is known.
More specifically, we sample — via the Monte Carlo method, using pseudo-random numbers —(say) B times from the multivariate — namely q-variate — normal distribution with mean vector and covariance matrix given by (3.1.5) and (3.2.29):

$$\widehat{\boldsymbol{\zeta}}^* \sim N_q(\widehat{\boldsymbol{\zeta}}, (\mathbf{X}'\mathbf{X})^{-1}\widehat{\sigma_w^2}) \tag{3.2.32}$$

where the superscript $*$ is the usual symbol for bootstrapped values. This sampling gives $\widehat{\boldsymbol{\zeta}}_b^*$ $(b = 1, \ldots, B)$. This $\widehat{\boldsymbol{\zeta}}_b^*$ gives $\widehat{y_b^*}$; see (3.2.15) with β_0, $\boldsymbol{\beta}$, $\mathbf{B}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\Delta}$ replaced by their bootstrapped estimates (i.e., estimates computed from the bootstrapped $\widehat{y_b^*}$). It also gives $\widehat{\sigma_{y_b^*}}$; see (3.2.16) where σ_ϵ^2 is replaced by the estimate computed from the bootstrapped parameters. These two bootstrapped regression models $\widehat{y_b^*}$ and $\widehat{\sigma_{y_b^*}}$ are used into the optimization process to obtain the bootstrapped optimal decision vectors $\widehat{\mathbf{d}_b^{+*}}$, which are computed through the `fmincon` function from the MATLAB Optimization Toolbox (see The MathWorks Inc., 2005a). This bootstrap sample gives a bundle of B estimated Pareto frontiers.

Note: Though we focus on estimating the variability of the Pareto curve, we could also have estimated the variability of the solution of the robust optimum problem. So the B bootstrap regression parameters ζ^* gives B values for $\mathbf{d}^+$ and the corresponding y^+ and $s(C)^+$. These B values can be used to derive a CI; see Efron and Tibshirani (1993).

In general, bootstrapping is a simple numerical/computerized method for obtaining the Estimated Density Function (EDF) of a — possibly complicated — statistic for a — possibly non-Gaussian — parent distribution. Examples are the well-known Student statistic for a non-Gaussian parent distribution, and the statistic that is formed by the solution of a Nonlinear Programming problem with Gaussian inputs (as is the case for our study). More details are given by Efron and Tibshirani (1993), Kleijnen (2008), and Kleijnen et al. (2008). We shall illustrate our methodology in the next chapters, through some practical applications.

3.3 Remarks on the assumption of normally distributed simulation outputs

When we fit a regression metamodel, we do assume that the outputs are normally distributed and this holds because the environmental variables $\mathbf{e}$ and the noise ε are supposed to be normally distributed. In general, some *classic assumptions* are often made in regression analysis, briefly recalled in the following: (i) white noise metamodel's residuals, (ii) simulation output having constant variance; (iii) full rank for the matrix of explanatory variables $\mathbf{X}$. Notice that, at this stage, no specific parametric family of density functions is assumed for the output $\mathbf{y}$.

When the classic assumptions hold, then the OLS is the Best Linear Unbiased Estimator (BLUE) of the regression coefficients $\boldsymbol{\beta}$. Mittelhammer (1996) discusses the effects that a violation in each of the classical assumptions has on the properties of the resulting estimator. More specifically, he proves that when the assumption (i) is violated, then the OLS estimator $\widehat{\boldsymbol{\beta}}$ is in general a *biased* estimator of $\boldsymbol{\beta}$, so the BLUE property is lost since $\widehat{\boldsymbol{\beta}}$ is not even unbiased. More important, when the assumption (ii) does not hold, then the OLS estimator $\widehat{\boldsymbol{\beta}}$ is still *unbiased* for $\boldsymbol{\beta}$. As for the BLUE property, however, in the general (and usual) case where the matrix $\boldsymbol{\Phi} = E(\varepsilon\varepsilon')$ is unknown, no linear estimator exists that satisfies the BLUE property. Mittelhammer (1996) also observes that in the very special case where $\boldsymbol{\Phi}$ is known, then a BLUE estimator exists, and it is commonly referred to as Generalized Least Squares (GLS) estimator.

Besides the classic assumptions discussed so far, it is sometimes convenient to further assume that the outputs — and, hence, the residuals ε — follow a Normal (Gaussian) distribution. In fact, this assumption provides additional properties to the estimator of the regression coefficients $\widehat{\boldsymbol{\beta}}$. First, we observe that, under the normality assumption, the classic assumptions necessarily imply that the residuals ε_i's are independent and identically distributed, with zero mean and variance σ^2 (equal to the variance of the Gaussian distribution). Then, the estimate $\widehat{\boldsymbol{\beta}}$ is multivariate normally distributed, whereas it is just asymptotically normally distributed when only the classic assumptions hold.

Due to the properties resulting from the normality assumption, it is important to understand when and to what extent this assumption is realistic. Obviously, the as-

sumption can be supported by some underlying theoretical or physical reasons related to the particular problem under study. When this is not the case, the normal distribution of the outputs is justified by virtue of the Central Limit Theorem (CLT): in particular, it can be argued that the elements in the disturbance vector are themselves defined as the summation of a large number of random variables, each eventually representing neglected explanatory variables which — although affecting the output — are not explicitly included in the regression model due to a lack of data or in order to keep the problem still tractable. There can also be other conditions which make the CLT applicable; they are discussed in Mittelhammer (1996, ch. 5,8).

Of course, there might be cases to which the CLT does not apply; for instance, Kleijnen (2008) considers the estimated quantile $w_{(\lceil 0.90c+0.5 \rceil)}$ as a possible output, and in general he does not expect normality for it — unless the simulation run c is *very* long. On the other hand, referring to inventory problems, Mittelhammer (1996) notices that some outputs of interest can be the costs averaged over the simulated periods or the service percentage calculated as the fraction of demand delivered from on-hand stock per period, so then the final output we are really interested in is represented by the average computed over the planning horizon. He expects these outputs to be normally distributed.

Based on what discuss so far, we can conclude that there can be arguments to support the assumption of normality for the simulation outputs; however, it is preferable to test for acceptability of the normality assumption, by performing appropriate statistical tests.

The Kolmogorov-Smirnov test provides a general procedure for testing whether a random sample is drawn from a specified population distribution (e.g. normal); however, this test can be performed only when the hypothesized distribution function is completely specified, i.e. all its parameters are supposed to be known.
An extension of the Kolmogorov-Smirnov test was proposed by Lilliefors, to test the hypothesis of normality, without specifying the mean and the variance of the normal distribution (see Conover, 1980). Given a random sample $X_1, \ldots, X_n$ of size n, the Lilliefors test consists of the following steps:

1. Estimate the mean μ and the variance σ^2 of the normal distribution, through the computation of the sample mean (3.3.1) and the sample variance (3.3.2):

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i \,, \tag{3.3.1}$$

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left(X_i - \bar{X} \right)^2 \,. \tag{3.3.2}$$

2. Compute the 'standardized' sample values Z_i as follows:

$$Z_i = \frac{X_i - \bar{X}}{s} \quad i = 1, \ldots, n \,. \tag{3.3.3}$$

3. Let $F^{\star}(x)$ denote the standard normal distribution function and $S(x)$ the empirical distribution function of the Z_i's. The Lilliefors test statistic is defined by
$$T_1 = sup_x |F^{\star}(x) - S(x)| \,. \tag{3.3.4}$$

4. Reject the null hypothesis $H_0 : Z \sim \mathcal{N}(\mu, \sigma^2)$ at the approximate level of significance α if T_1 exceeds its $1 - \alpha$ quantile.

In order to test whether the simulation outputs do follow a normal distribution, we apply the Lilliefors test using the implementation provided by the MATLAB Statistics Toolbox, through the function `lillietest`.

Chapter 4

Robust Optimization using Kriging

The purpose of the present chapter is to discuss a novel approach to perform robust simulation-optimization, still interpreting the simulated system under study according to the Taguchian viewpoint, but — differently from what described in Chapter 3 — we will use another metamodelling technique, namely Kriging. An overview of this technique is given in Section 4.1. Then, the proposed approach is discussed in details in Section 4.2.
The experiments and computational results of this methodology will be presented in the next chapters, where it will be applied to the inventory models described in Chapter 2.

4.1 Kriging Metamodelling

In this section we describe the characteristics of the Kriging metamodelling technique, in more detail with respect to what was done for the RSM approach, since this metamodelling technique is expected not to be so well-known as the regression technique is. We also refer to Sacks et al. (1989) and Santner et al. (2003) for a detailed exposition of both theory and implementation of Kriging technique.

A Kriging model can be seen as a combination of a global model plus a localized approximation, as reported in (4.1.1):

$$y(\mathbf{x}) = f(\mathbf{x}) + Z(\mathbf{x}) \,, \tag{4.1.1}$$

where $f(\mathbf{x})$ is a known function of $\mathbf{x}$ as a global model of the original function, and $Z(\mathbf{x})$ is a stochastic process with zero mean and non-zero variance, representing a local deviation from the global model. Usually, the regression model $f(\mathbf{x})$ appearing

in (4.1.1) can be written as

$$f(\mathbf{x}) = \sum_{i=1}^{p} \beta_i \, f_i(\mathbf{x}) \, , \tag{4.1.2}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, p$, are polynomial terms (typically of first or second order) and, in many cases, they are reduced to constants. The coefficients β_i, $i = 1, \ldots, p$, are regression parameters.
The $p + 1$ regression functions can be regarded as components of a vector

$$\mathbf{f}(\mathbf{x}) = [f_0(\mathbf{x}), \ldots, f_p(\mathbf{x})]^T \, . \tag{4.1.3}$$

Suppose the design sites are $(\mathbf{x}_1, \ldots, \mathbf{x}_{N_s})$, where $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \ldots, N_s$; then, we can compute the matrix $\mathbf{F}$ by evaluating the vector $\mathbf{f}(\mathbf{x})$ at the design sites, thus obtaining:

$$\mathbf{F} = \begin{bmatrix} f^T(\mathbf{x}_1) \\ \vdots \\ f^T(\mathbf{x}_{N_s}) \end{bmatrix} = \begin{bmatrix} f_0(\mathbf{x}_1), & \ldots, & f_p(\mathbf{x}_{N_s}) \\ \vdots & & \vdots \\ f_0(\mathbf{x}_1), & \ldots, & f_p(\mathbf{x}_{N_s}) \end{bmatrix} . \tag{4.1.4}$$

The covariance of $Z(\mathbf{x})$ is expressed by (4.1.5):

$$Cov[Z(\mathbf{x}_j), Z(\mathbf{x}_k)] = \sigma^2 \mathbf{R}(\mathbf{x}_j, \mathbf{x}_k), \qquad j, k = 1, \ldots, N_s \, , \tag{4.1.5}$$

where σ^2 is the so-called process variance and $\mathbf{R}$ is the correlation matrix, whose elements are given by $R_{jk} = R_\theta(\mathbf{x}_j, \mathbf{x}_k)$, representing the correlation function between any two of the N_s samples $\mathbf{x}_j$ and $\mathbf{x}_k$, with unknown parameters θ. $\mathbf{R}$ is a symmetric matrix of dimension $N_s \times N_s$, with diagonal elements equal to 1.
The form of the correlation function $R_\theta(\mathbf{x}_j, \mathbf{x}_k)$ can be chosen among a variety of functions proposed in the literature, although the exponential family is used most frequently; in this case the correlation function can be expressed in a parametric form as in (4.1.6):

$$R_{\theta,p}(\mathbf{x}_j, \mathbf{x}_k) = \prod_{i=1}^{n} exp\left(-\theta_i |x_{ji} - x_{ki}|^{p_i}\right) \, , \tag{4.1.6}$$

where n is the dimension of the input variable. When $p_i = 2$, then (4.1.6) is called the Gaussian correlation function.

The Kriging predictor can be written as a linear combination of the observed responses, i.e. in the following linear form

$$\widehat{y}(\mathbf{x}) = c^T(\mathbf{x}) y_s \, , \tag{4.1.7}$$

where y_s is the vector of the response function evaluated at the N_s design sites, $y_s = [y(\mathbf{x}_1), \ldots, y(\mathbf{x}_{N_s})]^T$. The weights $c(x)$ are obtained by minimizing the Mean Squared Error (MSE), which is given by

$$MSE[\widehat{y}(\mathbf{x})] = E\left[\left(c^T(\mathbf{x}) y_s - y(\mathbf{x})\right)^2\right] . \tag{4.1.8}$$

In order to keep the predictor unbiased, the following constraint has to be satisfied:

$$F^T c(\mathbf{x}) = f(\mathbf{x})\,. \tag{4.1.9}$$

Based on what discussed so far, it can be proven that the MSE in (4.1.8) can be rewritten as

$$MSE[\widehat{y}(\mathbf{x})] = \sigma^2[1 + c^T(\mathbf{x})Rc(\mathbf{x}) - 2\mathbf{c}^T(\mathbf{x})\mathbf{r}(\mathbf{x})]\,, \tag{4.1.10}$$

where $\mathbf{r}(\mathbf{x}) = [R(\mathbf{x}_1, \mathbf{x}), \ldots, R(\mathbf{x}_{N_s}, \mathbf{x})]^T$ is the vector of the correlations between $Z(\mathbf{x}_i)$ and $Z(\mathbf{x})$.
To minimize the MSE in (4.1.10) with respect to $\mathbf{c}(\mathbf{x})$ under the constraint (4.1.9), we introduce the Lagrangian function, defined as

$$\mathcal{L}(\mathbf{c}(\mathbf{x}), \lambda(\mathbf{x})) = \sigma^2[1+\mathbf{c}^T(\mathbf{x})R\mathbf{c}(\mathbf{x})-2\mathbf{c}^T(\mathbf{x})\mathbf{r}(\mathbf{x})]-\lambda^T(\mathbf{x})\left(F^T\mathbf{c}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\right)\,. \tag{4.1.11}$$

The gradient of 4.1.11 with respect to c is given by

$$\mathcal{L}'(\mathbf{c}(\mathbf{x}), \lambda(\mathbf{x})) = 2\sigma^2[R\mathbf{c}(\mathbf{x}) - \mathbf{r}(\mathbf{x})] - F\lambda\,. \tag{4.1.12}$$

The first order necessary conditions for optimality imply the following system of equations:

$$\begin{bmatrix} R & -1/(2\sigma^2)F \\ F^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c}(\mathbf{x}) \\ \lambda(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \mathbf{r}(\mathbf{x}) \\ \mathbf{f}(\mathbf{x}) \end{bmatrix}\,. \tag{4.1.13}$$

The solution of the system (4.1.13) is

$$\lambda(\mathbf{x}) = 2\sigma^2\left(F^T R^{-1} F\right)^{-1}(\mathbf{f}(\mathbf{x}) - F^T R^{-1}\mathbf{r}(\mathbf{x})\big) \tag{4.1.14}$$

$$\mathbf{c}(\mathbf{x}) = R^{-1}\left(\mathbf{r}(\mathbf{x}) + \frac{1}{2\sigma^2}F\lambda(x)\right) \tag{4.1.15}$$

which yields the following equation for the Kriging predictor:

$$\begin{aligned} \widehat{y(\mathbf{x})} &= \mathbf{c}^T(\mathbf{x})y_s = \\ &= \mathbf{r}^T R^{-1} y_s + (\mathbf{f} - F^T R^{-1}\mathbf{r})^T (F^T R^{-1} R)^{-1} F^T R^{-1} y_s = \\ &= \mathbf{r}^T R^{-1} y_s + (\mathbf{f}^T - \mathbf{r}^T R^{-1} F)(F^T R^{-1} R)^{-1} F^T R^{-1} y_s = \\ &= \mathbf{r}^T R^{-1}(y_s - F\widehat{\beta}) + \mathbf{f}^T\widehat{\beta}\,, \end{aligned} \tag{4.1.16}$$

where

$$\widehat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} y_s \tag{4.1.17}$$

is the generalized least-squares (GLS) estimate of β in 4.1.2.

The construction of the Kriging model is based on an estimation of the unknown correlation parameters θ. Assuming the stochastic process $Z(x)$ to be Gaussian, a likelihood function can be minimized using numerical optimization techniques to determine an estimate $\widehat{\theta}$ used to fit the model (Sacks et al., 1989). The likelihood function depends on the coefficients β in the regression model, the process variance

σ^2 and the correlation parameters θ. Given the correlation parameters, the Maximum Likelihood Estimation (MLE) of β is the GLS estimator given by (4.1.17), and the MLE of σ^2 is given by

$$\widehat{\sigma}^2 = \frac{1}{N_s}(y_s - F\widehat{\beta})^T R^{-1}(y_s - F\widehat{\beta})\,. \tag{4.1.18}$$

Therefore, the problem becomes the following:

$$\min_{\theta} (\det\, R)^{1/N_s}\widehat{\sigma}^2\,; \tag{4.1.19}$$

this is a global optimization problem, which requires a heuristic procedure to be solved.

In the proposed framework, Kriging models are constructed using algorithms and functions of the MATLAB DACE Toolbox (see Lophaven et al., 2002). In order to build a starting set of elements at the basis of the approximation model, several data sampling methods have been suggested in the field of design of experiments. The DACE toolbox offers a limited support to this preliminary activity, including the rectangular grid – obtained considering all possible combinations of levels for the factors involved in the experiments – and the (uniform) Latin Hypercube Sampling, whose technique has already been discussed in Chapter 3; besides these, we implement some other DoE techniques, to let the user choose the design expected to perform better or being more promising for the problem to be analyzed. Moreover, the DACE toolbox includes some heuristics to solve the problem in (4.1.19); in particular, for the optimization process to start, a starting value $\theta = \theta_0$ has to be specified; optionally, a lower bound θ_{lb} and an upper bound θ_{ub} on the values of its components can be provided as well.

In the experiments we performed — whose results will be discussed in the following chapters — we adopt the so-called *ordinary Kriging*, according to the implementation provided by the DACE toolbox: this model — still based on Eq. 4.1.1 — selects $p = 0$ in Eq. 4.1.3, fixing also $f_0 \equiv 1$; therefore the ordinary Kriging model can be rewritten as follows:

$$y(\mathbf{x}) = \mu + Z(\mathbf{x})\,. \tag{4.1.20}$$

Notice that the DACE toolbox also provides the more general formulation given in Eqs. 4.1.1-4.1.3.

4.2 Robust Optimization using Kriging

In this section we describe how to achieve robustness in simulation-optimization, using Kriging metamodels. The basic scheme of the proposed approach resembles the one described in Chapter 3, but it introduces some differences, which will be justified in the following discussion.

We discuss the following two approaches based on Kriging:

1. Inspired by Dellino et al. (2008a), we fit *two* Kriging metamodels, namely one model for the mean and one for the standard deviation — both estimated from the *simulation*'s I/O data.

2. Inspired by Lee and Park (2006), we fit a *single* Kriging metamodel to a relatively small number (say) n of combinations of the control variables $\mathbf{d}$ and the noise variables $\mathbf{e}$. Next, we use this metamodel to compute the *Kriging predictions* for the simulation output w for $N \gg n$ combinations of $\mathbf{d}$ and $\mathbf{e}$.

The general procedure is outlined in Fig. 4.1, making distinctions between the two approaches we are going to discuss.

Robust Optimization combining Taguchi & Kriging

Input: design ($\mathbf{D}$, $n_{\mathbf{d}} \times n_{\mathbf{e}}$) in the control-by-noise space $(\mathbf{d}, \mathbf{e})$;
probability distribution of noise factors;

begin
for each design point (d_j, e_j) in $\mathbf{D}$, $j = 1, \ldots, n_{\mathbf{d}} \times n_{\mathbf{e}}$
1a. Run the simulation model
1b. Compute $w_j = f(d_j, e_j)$
switch sel_approach
case Approach 1
2. Fit one Kriging metamodel for the mean of the response, $\widehat{y}$, and one for its standard deviation, $\widehat{\sigma_y}$
3. Validate the metamodels, using leave-one-out cross validation
case Approach 2
2. Fit a Kriging metamodel for the main output function, y
3. Validate the metamodel, using leave-one-out cross validation
4. Generate a new design in the control-by-noise space
5. Fit one Kriging metamodel for the mean of the predicted response, $\widehat{\overline{y}}$, and one for its standard deviation, $\widehat{s(y)}$
6. Validate the metamodels, using leave-one-out cross validation
7. Estimate the Pareto frontier, solving constrained optimization problems
8. Apply distribution-free bootstrap on the output data B times
9. Derive confidence intervals for the Pareto frontier, using the bootstrapped data sets
end

Figure 4.1: Robust Optimization approaches through Kriging.

In the first approach, we select the input combinations for the simulation model through a *crossed* (combined) design for the control and noise factors (as is usual in Taguchian design); i.e., we combine the (say) n_d combinations of the decision variables with the n_e combinations of the environmental variables. These n_d combinations are space-filling; the n_e combinations are sampled from their input distribution; this

sampling may use LHS. Simulating these $n_d \times n_e$ combinations gives the outputs w_{ij} with $i = 1, \ldots, n_d$ and $j = 1, \ldots, n_e$. These I/O data enable the computation of the following unbiased estimators of the conditional means and variances:

$$\overline{w_i} = \frac{1}{n_e} \sum_{j=1}^{n_e} w_{ij} \ (i = 1, \ldots, n_d)\,, \tag{4.2.1}$$

$$s_i^2 = \frac{1}{n_e - 1} \sum_{j=1}^{n_e} (w_{ij} - \overline{w_i})^2 \ \ (i = 1, \ldots, n_d)\,. \tag{4.2.2}$$

The latter estimator is not unbiased, because $E(\sqrt{s^2}) = E(s) \neq \sqrt{E(s^2)} = \sqrt{\sigma^2} = \sigma$; however, we ignore this bias.
Note that — although statisticians had criticized it in the Taguchian approach — we do use crossed design, because we need to estimate the mean and the variance of the output function and in order to do this we need multiple observations for the same combination of decision factors.

In the second approach, we select a (relatively small) number of input combinations for the simulation model through a space-filling design for the $n_d + n_e$ input factors. This implies that, at this stage, we do not distinguish between control and noise factors yet; moreover, the noise factors are not yet sampled from their distribution. It is worth noticing that the space-filling design avoids extrapolation in the next step, i.e. when using the Kriging metamodel to obtain predictions. For the larger design we select a space-filling design for the control factors, but a LHS design for the noise factors accounting for their distribution. For this large I/O set we compute the Kriging predictors for the conditional means and standard deviations; i.e., in the right-hand sides of (4.2.1) and (4.2.2) we replace n_e and n_d by N_e and N_d, (the large-sample analogues of the small-sample n_e and n_d) and w by $\widehat{w}$ where the 'hat' denotes Kriging prediction.

Validation of the Kriging metamodels is performed through cross-validation, similarly to what discussed in Chapter 3. Notice that — to avoid extrapolation — we apply cross-validation to a restricted set of input combinations $\mathbf{d}_i$, namely excluding those input combinations for which at least one of the input factors $(d_1, ..., d_{n_\mathbf{d}})$ contains an extreme value (i.e., the highest or lowest value for that input factor, across the n points already simulated). We refer to the remaining points as the cross-validation set, whose cardinality we denote by n_{cv}.
A remark should be added, when the underlying simulation model is stochastic, implying that replications are required. As pointed out in Chapter 3, replicated simulation outputs make it possible to compute the Studentized prediction error for each of the input combinations that have been left out during the cross validation procedure. However, an additional step is required, because we need to estimate the variance of the Kriging prediction at the left-out input combination $\mathbf{d}_i$. den Hertog et al. (2006) proved that the estimator of the Kriging predictor commonly used in the literature is biased; therefore, they propose an unbiased estimator, based on distribution-free bootstrapping.

First, we give a general description of this method; then, we will discuss the estimator for Kriging variance based on bootstrap. Suppose we have n independent observations $x_1, \ldots, x_n$, for convenience denoted by the vector $\mathbf{x} = (x_1, \ldots, x_n)$, from which we compute a statistic of interest $s(\mathbf{x})$. A bootstrap sample $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$ (the superscript $*$ is the usual symbol for bootstrapped values) is obtained by randomly sampling n times, with replacement, from the original population of data points $(x_1, \ldots, x_n)$. The bootstrap data set $(x_1^*, \ldots, x_n^*)$ consists of members of the original data set $(x_1, \ldots, x_n)$, some appearing zero times, some appearing once, some appearing twice, etc. This is repeated (say) B times; B is called the bootstrap sample size. Which value should be chosen for B is still a matter for debate. Efron and Tibshirani (1993) discusses the topic and suggests some rules of thumb, pointing out that very seldom more than $B = 200$ bootstrap samples are needed. Corresponding to each bootstrap sample is a bootstrap replication of s, namely $s(\mathbf{x}_b^*)$, which is the result of applying the same function $s(\cdot)$ to $\mathbf{x}_b^*$ as was applied to $\mathbf{x}$.
The implementation of the bootstrap algorithm is straightforward: we randomly select integers $i_1, i_2, \ldots, i_n$, each of which equals any value between 1 and n with probability $1/n$. The bootstrap sample consists of the corresponding elements in $\mathbf{x}$,

$$x_1^* = x_{i_1}, x_2^* = x_{i_2}, \ldots, x_n^* = x_{i_n} .$$

The bootstrap algorithm works by drawing B independent bootstrap samples, each time evaluating the corresponding bootstrap replications.

The method proposed by den Hertog et al. (2006) to compute an unbiased estimator for Kriging variance consists in the following steps:

1. Sample (with replacement) m numbers from the uniform distribution with constant probability $1/m$ for the m integers $\{1, 2, ..., m\}$). This yields a set of m replicate numbers $i_1, i_2, \ldots, i_m$.

2. For each input combination $\mathbf{d}_z$ in the cross-validation set, include the outputs $w_{i_1}(\mathbf{d}_z)$ to $w_{i_m}(\mathbf{d}_z)$ in the bootstrapped data set.

3. Compute the bootstrap output $\overline{w^*(\mathbf{d}_z)}$ averaged over replications, for all combinations d_z in the cross-validation set:

$$\overline{w^*(\mathbf{d}_z)} = \frac{1}{m} \sum_{r=1}^{m} w_{i_r}(\mathbf{d}_z) . \tag{4.2.3}$$

4. Calculate the bootstrapped estimated optimal Kriging weights λ^* and the corresponding bootstrapped Kriging predictor $y^*(\mathbf{d}_i)$ using the $n-1$ bootstrapped averages $\overline{w^*(\mathbf{d}_z)}$.

To decrease the sampling effects of bootstrapping, we repeat the whole procedure B times (in our experiments, we select $B = 200$), yielding estimated optimal Kriging weights λ_b^* and bootstrapped Kriging predictors $y_b^*(\mathbf{d}_i)$ with $b = 1, \ldots, B$. These

B values enable estimation of the variance of the Kriging predictor, analogously to 3.2.24:

$$\widehat{var(y^*(\mathbf{d}_i))} = \frac{1}{B-1}\sum_{b=1}^{B}\left(y_b^*(\mathbf{d}_i) - \overline{y^*(\mathbf{d}_i)}\right)^2 , \tag{4.2.4}$$

where

$$\overline{y^*(\mathbf{d}_i)} = \frac{1}{B}\sum_{b=1}^{B} y_b^*(\mathbf{d}_i)\,. \tag{4.2.5}$$

Then we compute the Studentized prediction error as follows:

$$t_{m-1}^{(i)} = \frac{\overline{w(\mathbf{d}_i)} - \widehat{y_{-i}}(\mathbf{d}_i)}{\sqrt{\widehat{var(\overline{w(\mathbf{d}_i)})} + \widehat{var(y^*(\mathbf{d}_i))}}}\,, \tag{4.2.6}$$

where $\overline{w(\mathbf{d}_i)}$ is given by (3.2.21), $\widehat{y_{-i}}$ is the Kriging prediction at the point left-out, $\mathbf{d}_i$, and

$$\widehat{var(\overline{w(\mathbf{d}_i)})} = \frac{\widehat{var(w(\mathbf{d}_i))}}{m} \tag{4.2.7}$$

$$\widehat{var(w(\mathbf{d}_i))} = \frac{1}{m-1}\sum_{r=1}^{m}\left(w_r(\mathbf{d}_i) - \overline{w(\mathbf{d}_i)}\right) . \tag{4.2.8}$$

Accordingly to what discussed in Section 3.2, we reject the metamodel if

$$\max_i t_{m-1}^{(i)} > t_{m-1;1-\alpha/(2n_{cv})}\,. \tag{4.2.9}$$

Both approaches use the Kriging metamodels for the estimated mean and standard deviation to find the robust optimum that minimizes the mean while satisfying a constraint on the standard deviation. Varying the value of the right-hand side for that constraint gives an estimation of the Pareto frontier.

Our goal is then to analyze the variability of this Pareto frontier. Whereas Dellino et al. (2008a) apply parametric bootstrapping, we apply *nonparametric* or *distribution-free bootstrapping*, which results in a less restrictive approach, since it does not require any particular assumption on the underlying distribution for the data to be bootstrapped.
Notice that bootstrapping assumes that the 'original' observations w_{ij} are Identically and Independently Distributed (IID); see Efron and Tibshirani (1993). LHS, however, implies that in each of the n_e intervals only a single combination of the environmental inputs is sampled. After some experiments to compare results based on crude sampling versus LHS, we decided to ignore this dependence in our bootstrap.

In order to apply nonparametric bootstrap to the output data w_{ij} we should pay attention to select the values of the output function — for each combination of the decision factors — according to the same vector of indices; if not, we will introduce

too much variability in the bootstrapped data, since we are mixing the influence of different environments on the output of each input combination. Therefore, we resample — with replacement — the n_e original simulation outputs $w_{:j}$, which gives the n_e bootstrapped observations $w^*_{:j}$; note that this operation is performed column-wise, i.e. for all i at a time, instead of repeating it for each i ($i = 1, \ldots, n_d$). Analogous to (4.2.1) and (4.2.2) we compute the n_d bootstrapped averages and variances:

$$\overline{w_i}^* = \frac{1}{n_e} \sum_{j=1}^{n_e} w^*_{ij} \ (i = 1, \ldots, n_d)\,, \tag{4.2.10}$$

$$s^{2*}_i = \frac{1}{n_e - 1} \sum_{j=1}^{n_e} \left(w^*_{ij} - \overline{w_i}^* \right)^2 \ (i = 1, \ldots, n_d\,)\,. \tag{4.2.11}$$

To reduce the sampling error in this resampling, we repeat this sampling B times; this sample size gives the bootstrapped averages and variances $\overline{w_{i;b}}^*$ and $s^{2*}_{i;b}$ ($b = 1, \ldots, B$); see (4.2.10) and (4.2.11). To these bootstrapped output data we apply Kriging.

In order to estimate the variability of the original (i.e., non-bootstrapped) Pareto frontier through the bootstrapped Kriging metamodels, we proceed as follows:

- Each point of the Pareto frontier $(\widehat{\overline{y}}_{opt}, \widehat{s(y)}_{opt})$ is obtained by an optimal vector of decision variables $\mathbf{d}_{opt}$.
- Due to the uncertainty introduced by the environmental factors — which we account for through the bootstrap sampling — it might happen that the optimal combination for the decision variables does not generate the expected optimal output values.
- To have an idea of the variability in the output, we predict the mean and standard deviation of the output at the optimal input $\mathbf{d}_{opt}$, based on the boostrapped Kriging metamodels.
- This computation can be applied to all the points on the Pareto frontier.

Part II

Computational Experiments

Chapter 5

The Economic Order Quantity Model

The Economic Order Quantity (EOQ) model is the first example we use to illustrate and test our methodology. Although its simplicity — and maybe even thanks to that — it helps in describing each step of the heuristic procedure, and the computational results are easy to understand (because there are no complicated dependence between the factors involved). The main characteristics of the model and the numerical example we will consider have been already discussed in Chapter 2, so we will present directly the results of our experiments.

The chapter is structured as follows: Section 5.1 describes some preliminary experiments, aiming at studying the sensitivity of the classic EOQ model to small variations in the parameters. Then, Section 5.2 presents the results of the classic optimization on the EOQ model, using regression metamodels: at this stage, no uncertainty on the parameters is assumed yet. Afterwards, Section 5.3 discusses the results of the robust optimization of the EOQ model based on RSM, when demand rate is affected by uncertainty: more specifically, we apply the procedures described in Chapter 3. Finally, Sections 5.4 and 5.5 detail the experiments performed on the EOQ model according to the methods described in Chapter 4, starting from the classic optimization and then moving to the robust formulation.

5.1 Sensitivity Analysis

Before dealing with the core problem of this work, i.e. robust optimization w.r.t. to parameters uncertainty, it might be interesting to perform some preliminary experiments, with the purpose of studying how possible deviations in the parameters from their nominal values (i.e., those values chosen assuming no uncertainty) let the optimal order quantity move from the nominal optimum, i.e. the optimal solution corresponding to nominal parameters values. Besides that, we conduct an empirical

study by means of a numerical example, to verify what stated analytically.

Starting from the EOQ formula in Eq. 2.1.5, we consider the influence of varying the unit holding cost h, computing the partial derivative of Q^* w.r.t. h:

$$\frac{\Delta Q_o}{\Delta h} \simeq \frac{\partial Q_o}{\partial h} = \frac{\partial}{\partial h}\left[\sqrt{\frac{2aK}{h}}\right] = \sqrt{2aK}\,\frac{\partial}{\partial h}\left(\frac{1}{\sqrt{h}}\right) = -\frac{1}{2h}\sqrt{\frac{2aK}{h}} \Rightarrow$$

$$\Rightarrow \frac{\Delta Q_o}{\Delta h} \simeq -\frac{1}{2\overline{h}}\,\overline{Q}_o \Rightarrow \Delta Q_o \simeq -\frac{1}{2\overline{h}}\,\overline{Q}_o\,\Delta h \tag{5.1.1}$$

where $\overline{Q}_o$ denotes the optimal order quantity value achieved at the nominal holding cost $\overline{h}$.

Considering the example introduced in Chapter 2, Fig. 5.1 shows the curve of the total cost w.r.t. the batch size; it also depicts the curves described by $y = aK/Q$ and $y = \frac{1}{2}hQ$: they represent the two cost components (in particular, the ordering costs and the inventory carrying costs) depending on the order quantity Q, which determine the asymptotic behaviour of the curve $C(Q)$.

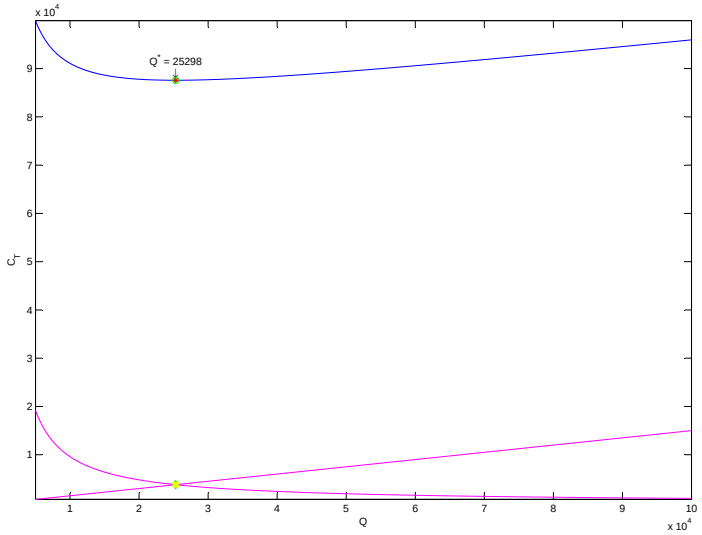


Figure 5.1: Total cost C versus order quantity Q.

Eq. 5.1.1 implies that a positive variation of the unit holding cost causes a decreasing in the optimal order quantity; viceversa, if we diminish the holding cost coefficient, the optimal solution moves to the right, increasing its value. This is evident from Fig. 5.2, which shows how the total cost curve changes due to a variation of the unit holding cost, in the range $[-1\%, +1\%]$ w.r.t. a nominal value $\overline{h}$. Analyzing the results

obtained, it can be noted that a variation of $\Delta h = \pm 1\%$ implies a variation of the optimal order quantity $\Delta Q_o \simeq \mp 0.5\%$.

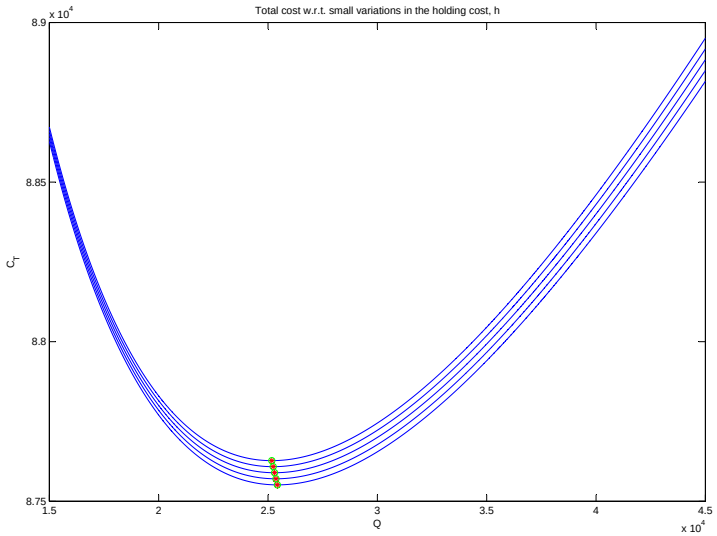


Figure 5.2: Total cost C versus order quantity Q, when varying the holding cost h.

Comparing Fig. 5.2 with Fig. 5.1, it can be noticed that the way the curvature changes is due to the increasing slope of the line describing the inventory carrying costs.

The sensitivity of the optimal order quantity w.r.t. fluctuations in the demand rate can be studied in a similar way; so, computing the partial derivative of Q_o w.r.t. a, we obtain:

$$\begin{aligned} &\frac{\Delta Q_o}{\Delta a} \simeq \frac{\partial Q_o}{\partial a} = \frac{\partial}{\partial a}\left[\sqrt{\frac{2aK}{h}}\right] = \sqrt{\frac{2K}{h}}\,\frac{\partial}{\partial a}\left(\sqrt{a}\right) = \sqrt{\frac{K}{2ah}} \Rightarrow \\ &\Rightarrow \frac{\Delta Q_o}{\Delta h} \simeq \frac{1}{2\overline{a}}\,\overline{Q}_o \Rightarrow \Delta Q_o \simeq -\frac{1}{2\overline{a}}\,\overline{Q}_o\,\Delta a \end{aligned} \tag{5.1.2}$$

As it is clear from Eq. 5.1.2, a slight positive increase in the demand rate causes the optimal order quantity to increase as well; viceversa, if the demand rate decreases, the optimal solution moves to the left, decreasing its value. The behaviour described so far is depicted in Fig. 5.3, where the demand rate varies in the range $[-1\%, +1\%]$ w.r.t. a nominal value $\overline{a}$. In this case, a variation in the demand rate $\Delta a = \pm 1\%$

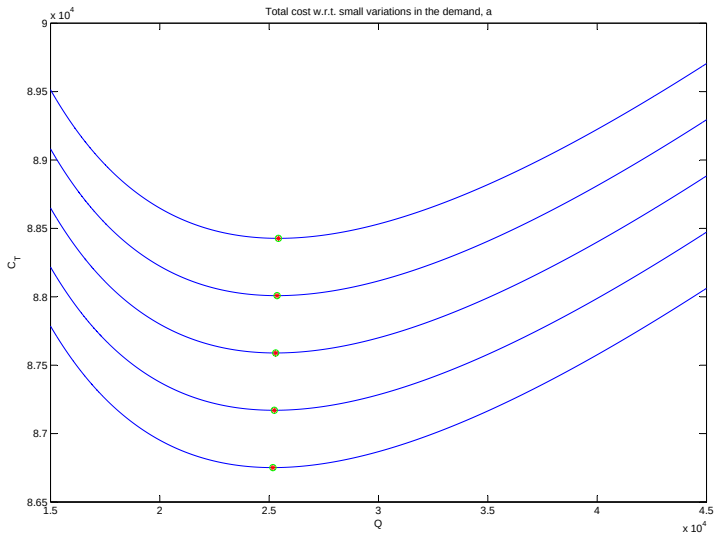


Figure 5.3: Total cost C versus order quantity Q, when varying the demand rate a.

causes the minimum cost solution to move from the nominal value of a quantity $\Delta Q_o \simeq \pm 0.5\%$.

Comparing Fig. 5.3 with Fig. 5.1, it can be noticed that the way the curvature changes is due to the increasing distance from the origin of the vertex of the hyperbola describing the ordering costs, which asymptotically delimits the total cost curve.

It is also likely that both the demand rate and the holding cost vary at the same time; in that case, in order to examine the way the total cost curve changes due to contemporary variations of the two parameters, we should combine the results that have been described before, when one single parameter varied at a time. The resulting behaviour is shown in Fig. 5.4. Moving from what observed before, it is noticeable that a variation of both the parameters in the same direction (i.e. $\Delta h = \Delta a = \pm 1\%$) implies no changes in the value of the optimal order quantity: this is due to the fact that the two parameters produce equal though opposite effects on the position of the optimal solution, thus cancelling each other.

Comparing Fig. 5.4 with the previous Figs. 5.2-5.3, it is clear that the latter are easily obtainable from the former, by fixing only one of the two parameters.

Finally, to complete the sensitivity analysis on the basic EOQ model w.r.t. its parameters, it should be mentioned what happens when the unit cost c or the setup cost K are allowed to vary. When c moves from its nominal value, the optimal order quantity Q_o keeps its nominal value, since there is no dependence on this parameter;

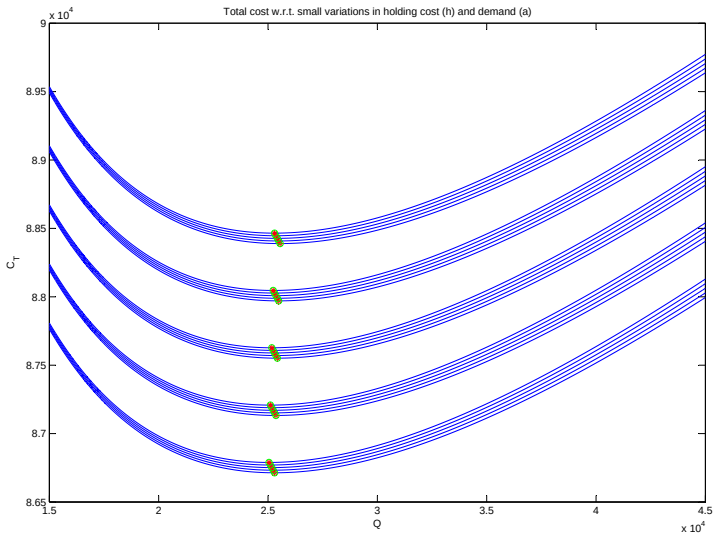


Figure 5.4: Total cost C versus order quantity Q, when varying both the demand rate a and the holding cost h.

however, the optimal total cost C_o changes its value, according to Eq. 2.1.6. The effect of varying c in the range $[-1\%, +1\%]$ w.r.t. its nominal value is shown in Fig. 5.5; notice that the curve of the total cost keeps the same curvature and it just shifts along the ordinate axis.

Variations in the setup cost K have almost the same impact as those affecting the demand rate a, since the functional dependence of Q_o on each of these parameters is similar. Fig. 5.6 shows how the shape of the curve of the total cost changes when the setup cost varies in the range $[-1\%, +1\%]$ w.r.t. its nominal value; even though the curves are quite different from those depicted in Fig. 5.3, we can observe that — in both cases — an increasing (respectively, decreasing) in the value of the parameter allowed to vary determines an increasing (respectively, decreasing) in the value of the optimal order quantity.

5.2 Simulation Optimization of the classic EOQ using RSM

In this section we apply the heuristic procedure described in Chapter 3 to the simulation optimization of the classic EOQ inventory model, still referring to the example

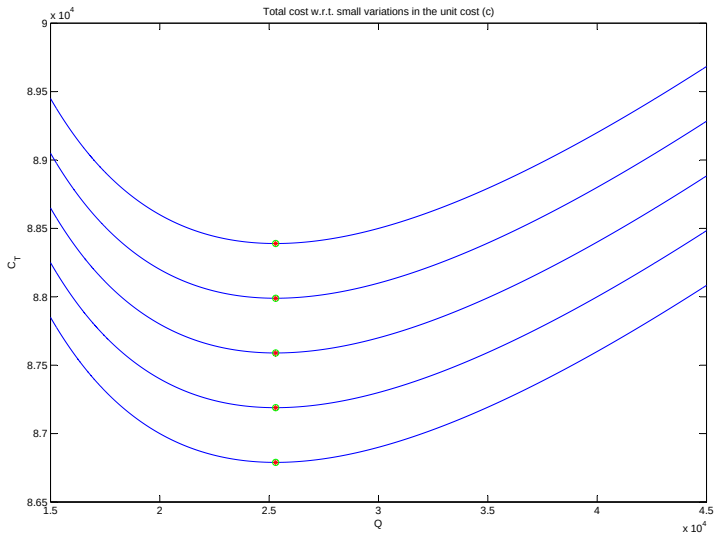


Figure 5.5: Total cost C versus order quantity Q, when varying the ordering cost c.

described in Section 2.1.4.

As it has been already pointed out, the EOQ simulation is deterministic. Because all cycles are identical, we simulate a single cycle only. We start this cycle with an inventory level of Q units. Based on Eqs. 2.1.5-2.1.6 the true optimal input is $Q_o = 25298$ and the corresponding output is $C_o = 87589$; of course, this optimum remains unknown to our procedure, and we use it only to guide our design of the simulation experiment and to verify its results.

The steps of our simulation experiment have been described in Section 3.1.1, where we also pointed out that our experiments refer to a *simplified* RSM; having Fig. 3.1 in mind, we detail the algorithm setting in the following.

1. *Design*: We assume that in practice the analysts have some knowledge about the location of the relevant experimental area. To select the experimental area, we therefore start with the interval $[0.5\,Q_o, 1.5\,Q_o]$. This selection, however, would imply that the midpoint coincides with the true optimum input ($Q_o = 25298$) — which rarely occurs in practice. We therefore shift the interval a little bit (namely, by less than 5000 units) to the right so that it is centered at the 'round' value $\overline{Q} = 30000$. Furthermore, we pick five equally spaced points (a Central Composite Design or CCD would also have five points, albeit not equally spaced; see Myers and Montgomery (2002) and Table 5.10), including the extreme points, $0.5 \times 30000 = 15000$ and $1.5 \times 30000 = 45000$; see row 1 of

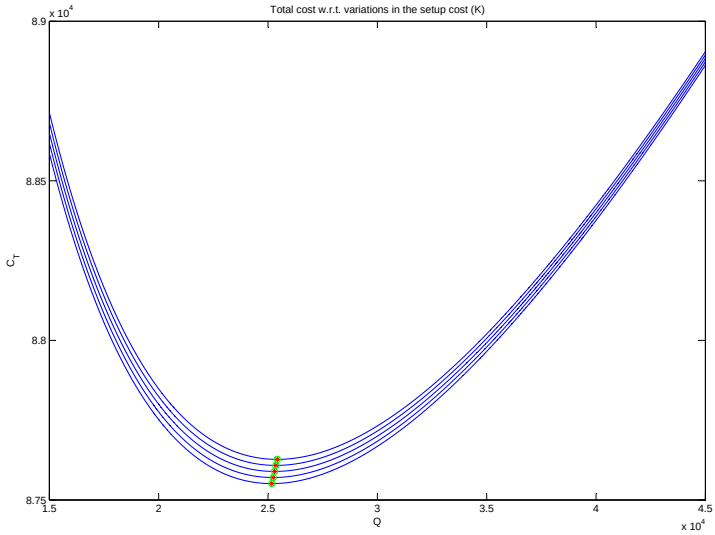


Figure 5.6: Total cost C versus order quantity Q, when varying the setup cost K.

Table 5.1: I/O data of EOQ simulation

Q	15000	22500	30000	37500	45000
C	88650	87641.66	87700	88185	88883.34

Table 5.1 below. The input parameters are fixed to their nominal values.

2. *Simulation Model*: We program the simulation model in Arena (Kelton et al., 2007). Next we run this simulation, and obtain $C(Q_i) = C_i$, which denotes the cost corresponding with input value i ($i = 1, \ldots, 5$) selected in step 1; see the Input/Output (I/O) combinations (Q_i, C_i) displayed in Table 5.1.

3. *Regression Metamodel*: Based on the I/O data from steps 1-2, we estimate a second-order polynomial regression metamodel, using Ordinary Least Squares (OLS). We could use either the original or the coded factor values; see (3.2.10). However, we focus on the estimated effects of the coded decision variable, because these effects (say) $\widehat{\boldsymbol{\beta}}$ show their relative importance; moreover, their numerical accuracy is better: the condition number for $\mathbf{X}$ is 3.08, whereas it is $1.07 \cdot 10^{10}$ when using the original Q. This $\widehat{\boldsymbol{\beta}}$ is displayed in the row with $i = 0$ (zero I/O data eliminated) in Table 5.2. (We also compute the estimated effects of the original variable; e.g., the estimated quadratic effect is then of order 10^{-6},

so it seems unimportant; however, this coefficient is multiplied by Q^2, which is of order 10^8, so their joint effect is of order 10^2.)
Fig. 5.7 shows the regression metamodel for the EOQ, comparing it with the analytical I/O function. The optimal solution for each of the two curves is also plotted.

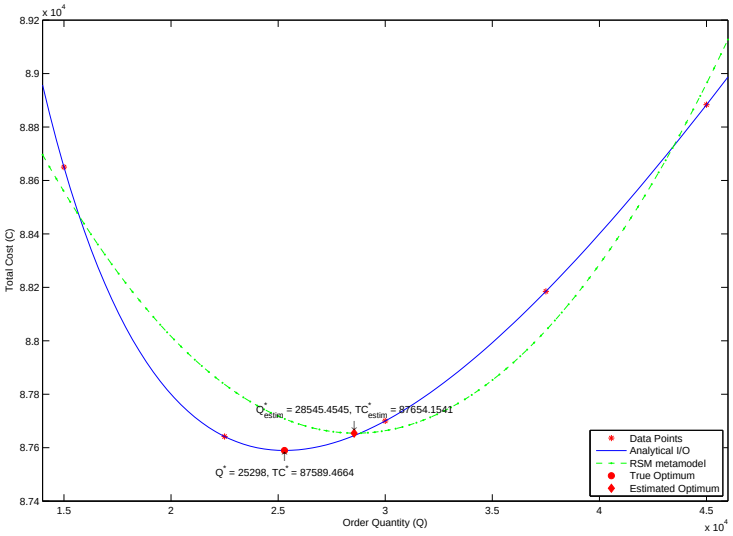


Figure 5.7: EOQ regression metamodel.

4. *Metamodel cross-validation*: The remaining rows of Table 5.2 display the re-estimated regression parameters following from (3.2.19), and the re-estimated regression prediction following from (3.2.20). This table also presents the relative prediction errors $\widehat{y_{(-i)}}/C_i$, which supplement the scatterplot in Figure 5.8. Notice that this is not the only way to evaluate the metamodel accuracy: sometimes, it could be preferable to evaluate the residuals, $e_i = y_i - \widehat{y}_i$, $i = 1, \ldots, n$, eventually working with scaled residuals, which often convey more information than the ordinary residuals do (see Myers and Montgomery, 2002). The estimated regression coefficients in different rows remain more or less the same. Anyhow, we decide to accept the regression metamodel because we think it is an approximation that is adequate enough for our goal, which is the illustration of robust optimization through the EOQ model (for the roles of different goals in simulation see Kleijnen and Sargent (2000)).

   Note: Table 5.2 implies that the estimated main effect is not significantly different from zero, whereas the quadratic effect is; see the t statistic in (3.2.30). As we discussed before, we do not replace the estimated main effect by zero.

Table 5.2: Cross-validation of EOQ regression metamodel

i	$\widehat{\beta_{0(-i)}}$	$\widehat{\beta_{1(-i)}}$	$\widehat{\beta_{1;1(-i)}}$	$\widehat{y_{(-i)}}$	$\widehat{y_{(-i)}}/C_i$
0	87663.4257	202.004	1097.15		
1	87731.998	522.008	640	87849.94	0.991
2	87769.82	139.94	1008.49	87952.11	1.004
3	87628.88	202.004	1137.79	87628.92	0.999
4	87583.63	155.46	1163.64	87951.95	0.997
5	87603.997	479.34	1493.34	89576.98	1.008

Table 5.3: I/O data for EOQ simulation with smaller Q-range

Q	22500	26250	30000	33750	37500
C	87641.66	87594.64	87700	87906.95	88185

We point out that the estimated effects are not independent, because $(\mathbf{X}'\mathbf{X})^{-1}$ in (3.2.29) is not diagonal.

The estimated optimum (say) $\widehat{Q_o}$ follows from the first-order optimality condition $\partial\widehat{C}/\partial Q = \widehat{\beta_1} + 2\widehat{\beta_{1;1}}x_1 = 0$, where x_1 is the coded variable corresponding with Q, which gives $\widehat{Q_o} = 28636$. This $\widehat{Q_o}$ gives the estimated minimal cost $\widehat{C_o} = 87654$. In this example, we know the true optimum so we can easily verify the estimated optimum: $\widehat{Q_o}/Q_o = 28636/25298 = 1.13$ and $\widehat{C_o}/C_o = 87654/87589 = 1.001$ so the cost virtually equals the true minimum, even though the input is 13% off. This illustrates the well-known insensitivity property of the EOQ formula.

We also experiment with a smaller experimental area; i.e., a smaller Q-range. We assume that the center of this new area is still close to the true optimum and we pick the same number of points in the interval $\left[0.75\,\bar{Q}, 1.25\,\bar{Q}\right]$, still coding them into the interval $[-1, +1]$. The Taylor series argument suggests that this smaller area gives a better approximation locally. Row 1 of Table 5.3 shows the Q values in the smaller experimental area; row 2 gives the corresponding simulation outputs.

Regression analysis of the I/O data in Table 5.3 gives Table 5.4 and the scatterplot of Figure 5.9. Comparison with Table 5.2 and Figure 5.8 shows that the smaller Q-range gives a more accurate metamodel. The resulting estimated optimum $\widehat{Q_o}$ is 25115, which gives $\widehat{C_o} = 87607$ so $\widehat{Q_o}/Q_o = 25115/25298 = 0.99$ and $\widehat{C_o}/C_o = 87618/87589 = 1.0003$, meaning that the optimal order quantity is only 1% below the true EOQ and the corresponding cost virtually equals the true cost. Comparison with the old results ($\widehat{Q_o}/Q_o = 1.13$ and $\widehat{C_o}/C_o = 1.001$) shows that the smaller Q-range improves the estimated optimum, indeed.

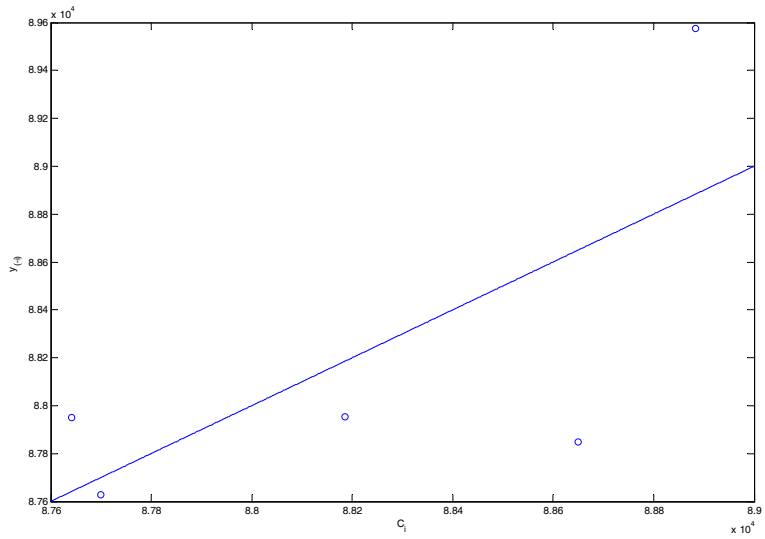


Figure 5.8: Scatterplot of the EOQ regression metamodel

5.3 Robust optimization of EOQ model using Taguchi and RSM

In this section, we focus on the robust formulation of the EOQ model, introduced in Section 2.1.3. Restricting our attention to possible variations on the demand rate — supposed to be normally distributed, we experiment with a 'low' and 'high' uncertainty: $\sigma_a = 0.10\mu_a$ and $\sigma_a = 0.50\mu_a$. Because these standard deviations can give a negative value for a — although with a small probability —, we adjust the normal distribution in (2.1.15) slightly by cutting the negative tail off, according to what discussed in Section 3.2. However, we ignore this adjustment in our further analysis.

Following Myers and Montgomery (2002), we select 'a few' values (levels) for the environmental factors. Those authors use only two values per environmental factor (which suffices to estimate its main effect and its interactions with the decision factors). We, however, use *Latin Hypercube Sampling* (LHS) to select 'a few' values for the environmental factors (because LHS is popular in risk and uncertainty analysis; see Kleijnen (2008)). We follow the procedure described in Section 3.2, taking into account that demand values cannot be negative. So, LHS splits the range of possible a values ($0 < a < \infty$) into $n_e = 5$ equally likely subranges, namely $(0, \mu_a - 0.85\sigma_a]$, $(\mu_a - 0.85\sigma_a, \mu_a - 0.73\sigma_a]$, $(\mu_a - 0.73\sigma_a, \mu_a + 0.73\sigma_a]$, $(\mu_a + 0.73$

Table 5.4: Cross-validation of EOQ regression metamodel with smaller range

i	$\widehat{\beta_{0(-i)}}$	$\widehat{\beta_{1(-i)}}$	$\widehat{\beta_{1;1(-i)}}$	$\widehat{y_{(-i)}}$	$\widehat{y_{(-i)}}/C_i$
0	87698.26	279.798	214.78		
1	87704.57	309.26	172.69	87633.242	0.999
2	87707.76	274.26	206.86	87612.056	1.000
3	87696.62	279.798	216.71	87698.26	0.9999
4	87690.03	274.99	221.64	87891.854	0.9997
5	87692.38	307.23	253.97	88192.838	1.001

Table 5.5: I/O simulation data for EOQ model with uncertain demand rate

		a				
		4530,34	5478,85	7687,37	9329,26	11559,02
Q	15000	51177,72	61421,54	85273,65	103006	127087,4
	22500	51094,63	61085,52	84348,68	101643,2	125130
	30000	51615,59	61480	84448,7	101524,3	124713,8
	37500	52378,16	62166,7	84958,71	101902,9	124914,1
	45000	53261,54	62999,49	85673,71	102530,4	125422,6

$\sigma_a, \mu_a + 0.85\sigma_a]$, $(\mu_a + 0.85\sigma_a, \infty)$. Notice that the 'base' value μ_a has zero probability, but a value 'close' (namely less than $0.73\sigma_a$ away) has 20% probability. This gives the a values in Table 5.5, which uses the relatively high uncertainty $\sigma_a = 0.50\mu_a$. It might be reasonable to add an upper bound to a, since in practice the demand will not go beyond a given value: if no other information is available, we can fix the upper bound at (say) $\mu_a + 3\sigma_a$.

For the decision variable Q we select the five values that we also used in Table 5.1. We *cross* the two designs for a and Q respectively, as is usual in a Taguchian approach. However, we could also have used LHS to get a combined design for a and Q. We also use a CCD instead of LHS (see Table 5.10); Myers and Montgomery (2002) also discuss designs more efficient than crossed designs.

We run the EOQ simulation model for all 5×5 combinations of the inputs (decision and environmental inputs), which gives Table 5.5.

We again code the inputs; see (3.2.10). So x_1 corresponds with Q and x_2 with a; e.g., $a = 7687,37$ corresponds with $x_2 = -0.1017$ (not exactly zero, because of the sampling that LHS does). Furthermore, if $\sigma_a = 0.50\mu_a = 4000$ and $b_2 = 2.85 \times 10^{-4}$, then the standard deviation of x_2 is $\sigma_2 = 4000 \times 2.85 \times 10^{-4} = 1.14$.

To analyze these I/O data, we might compute the estimated *conditional* variance $\widehat{var(C|Q_i)}$ from the row with Q_i $(i = 1, \ldots, 5)$ in Table 5.5; also see Lee and Nelder (2003). Instead we follow Myers and Montgomery (2002) and estimate the variance using all the elements in this table; also see (3.2.16). The latter approach gives a better estimator, provided that the RSM metamodel (3.2.8) is correct.

To compute the OLS estimates, we must re-arrange the 5×5 elements of Table

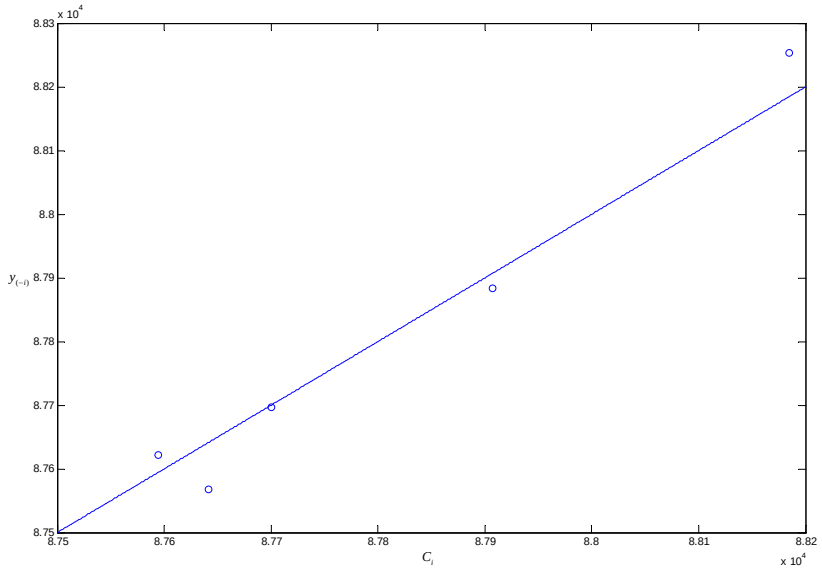


Figure 5.9: Scatterplot of the EOQ regression metamodel for smaller Q-range

5.5 into the $n \times q$ **X**-matrix of (3.1.3) where now $n = 25$ and $q = 5$; **w** now becomes a vector with the 25 simulation outputs C. This gives the estimated intercept $\widehat{\beta_0}$, the estimated first-order effect $\widehat{\beta_1}$ and second-order effect $\widehat{\beta_{1;1}}$ of Q, the estimated first-order effect $\widehat{\gamma_1}$ of a, and the interaction $\widehat{\delta_{1;1}}$, which are displayed in the row denoted by 0 (zero rows eliminated) in Table 5.6. The rest of this table displays the cross-validation results (analogous to Table 5.2). This table gives the scatterplot in Figure 5.10. This table and this figure suggest that this metamodel is adequate for robust optimization through RSM. Comparing Figures 5.8 and 5.10 suggests that the first figure is much worse; however, using the same scale in both figures (not displayed) changes that impression.

Note: To check the negative sign of $\widehat{\delta_{1;1}}$ (interaction between Q and a), we use the analytical solution (2.1.4) to derive $\partial^2 C/\partial Q \partial a = -K/Q^2$, which is indeed negative.

Apart from cross validation, we also test the relative importance of the individual regression coefficients, in order to check whether the structure of the regression model can be reduced or not. Table 5.7 shows the results obtained for the regression model built over $Q \in (0.5\bar{Q}, 1.5\bar{Q})$ and $\sigma_a = 0.5\bar{a}$; we fix $\alpha = 0.05$, which provides a confidence level of 95%, so the threshold below which the null hypothesis is accepted

Table 5.6: Cross-validation of regression metamodel for RO of EOQ

i	$\widehat{\beta_{0(-i)}}$	$\widehat{\beta_{1(-i)}}$	$\widehat{\beta_{1;1(-i)}}$	$\widehat{\gamma_{1(-i)}}$	$\widehat{\delta_{1;1(-i)}}$	$\widehat{y_{(-i)}}$	$\widehat{y_{(-i)}}/C_i$
0	88150.40	190.56	1058.33	36774.03	-899.67		
1	88144.21	172.94	1088.31	36755.96	-863.54	51440.09	1.005
2	88147.70	181.73	1072.54	36768.01	-887.64	61545.93	1.002
3	88152.19	198.89	1046.41	36774.09	-899.80	85169.34	0.999
4	88154.29	214.81	1026.29	36764.26	-880.14	102725.72	0.997
5	88157.15	259.16	976.22	36714.38	-780.37	126368.95	0.994
6	88150.48	190.51	1058.24	36773.93	-899.57	51096.08	1.000
7	88154.53	188.27	1054.63	36770.90	-896.54	61150.15	1.001
8	88164.19	182.52	1046.82	36773.90	-899.54	84550.05	1.002
9	88172.91	177.00	1040.41	36784.95	-910.59	101956.72	1.003
10	88190.37	165.55	1028.40	36817.51	-943.16	125653.78	1.004
11	88124.57	190.56	1090.86	36793.63	-899.67	51330.94	0.994
12	88131.43	190.56	1081.72	36783.93	-899.67	61275.30	0.997
13	88146.40	190.56	1063.03	36774.05	-899.67	84407.52	1.000
14	88158.08	190.56	1049.58	36776.69	-899.67	101600.85	1.001
15	88177.60	190.56	1028.69	36795.56	-899.67	124973.16	1.002
16	88136.05	182.81	1071.52	36789.93	-883.77	52147.29	0.996
17	88137.51	183.42	1069.82	36783.76	-889.94	61965.54	0.997
18	88139.92	184.45	1067.07	36774.13	-899.57	84805.76	0.998
19	88141.21	185.03	1065.63	36769.57	-904.12	101775.07	0.999
20	88142.70	185.75	1064.09	36765.65	-908.04	124813.22	0.999
21	88140.63	218.39	1105.69	36745.49	-956.75	53675.97	1.008
22	88144.24	210.75	1090.83	36760.27	-927.18	63283.89	1.005
23	88148.76	198.15	1069.18	36773.97	-899.79	85768.71	1.001
24	88150.72	188.53	1055.65	36773.21	-901.30	102506.95	1.000
25	88152.94	164.73	1027.41	36751.56	-944.60	125152.04	0.998

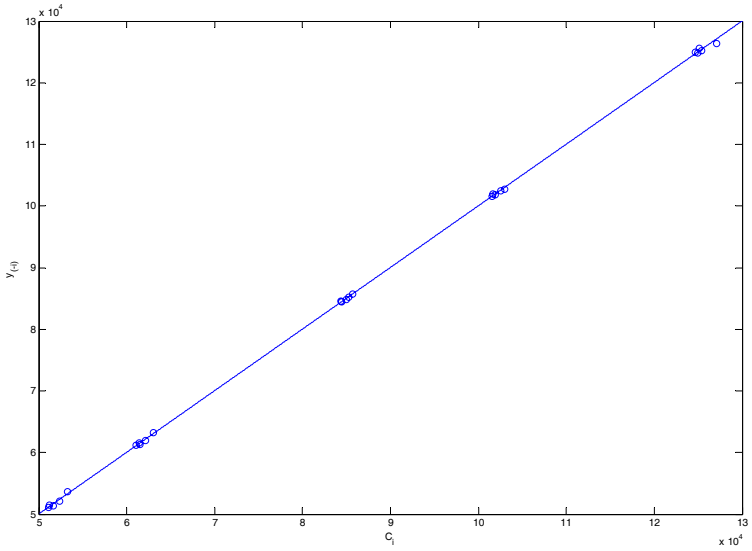


Figure 5.10: Scatterplot of the EOQ regression metamodel for RO

is $t_{1-\alpha/2,n-q} = 2.086$.

Table 5.7: t-statistics applied to RSM with $Q \in (0.5\bar{Q}, 1.5\bar{Q})$ and $\sigma_a = 0.5\mu_a$

RSM coeff.	$t_0(\beta_j)$	CI
$\widehat{\beta_0}$	13111.04	$[88010.14, 88290.65]$
$\widehat{\beta_1}$	3.11	$[62.67, 318.45]$
$\widehat{\beta_{1;1}}$	10.29	$[843.91, 1272.74]$
$\widehat{\gamma_1}$	620.67	$[36650.44, 36897.62]$
$\widehat{\delta_{1;1}}$	-10.74	$[-1074.45, -724.89]$

Table 5.9 and Fig. 5.11 show the cross validation results for the experiment with smaller uncertainty $\sigma_a = 0.10\mu_a$, whose I/O data are represented in Table 5.8.

Using a RSM metamodel like the one in the first row of Table 5.6, Myers and Montgomery (2002) derive *contour plots* for the mean and variance. Because our EOQ example has a single decision variable, we do not superimpose contour plots but present the following two plots:

- Figure 5.12 shows the plot for Q (the decision variable) versus $\widehat{C}$ (mean output, estimated through regression analysis); see (3.2.15) with the regression param-

Table 5.8: I/O simulation data for EOQ model with uncertain demand rate

		a				
		6076.55	7438.96	7.832.04	8595.36	9101.10
Q	15000	67876.77	82590.79	86836.02	95079.90	111541.90
	22500	67381.35	81732.06	85872.48	93912.80	99239.95
	30000	67696.14	81865.20	85953.20	93891.76	99151.47
	37500	68335.02	82395.09	86451.64	94329.13	99548.38
	45000	69135.94	83123.34	87158.94	94995.71	100188

Table 5.9: Cross-validation of regression metamodel for RO of EOQ with smaller a-range

i	$\widehat{\beta_{0(-i)}}$	$\widehat{\beta_{1(-i)}}$	$\widehat{\beta_{1;1(-i)}}$	$\widehat{\gamma_{1(-i)}}$	$\widehat{\delta_{1;1(-i)}}$	$\widehat{y_{(-i)}}$	$\widehat{y_{(-i)}}/C_i$
0	83374.03	307.26	1070.93	15824.45	-387.14		
1	83373.72	296.40	1082.41	15814.75	-367.74	67977.23	1.001
2	83375.07	313.61	1062.48	15825.97	-390.18	82516.93	0.999
3	83375.95	316.08	1058.27	15824.30	-386.85	86725.30	0.999
4	83378.77	320.83	1047.87	15815.61	-369.45	94878.21	0.998
5	83382.36	325.08	1036.43	15802.71	-343.66	100240.08	0.997
6	83383.68	300.95	1064.25	15813.17	-375.86	67498.17	1.002
7	83387.18	299.36	1060.43	15820.66	-383.35	81915.79	1.002
8	83388.19	299.02	1059.09	15824.72	-387.41	86079.62	1.002
9	83390.53	298.34	1055.76	15836.08	-398.77	94178.12	1.003
10	83392.58	297.83	1052.67	15847.45	-410.14	99559.36	1.003
11	83353.46	307.26	1092.07	15842.31	-387.14	67511.15	0.997
12	83367.41	307.26	1078.48	15825.81	-387.14	81799.08	0.999
13	83370.36	307.26	1075.25	15824.40	-387.14	85915.31	1.000
14	83375.54	307.26	1069.02	15825.18	-387.14	93908.39	1.000
15	83378.87	307.26	1064.54	15828.47	-387.14	99207.34	1.001
16	83355.49	295.13	1083.75	15846.11	-365.47	68110.62	0.997
17	83362.44	300.30	1080.18	15827.79	-383.80	82233.12	0.998
18	83363.51	301.14	1079.72	15824.25	-387.34	86297.79	0.998
19	83364.95	302.35	1079.27	15818.05	-393.54	94183.08	0.998
20	83365.52	302.94	1079.30	15813.89	-397.69	99401.86	0.999
21	83372.90	346.65	1112.57	15789.26	-457.51	69500.37	1.005
22	83372.29	317.81	1084.94	15821.92	-392.19	83245.96	1.001
23	83372.43	314.61	1081.48	15824.57	-386.90	87251.28	1.001
24	83373.06	310.04	1075.65	15826.26	-383.51	95037.07	1.000
25	83373.96	307.40	1071.20	15824.62	-386.79	100190.39	1.000

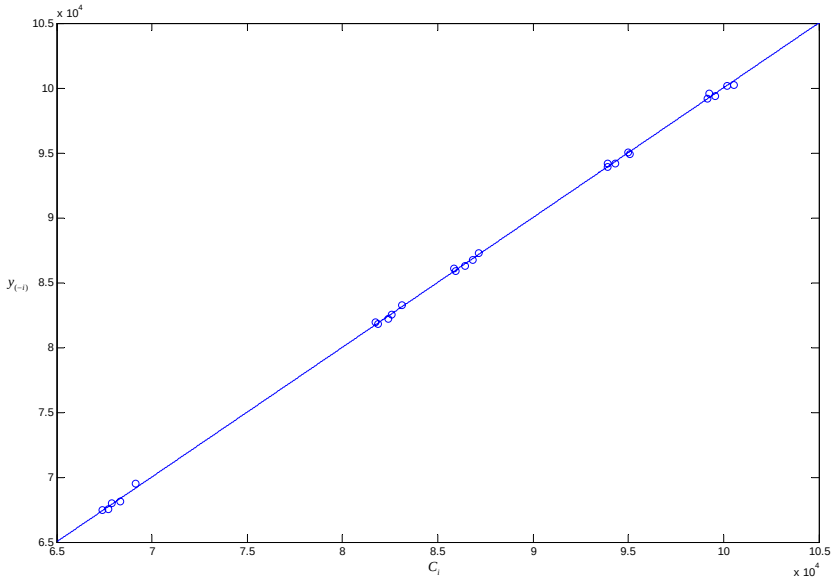


Figure 5.11: Scatterplot of the EOQ regression metamodel for smaller a-range

eters β_0, $\boldsymbol{\beta}$, $\mathbf{B}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\Delta}$ replaced by their estimates. Indeed, $E(a)$ in (3.2.15) is a known input value: it is not exactly equal to μ_a in (2.1.15) because we resample negative a values, which have a probability of nearly 2% for high σ_a and virtually zero for small σ_a; we could also have estimated $E(a)$ through $\overline{a} = \sum_{1}^{5} a_i/5$ where a_i is shown in Table 5.5.

- Figure 5.13 shows Q versus $\widehat{\sigma_C}$, the estimated standard deviation of C predicted through the RSM metamodel. We prefer the standard deviation over the variance because the former uses the same scale as the simulated cost C and its regression estimate $\widehat{C}$. We use (3.2.16) with $\boldsymbol{\gamma}$, $\boldsymbol{\Delta}$, and σ_ϵ^2 replaced by their estimates, including the MSR estimator (3.2.18) of σ_ϵ^2. Notice that σ_a^2 is a known input value, so we also know the variance of the corresponding coded variable x_2, namely $\sigma_2^2 = 1.14^2 = 1.3$. Altogether we obtain $\widehat{\sigma_C} = [(\widehat{\gamma}_1 + \widehat{\delta_{1;1}x_1})^2\sigma_2^2 + \widehat{\sigma}_\epsilon^2]^{1/2} = [(36755.96 - 863.54x_1)^2 \times 1.3 + 4.6224 \times 10^4]^{1/2}$. Figure 5.13 shows this second-order polynomial, which actually resembles a linearly decreasing function in the relatively small domain of Q that is pictured; also see the next Note.

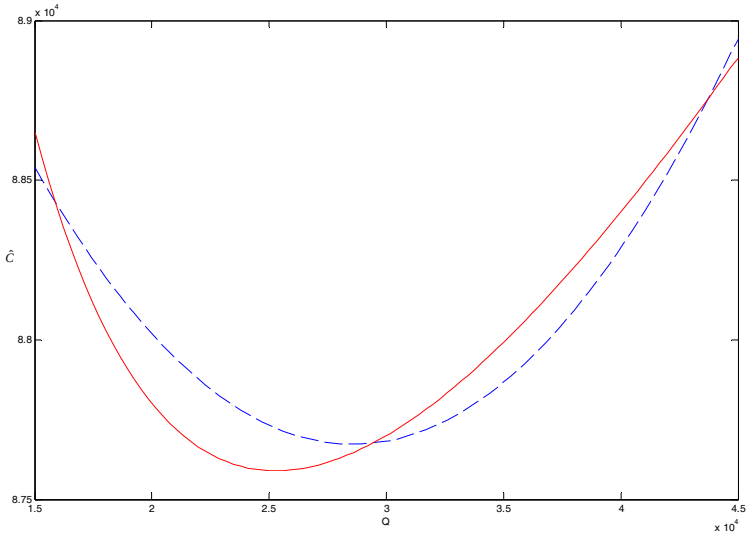


Figure 5.12: Regression estimated (dashed) curve and true (solid) curve for mean cost versus order quantity

Note: For this simple example we know the true I/O function of the simulation model, namely (2.1.4). So the true standard error of the cost C is

$$\sigma_C = \sigma\left(\frac{aK}{Q} + ac + \frac{hQ}{2}\right) = \sigma\left(\frac{hQ}{2} + [\frac{K}{Q} + c]a\right) = \left(\frac{K}{Q} + c\right)\sigma_a = c\sigma_a + \frac{K\sigma_a}{Q}. \tag{5.3.1}$$

It could have been computed analytically as well, using Eq. 5.3.2

$$\sigma^2[C(Q,a)|Q] = E[C^2(Q,a)|Q] - E^2[C(Q,a)|Q]. \tag{5.3.2}$$

Since Q is not a random variable, we can express the conditional expectation of $C(Q,a)$ given Q as

$$E[C|Q] = \int_0^{+\infty} C(Q,a)p(a)da\,, \tag{5.3.3}$$

where

$$p(a) = \frac{1}{\sigma_a\sqrt{2\pi}} e^{-(a-\mu_a)^2/(2\sigma_a^2)}\,. \tag{5.3.4}$$

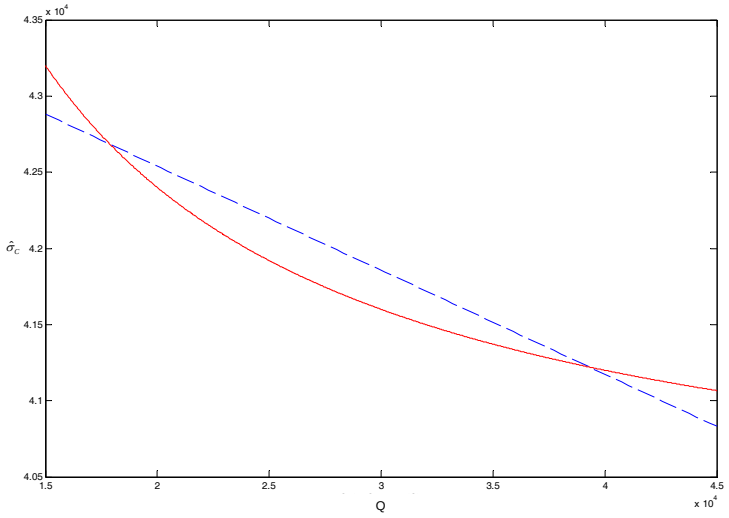


Figure 5.13: Regression estimated (dashed) curve and true (solid) curve for standard deviation of cost versus order quantity

Combining Eqs. 5.3.2 and 5.3.3, we obtain:

$$\begin{aligned}\sigma^2[C|Q] &= \int_0^{+\infty} \left(\frac{aK}{Q} + ac + \frac{hQ}{2}\right)^2 \cdot p(a)da - \left[\int_0^{+\infty} \left(\frac{aK}{Q} + ac + \frac{hQ}{2}\right) p(a)\, da\right]^2 \\ &= \left(\frac{K}{Q} + c\right)^2 \int_0^{+\infty} a^2 p(a)da + 2\left(\frac{K}{Q} + c\right)\left(\frac{hQ}{2}\right)\int_0^{+\infty} a\, p(a)\, da \\ &\left(\frac{hQ}{2}\right)^2 \int_0^{+\infty} p(a)\, da - \left[\left(\frac{K}{Q} + c\right)\int_0^{+\infty} a\, p(a)da + \left(\frac{hQ}{2}\right)\int_0^{+\infty} p(a)da\right]^2\end{aligned}$$

We recall that for a normal random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ the cumulative distribution function is defined as follows:

$$\Phi_{\mu,\sigma^2}(x) = \int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} e^{-(u-\mu)^2/(2\sigma^2)} du = \Phi_{0,1}\left(\frac{x-\mu}{\sigma}\right), \qquad (5.3.5)$$

where tables are provided to compute the standard cumulative distribution function $\Phi_{0,1}(x)$. In particular, we know that $\Phi_{\mu,\sigma^2}(+\infty) = 1$ and looking up in the tables, we

find that $\Phi_{\mu,\sigma^2}(0) = \Phi_{0,1}(-\mu_a/\sigma_a) = \Phi_{0,1}(-2) \approx 0$. So:

$$\begin{aligned}
\sigma^2[C|Q] &= \left(\frac{K}{Q}+c\right)^2 E[a^2] + \left(\frac{hQ}{2}\right)^2 \left[\Phi_{\mu_a,\sigma_a^2}(+\infty) - \Phi_{\mu_a,\sigma_a^2}(0)\right] + 2\left(\frac{K}{Q}+c\right)\left(\frac{hQ}{2}\right)\mu_a \\
&\quad - \left(\frac{K}{Q}+c\right)^2 \mu_a^2 - \left(\frac{hQ}{2}\right)^2 \left[\Phi_{\mu_a,\sigma_a^2}(+\infty) - \Phi_{\mu_a,\sigma_a^2}(0)\right]^2 \\
&\quad - 2\left(\frac{K}{Q}+c\right)\left(\frac{hQ}{2}\right)\mu_a \left[\Phi_{\mu_a,\sigma_a^2}(+\infty) - \Phi_{\mu_a,\sigma_a^2}(0)\right] \\
&\approx \left(\frac{K}{Q}+c\right)^2 \sigma_a^2 + \left(\frac{K}{Q}+c\right)^2 \mu_a^2 + \left(\frac{hQ}{2}\right)^2 + 2\left(\frac{K}{Q}+c\right)\left(\frac{hQ}{2}\right)\mu_a \\
&\quad - \left(\frac{K}{Q}+c\right)^2 \mu_a^2 - \left(\frac{hQ}{2}\right)^2 - 2\left(\frac{K}{Q}+c\right)\left(\frac{hQ}{2}\right)\mu_a \\
&\approx \left(\frac{K}{Q}+c\right)^2 \sigma_a^2 \qquad (5.3.6)
\end{aligned}$$

We also plot this σ_C against Q in Figure 5.13 (assuming fixed cost parameters K and c, and demand variance σ_a^2). Comparing the two curves in Figure 5.13 shows that the estimated curve is an adequate approximation.

From Figures 5.12 and 5.13 we derive the 'estimated robust optimal' order quantity (say) $\widehat{Q^+}$, which we define as the quantity that minimizes the estimated mean $\widehat{C}$ while keeping the estimated standard deviation $\widehat{\sigma_C}$ below a given threshold T:

$$\begin{aligned}
\min_{Q} \quad & \widehat{C} \\
s.t. \quad & \widehat{\sigma_C} \le T
\end{aligned} \qquad (5.3.7)$$

We solve this constrained minimization problem through MATLAB's `fmincon`. For example, if $T = 4.25 \times 10^4 = 42500$, then Figure 5.13 implies $\widehat{Q^+} = 2.8568 \times 10^4 = 28568$. However, let T become smaller, e.g., $T = 4.15 \times 10^4 = 41500$. Then Figure 5.13 implies $\widehat{Q^+} = 3.5222 \times 10^4 = 35222$; see Figure 5.14, in which the curve becomes a horizontal line with 'height' $\widehat{Q^+}$ if the threshold is high enough.

We point out that Section 5.2 gave the classic EOQ $\widehat{Q_o} = 28636$, assuming the demand rate equals the nominal value. Now we use a different model, assuming different demand rates. The latter model gives an estimated optimal order quantity $\widehat{Q^+}$ that differs from $\widehat{Q_o}$. This difference is nearly 25% if the managers are risk-averse (low threshold T).

We assume that managers cannot give a single, fixed value for the threshold, so we vary the threshold over the interval $[41067, 43200]$. This gives the estimated *Pareto frontier* in Figure 5.15. This figure demonstrates that if managers prefer low costs variability, then they must pay a price; i.e., the expected cost increases.

We repeat the experiment with a smaller σ_a (lower demand variability), which implies a less volatile environment. Some reflection shows that we cannot keep the

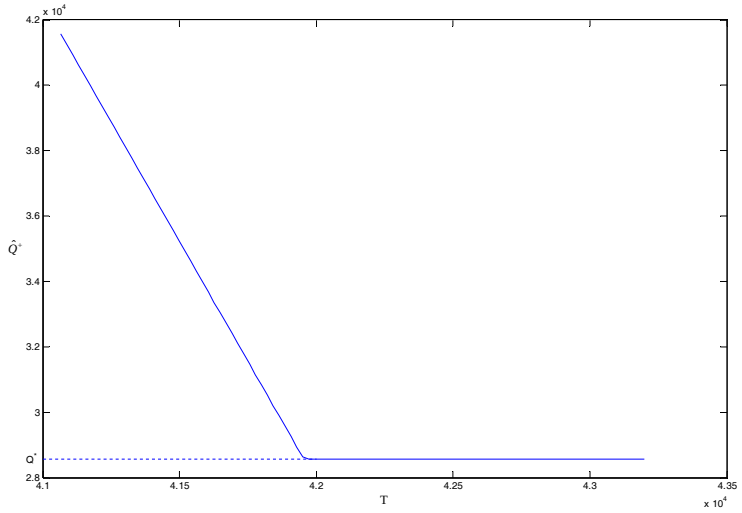


Figure 5.14: Regression estimated robust optimal value for EOQ against threshold for standard deviation of cost

threshold values T the same in environments with different magnitudes of *volatility*. The new threshold values give the estimated Pareto frontier of Figure 5.16. Comparing the estimated Pareto frontiers of Figures 5.15 and 5.16 demonstrates that a less volatile world gives lower mean cost. Moreover, this comparison quantifies the benefits of obtaining more information on the uncertain demand rate (e.g., a marketing survey may decrease the standard deviation of the demand rate).

The estimated Pareto frontier is built on the estimates $\widehat{\boldsymbol{\zeta}}$, so we further analyze this frontier. Whereas Myers and Montgomery (2002) use rather complicated confidence intervals, we use *parametric bootstrapping*. Sampling from a multivariate normal distribution according to Eq. 3.2.32 and performing all the steps described in Section 3.2, we obtain two bootstrapped regression models, $\widehat{C_b^*}$ and $\widehat{\sigma_{C_b^*}}$ $(b = 1, \ldots, B)$. We then solve the constrained optimization process using these metamodels, to derive B estimated Pareto frontiers: the results are depicted in Figure 5.17, where we select $B = 50$ and derive the true Pareto frontier from the analytical costs and its standard deviation (5.3.1); we also display the original estimated frontier of Figure 5.15. This figure demonstrates that bootstrapping gives a good idea of the variability of the estimated Pareto frontier; the bundle of bootstrapped curves 'envelop' the original estimated curve and the true curve. We observe that the bundle of bootstrapped

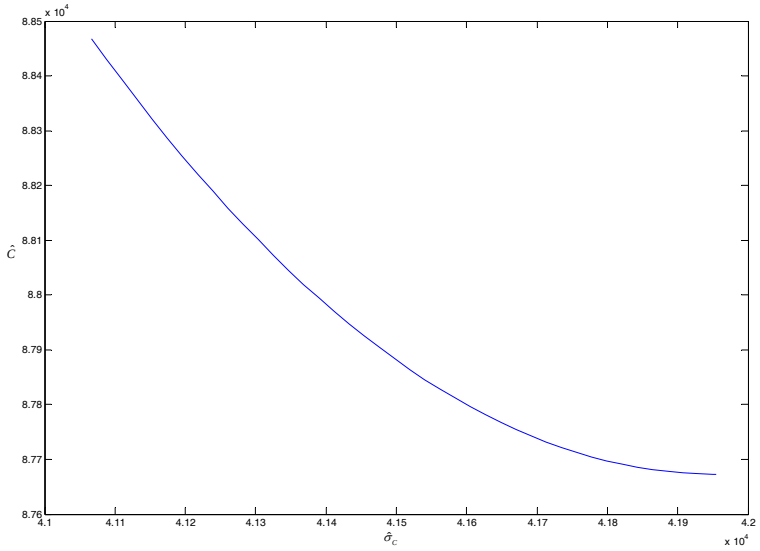


Figure 5.15: Estimated Pareto frontier for EOQ simulation, based on regression metamodels, with threshold for standard deviation of cost

estimated costs does not completely "cover" the true curve; neither does the bundle for the bootstrapped standard deviations; see Figures 5.18 and 5.19.

Moreover, it is possible to further detail such analysis by determining a confidence region based on the bootstrap estimates of the Pareto frontier. Such a region will provide a quantitative measure of the variability of the outputs of the performed optimization process, due to the variability of the estimated metamodel parameters. Moreover, as discussed by Kleijnen and Gaury (2003), confidence regions can also be useful to determine whether the difference in the solutions obtained by alternative approaches is significant or not. At this aim, we follow Kleijnen and Gaury (2003), and compute a $(1-\alpha)$ confidence region according to the following equation, as suggested by Johnson and Wichern (1992):

$$B\left(\overline{\mathbf{Y}}-\mathbf{Y}_b\right)^T\mathbf{S}^{-1}\left(\overline{\mathbf{Y}}-\mathbf{Y}_b\right)\leq\frac{2(B-1)}{B-2}f_{2;B-2}(1-\alpha) \tag{5.3.8}$$

where $\mathbf{Y}_b=\left(\widehat{C_b^*},\widehat{\sigma_{C_b^*}}\right)$ denotes the b-th pair of bootstrapped criteria, $\overline{\mathbf{Y}}=\sum_{b=1}^{B}\mathbf{Y}_b/B$ is the classic estimator for the mean, $\mathbf{S}=\sum_{b=1}^{B}(\mathbf{Y}_b-\overline{\mathbf{Y}})(\mathbf{Y}_b-\overline{\mathbf{Y}})^T$ is the classic estimator for the covariance matrix, and $f_{2;B-2}(1-\alpha)$ denotes the $(1-\alpha)$ quantile of the F-statistic with $(2,B-2)$ degrees of freedom.

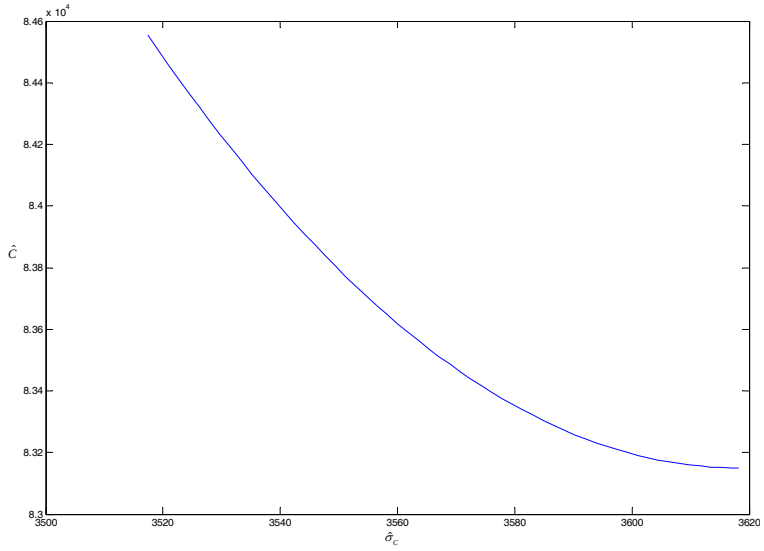


Figure 5.16: Less volatile world: estimated Pareto frontier for EOQ simulation with threshold for standard deviation of cost

Notice that this method asks the two output factors to be bivariate normally distributed; therefore, we first need to verify this hypothesis. We performed the test described in what follows, again according to Johnson and Wichern (1992): we define the Squared Generalized Distance as

$$D_b^2 = \left(\mathbf{Y}_b - \overline{\mathbf{Y}}\right)^T \mathbf{S}^{-1} \left(\mathbf{Y}_b - \overline{\mathbf{Y}}\right) , \qquad b = 1, \ldots, B \tag{5.3.9}$$

When the parent population is bivariate normal, each squared generalized distance D_b^2 should follow a χ^2 distribution. Therefore a simple — though rough — procedure to check whether the normality hypothesis holds would be to compute the fraction of points within a contour and compare it with the theoretical probability according to $D_b^2 \leq \chi_2^2(0.50)$. If around half of the squared generalized distances is smaller than the 50% quantile of the χ^2 statistic with 2 degrees of freedom, then the null hypothesis of bivariate normality is accepted. A more formal method for judging the joint normality of the bootstrapped data set is based on the so-called chi-square plot, which is obtained as follows:

1. Order the squared generalized distances D_b^2 from smallest to largest as $D_{(1)}^2, \ldots, D_{(B)}^2$.
2. Graph the pairs $\left(D_{(b)}^2, \chi_2^2((b-1/2)/B)\right)$, where $\chi_2^2((b-1/2)/B)$ is the $(b-$

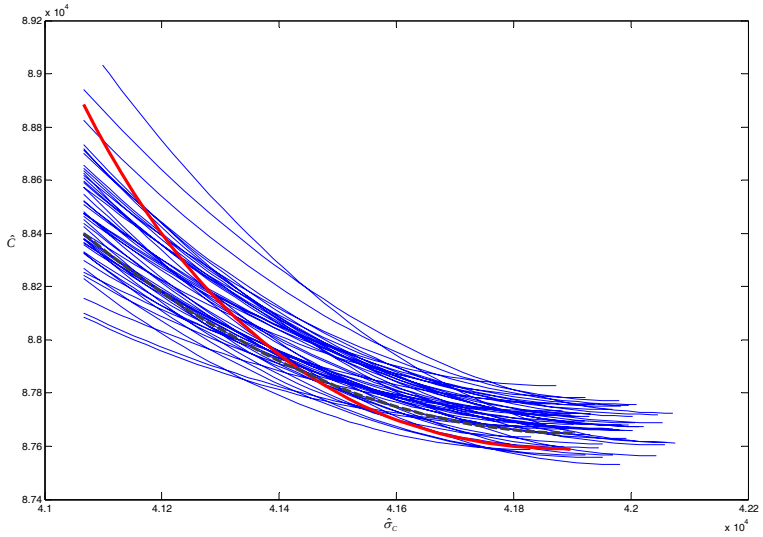


Figure 5.17: Bootstrapped Pareto frontiers, original estimated frontier (dashed curve), and true frontier (heavy curve), based on regression metamodels

$1/2)/B$ quantile of the χ^2 distribution with 2 degrees of freedom.

The resulting plot should resemble a straight line. A systematic curved pattern suggests lack of normality, whereas a few points highly deviating from the linear behaviour denote possible outliers, which might require further analyses.
To illustrate the procedure described so far, we select a threshold value, e.g. $T = 42660$, and extract the optimal solutions of the optimization problem in Eq. 5.3.7 solved on the B pairs of bootstrapped metamodels; these points are plotted in Fig. 5.20 and used to compute the 90% confidence region (i.e. $\alpha = 0.10$), which is delimited by the ellipse shown in the figure. The test for normality returned the chi-square plot in Fig. 5.21: we can observe that — based on this plot — the normality assumption is not fully satisfied; nevertheless, for our purposes, we consider such result to be acceptable. In order to develop a more accurate study, we should mention that, in case the outputs are not normally distributed, we can properly transform them in order to let them be "near normal". The transformation we suggest to apply is the so-called *power transformation*, which is now shortly described: focussing on the bivariate case, let λ_1, λ_2 be the power transformations for the two measured characteristics; then,

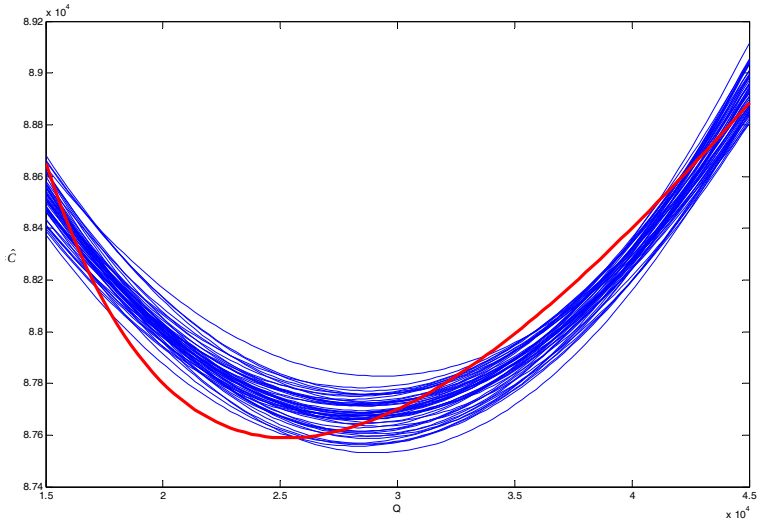


Figure 5.18: Bootstrapped estimated costs, based on regression metamodels, and true cost (heavy curve)

each transformed variable is defined as follows:

$$x^{(\lambda_k)} = \begin{cases} \dfrac{x^{\lambda_k} - 1}{\lambda_k} & \lambda_k \neq 0 \\ \ln x & \lambda_k = 0 \end{cases}, \tag{5.3.10}$$

where x denotes the original non-normal variable and $x^{(\lambda_k)}$ is the corresponding transformed variable ($k = 1, 2$). Each λ_k can be selected by maximizing

$$\ell_k(\lambda_k) = -\frac{n}{2} \ln \left[\frac{1}{n} \sum_{j=1}^{n} \left(x_{kj}^{(\lambda_k)} - \overline{x_k^{\lambda_k}} \right)^2 \right] + (\lambda_k - 1) \sum_{j=1}^{n} \ln x_{kj} \tag{5.3.11}$$

where $x_{k1}, \ldots, x_{kn}$ are the n observations on the k-th variable ($k = 1, 2$) and

$$\overline{x_k^{(\lambda_k)}} = \frac{1}{n} \sum_{j=1}^{n} x_{kj}^{(\lambda_k)} . \tag{5.3.12}$$

For further details on power transformation methods, refer to Johnson and Wichern (1992).

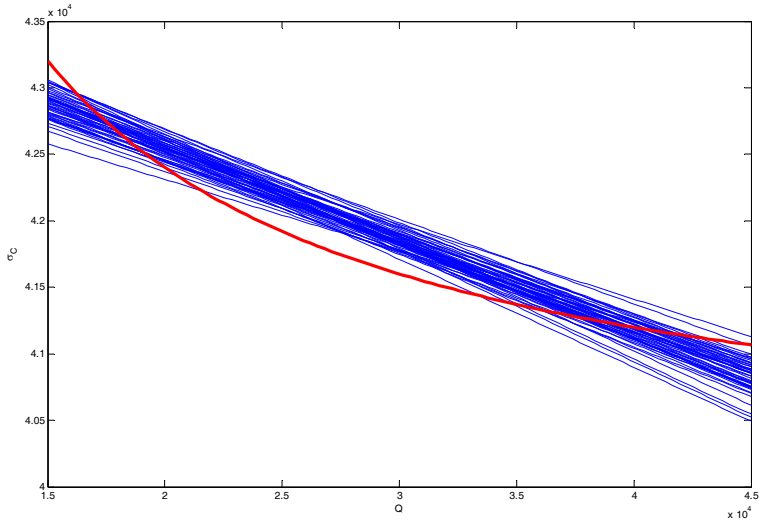


Figure 5.19: Bootstrapped standard deviations of the cost, based on regression metamodels, and true standard deviation of the cost (heavy curve)

Actually, we can validate our (fast) bootstrap procedure as follows. Our EOQ simulation is the opposite of *expensive simulation*: some realistic simulations take hours or weeks for a single run, whereas bootstrapping this simulation's results still takes only seconds. So we repeat our LHS sample (say) L times; i.e., we sample the demand rate a from the normal distribution in (2.1.15) cut-off at zero, while keeping the five Q values in Table 5.5 fixed. This sample of L *macroreplicates* gives the regression estimate $\widehat{\boldsymbol{\zeta}}_l$ with $l = 1, \ldots, L$. This $\widehat{\boldsymbol{\zeta}}_l$ gives $\widehat{C_l}$ (costs estimated through RSM metamodel) and $\widehat{\sigma_{C_l}}$ (corresponding standard deviation). Figures 5.22 and 5.23 display the 50 $\widehat{C}$-curves and the 50 $\widehat{\sigma_C}$-curves respectively, computed from 50 macroreplicates. Note that the latter figure suggests that the 50 estimated curves coincide, but zooming-in reveals that the 50 curves do not coincide: these curves have little spread; see Figures 5.24 and 5.25. We point out that each macroreplicate gives a different mean and standard deviation for the coded variable x_2; e.g., $x_{2;l} = \min_k a_{l;k}$ with $l = 1, \ldots, 50$ and $k = 1, \ldots, 5$.

There is no solution for the constrained optimization problem if the LHS happens to result in an extremely high $\widehat{\sigma_C}$. Actually this happened once in our 50 macroreplicates; we simply threw away this macroreplicate, and sampled again.

Together with the threshold T this gives the estimated Pareto frontier. Repeating

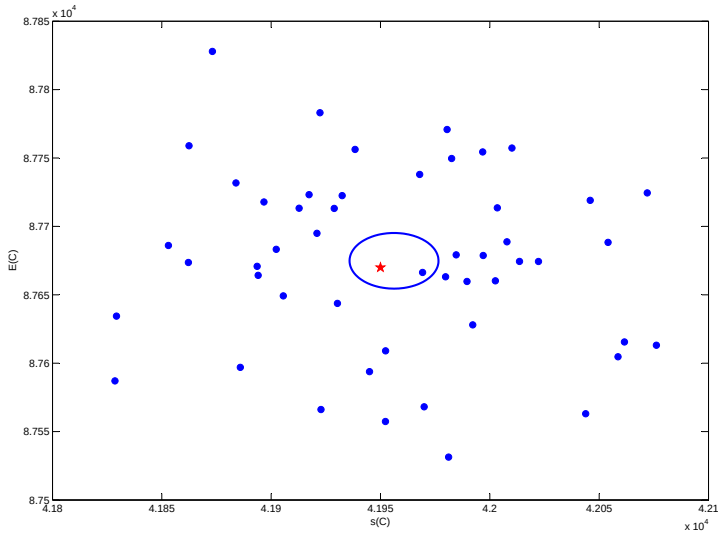


Figure 5.20: 90% confidence region based on bootstrapped estimates of mean cost and standard deviation, based on regression metamodels, for $T = 42660$.

this LHS L times gives a set of L estimated Pareto frontiers; see Figure 5.26 with $L = 50$. This figure suggests that these estimated curves all intersect near the point $(4.11, 8.83)$, but zooming-in around this point reveals that the 50 curves do not intersect in a single point. Notice that Figure 5.26 assumes that a second-order polynomial is a perfect approximation of the true I/O function, whereas the true EOQ formulas in (2.1.4) and (5.3.1) show that this assumption is false. Comparing this figure and Figure 5.17 shows that the macroreplicates give a tighter bundle. As will be explained later on, this phenomenon is explained by the negative correlations between estimated regression coefficients in the macroreplicates. In general, we could argue that — compared with bootstrapping — macroreplicates use much more computer time, and provide more information so the spread in the estimated Pareto curves is smaller.

Figure 5.27 shows that if we replace LHS by crude sampling in the macroreplicates, then bigger spread results; i.e., LHS is indeed a variance reduction technique. This bigger spread is caused by a bigger spread in the estimated regression coefficients; e.g. Figure 5.28 shows the Box plot for the estimated interaction $\widehat{\delta_{1;1}}$.

It is interesting that the spread of the estimated regression coefficients is smaller for the bootstrap than for the macroreplicates using LHS; nevertheless, the boot-

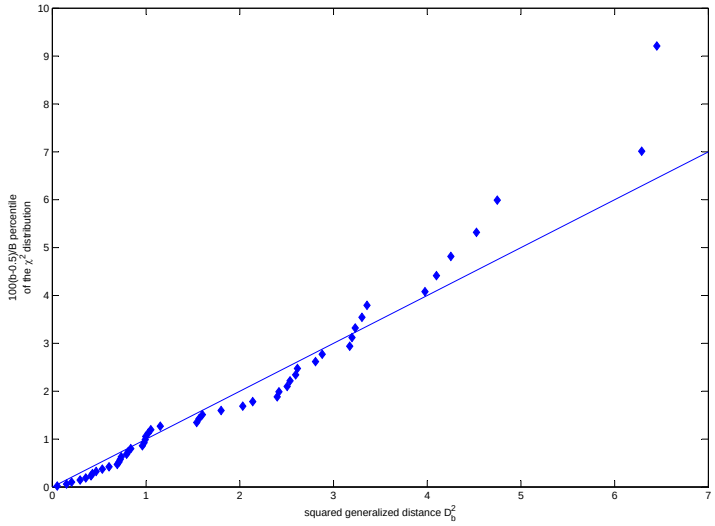


Figure 5.21: Chi-square plot of the bootstrapped estimates of mean cost and standard deviation, for $T = 42660$.

strap gives more spread in the Pareto curves. The explanation is that the estimated regression coefficients in the metamodel for the standard deviation are negatively correlated (so they compensate variations in each other's values) in the macroreplicates, whereas they are independent in the bootstrap. More precisely, the covariance matrix in (3.2.32) implies that in our experiment $cov(\widehat{\gamma^*_{1(-i)}}, \widehat{\delta^*_{1;1(-i)}}) = 0$; for the macroreplicates we use MATLAB Symbolic Math Toolbox (see The MathWorks Inc., 2007) to derive that the correlation coefficient $cor(\widehat{\gamma_{1(-i)}}, \widehat{\delta_{1;1(-i)}})$ is -1; see Figure 5.29.

Finally, we compare the (traditional Taguchian) crossed design in Table 5.5 with a CCD. A CCD for two factors (Q and a) consists of a 2^2 design (the four combinations of the two extreme values per factor -1 and 1), the four 'axial' points $((0, -\sqrt{2})$, $(0, \sqrt{2})$, $(-\sqrt{2}, 0)$, $(\sqrt{2}, 0))$, and the central point $((0, 0))$ in coded values; the value $\sqrt{2}$ is selected to make the CCD 'rotatable' (see Myers and Montgomery (2002)). The original input values plus the corresponding output values are displayed in Table 5.10.

Note: A CCD is not a subset of Table 5.5, because a CCD does not sample any factor value, whereas Table 5.5 uses LHS for the environmental factor a. Consequently, Table 5.5 does not have (say) coded values -1 and 1 for a, which are at exactly the same distance from 0.

We again validate the resulting metamodel through cross-validation; details are

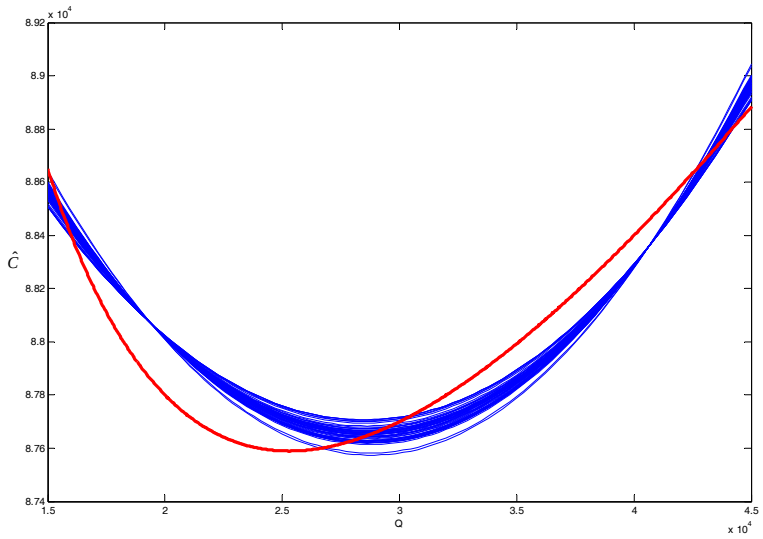


Figure 5.22: Replicated regression estimated costs, and true cost (heavy curve)

given in Table 5.11 and Figure 5.30.

We repeat our analysis for this CCD. This gives the Pareto frontier of Figure 5.31. Comparison of Figures 5.17 and 5.31 shows that the CCD with its nine combinations gives a better estimate of the true frontier (the heavy curve in Figure 5.17) than the 5×5 crossed-design does. We conjecture that the bigger design gives a more accurate OLS estimator $\widehat{\boldsymbol{\zeta}}$ of the *wrong* (misspecified) metamodel (namely, a second-order polynomial) for the true I/O function implied by the EOQ simulation model.

5.4 Simulation Optimization of the classic EOQ using Kriging

Before applying the approach discussed in Section 4.2 to the EOQ model, we use classic optimization; i.e., we ignore the uncertainty of the environment (in the next section we do account for a specific type of uncertainty).

We use the same data used for fitting the regression model, which were shown in Table 5.1. Based on these I/O data, we estimate a *Kriging* metamodel; see Fig. 5.32, which also displays the true I/O function and the second order polynomial metamodel used in Section 5.2.

Table 5.10: I/O simulation data for EOQ model with CCD design

Q	a	C
19393.40	5559.67	61945.81
19393.40	10529.69	114721.4
40606.60	5559.67	63330.64
40606.60	10529.69	114499.6
15000	8044.68	89132.55
45000	8044.68	89342.05
30000	4530.34	51615.54
30000	11559.02	124713.8
30000	8044.68	88164.67

Table 5.11: Cross-validation of regression metamodel for RO of EOQ, based on CCD

i	$\widehat{\beta_{0(-i)}}$	$\widehat{\beta_{1(-i)}}$	$\widehat{\beta_{1;1(-i)}}$	$\widehat{\gamma_{1(-i)}}$	$\widehat{\delta_{1;1(-i)}}$	$\widehat{y_{(-i)}}$	$\widehat{y_{(-i)}}/C_i$
0	88136.81	182.41	529.35	25915.14	-401.66		
1	88188.84	110.87	542.36	25843.59	-258.57	62518.17	1.0092
2	88155.43	156.81	534.00	25940.74	-452.86	114926.23	1.0018
3	88137.85	183.85	529.61	25913.70	-404.54	63342.16	1.0002
4	88104.44	137.91	521.26	25870.64	-490.66	114143.59	0.9969
5	88182.69	271.64	414.63	25915.14	-401.66	88627.80	0.9943
6	88110.59	233.40	594.89	25915.14	-401.66	89630.45	1.0032
7	88063.51	182.41	578.21	25962.65	-401.66	51346.78	0.9948
8	88178.26	182.41	501.71	25942.01	-401.66	124865.80	1.0012
9	88126.36	182.41	536.32	25915.14	-401.66	88126.36	0.9996

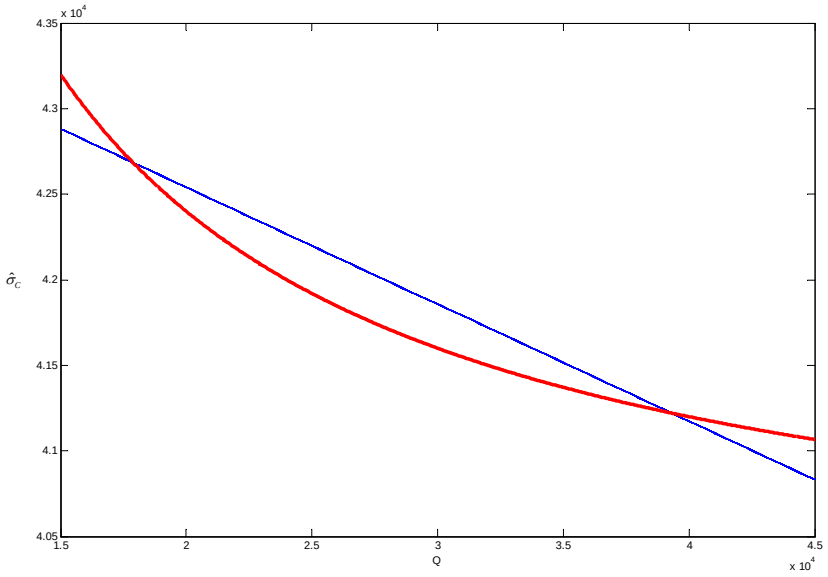


Figure 5.23: Replicated standard deviations of the cost based on regression metamodels, and true standard deviation (heavy curve)

To *validate* this Kriging metamodel, we use cross-validation. This gives the scatterplot in Fig. 5.33 for Kriging, which also shows the corresponding scatterplot for the RSM model from Fig. 5.8. The relative prediction errors $\widehat{y_{-i}}/C_i$ for Kriging are given in Table 5.12, which can be compared to the analogue Table 5.2. This validation shows that Kriging does not perform much better than the second-order polynomial does; we justify this result through observing that the EOQ model has a simple, smooth I/O function that is well approximated by a second-order polynomial in our relatively small experimental area (Taylor series argument); see also Allen et al. (2003).

To estimate the optimum (say) $\widehat{Q_o}$, we use MATLAB's `fmincon` (but we could have used any other solver). This gives $\widehat{Q_o} = 25330$ and the estimated minimal cost $\widehat{C_o} = 87523$. To verify the estimated optimum, we compute $\widehat{Q_o}/Q_o = 25330/25298 = 1.0013$ and $\widehat{C_o}/C_o = 87523/87589 = 1.0008$ so not only the cost virtually equals the true minimum, but also the optimal input is very close to the true optimal order quantity, being only 0.13% off.

We also experiment with a smaller experimental area; i.e., a smaller Q range. The smaller Q range gives a more accurate metamodel; both the resulting estimated

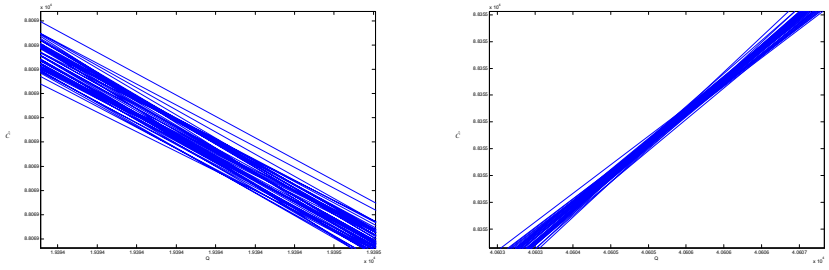


Figure 5.24: Zoom: Mean Cost, estimated through 50 macroreplicates

Table 5.12: Cross-validation of Kriging metamodel for EOQ total cost.

i	$\widehat{y_{-i}}$	$\widehat{y_{-i}}/C_i$
1	84924.76	0.9922
2	85147.72	1.0058
3	85371.79	1.0073
4	85523.74	1.0030
5	85068.57	0.9894

optimum and the corresponding cost are much smaller than 0.1% w.r.t. the analytical solution. This is also evident from Figure 5.34, where the RSM and Kriging metamodels are depicted, together with the analytical model for the total cost.

5.5 Robust optimization of EOQ model using Taguchi and Kriging

Now we deal with the robust simulation-optimization of the EOQ, and — as mentioned in Section 2.1.3 — we assume that a (demand per time unit) is an *unknown* constant, having a Normal (Gaussian) distribution with mean μ_a and standard deviation σ_a: $a \sim \mathcal{N}(\mu_a, \sigma_a)$. Analogously to what we did in Section 5.3, we assume $\mu_a = 8000$ ('base' value used in classic optimization), and we experiment with $\sigma_a = \{0.10; 0.50\}\mu_a$ (uncertainty about the true input parameter). We apply the two general methods that have been discussed in Section 4.2.

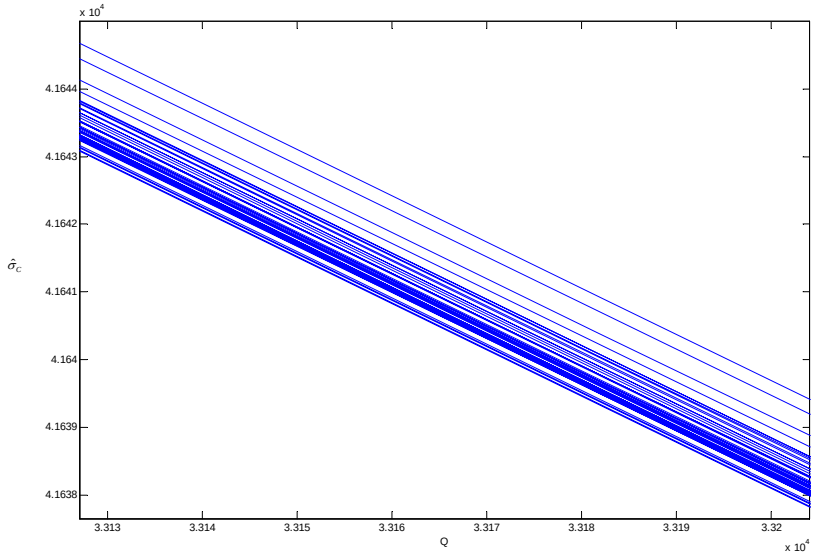


Figure 5.25: Zoom: Standard deviation of Cost, estimated through 50 macroreplicates

Kriging models for mean and standard deviation estimated from simulation I/O data

We use the same DoE built in Section 5.3, i.e. we select $n_Q = 5$ equally spaced values for the control variable Q and $n_a = 5$ values for the noise factor a according to a LHS design. We cross the two designs and run the simulation model for these 5×5 combinations of the (control and noise) inputs. The simulation I/O data are those given in Table 5.5. From this table we use (4.2.1) and (4.2.2) to estimate the mean and the standard deviation of the cost C given Q:

$$\overline{C_i} = \frac{1}{n_a} \sum_{j=1}^{n_a} C_{ij} \quad (i = 1, \ldots, n_Q)\,, \tag{5.5.1}$$

$$s_i(C) = \left[\frac{1}{n_a - 1} \sum_{j=1}^{n_a} \left(C_{ij} - \bar{C}_i \right)^2 \right]^{1/2} \qquad (i = 1, \ldots, n_Q)\,. \tag{5.5.2}$$

Using these estimators, we fit one Kriging metamodel for the mean cost and one Kriging metamodel for the standard deviation of cost. Obviously, each of these meta-

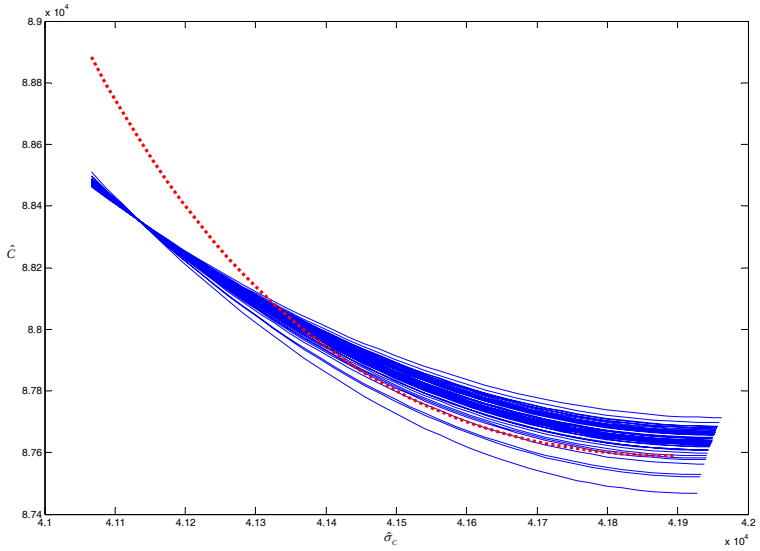


Figure 5.26: Pareto frontiers estimated from 50 macroreplicates, and true frontier (dotted curve)

models is based on $n_Q = 5$ observations. The two models are shown in Figs. 5.35 and 5.36.

We validate the two metamodels, using leave-one-out cross validation. The scatterplots in Figs. 5.37 and 5.38 display y_1 and y_2, the Kriging predictors for the mean and standard deviation. These two figures give the impression that these two Kriging metamodels are not accurate. However, the scaling of scatterplots may be misleading, so we also compute the relative prediction errors: see Table 5.13. Because of the results in this Table we accept the two Kriging metamodels.

Table 5.13: Cross-validation of Kriging metamodels for mean and standard deviation of costs

i	$\widehat{y_{1,-i}}$	$\widehat{y_{1,-i}}/C_i$	$\widehat{y_{2,-i}}$	$\widehat{y_{2,-i}}/s_i(C)$
1	84924.76	0.9922	30349.84	0.9855
2	85147.72	1.0058	30155.58	1.0040
3	85371.79	1.0073	29566.37	0.9970
4	85523.74	1.0030	29488.29	1.0020
5	85068.57	0.9894	29501.21	1.0077

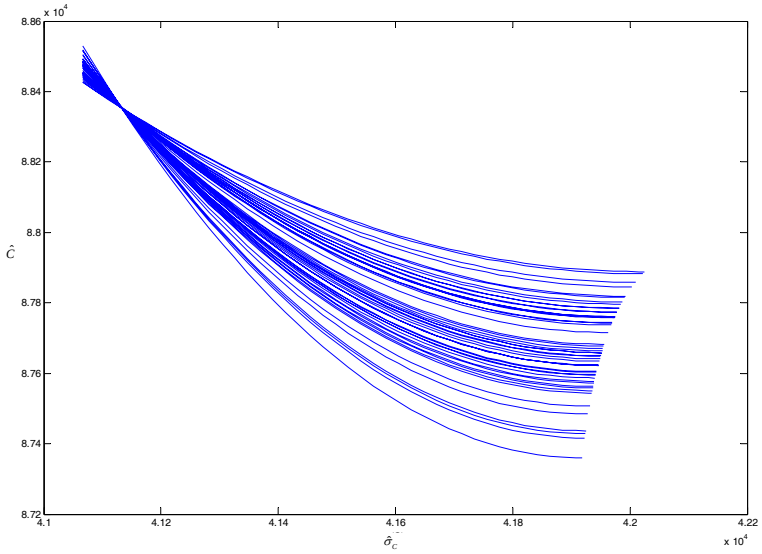


Figure 5.27: Crude sampling: replicated regression estimated Pareto frontiers

Based on these two Kriging metamodels, we find the order quantity that minimizes the mean cost, while the standard deviation does not exceed the given threshold T. We solve this constrained optimization problem. Next we vary this threshold, and find the set of optimal solutions that estimates the Pareto frontier; see Fig. 5.39.

Note: We select a range for the threshold T that differs from the range chosen to solve the corresponding problem based on RSM in Section 5.3, because selecting the same range would have resulted in an unconstrained optimization problem so the Pareto frontier would have been a single point.

We repeat the experiment using a *larger* design, namely $n_Q = 25$ values for Q. This experiment implies $25 \times 5 = 125$ simulation runs, but (5.5.1) and (5.5.2) imply that only 25 observations are used to fit the two Kriging metamodels. The resulting two metamodels for the mean and standard deviation are not reported here, since they are not so different in the shape from the previous metamodels; neverthless, we mention that these metamodels give more accurate predictions.

Single Kriging metamodel for mean and standard deviation

In Chapter 3 and in Section 5.3 — following Myers and Montgomery (2002) — we assume that a valid metamodel is the low-order incomplete polynomial metamodel in (3.2.8); that model implies the mean and variance presented in (3.2.15) and (3.2.16).

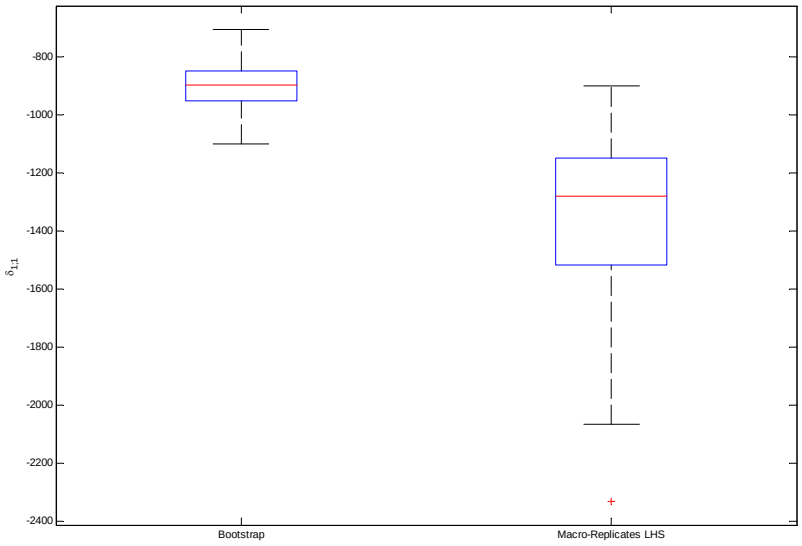


Figure 5.28: Box plot for the estimated interaction $\delta_{1;1}$

In a similar way we now assume that the generic Kriging model in (4.1.7) is a valid metamodel. That model has coefficients $\mathbf{c}$ that vary with the point to be predicted; it does not give an explicit mean and variance like (3.2.15) and (3.2.16).

To estimate the Kriging coefficients $\mathbf{c}$, we select a relatively small number of input combinations for the simulation model through a *space-filling* design for both the noise factor a and the control factor Q; i.e., we select $n_a \times n_Q = 5 \times 5 = 25$ input combinations. To avoid extrapolation when using the Kriging metamodel in the next procedure, we select $\min a_j = \max(\mu_a - 3\sigma_a, \epsilon)$, with ϵ a small positive number, and $\max a_j = \mu_a + 3\sigma_a$. This "restriction" is important for two reasons: first, this makes our results more realistic, since from a practical point of view the demand rate will never be close to $+\infty$; moreover, from a technical point of view, we need to limit the possible values for the demand rate to a given range. If not — depending on the values that have been sampled in the initial design — the Kriging model might be asked to *extrapolate* every time sampled demand rate values do not fall into the initial range.

To have an idea of what this imply, we present a plot of the initial 5×5 design, when no bounds on the demand rate are applied, compared to a bigger 25×100 set where Kriging predictions are required; we checked that 36 out of 100 values of the

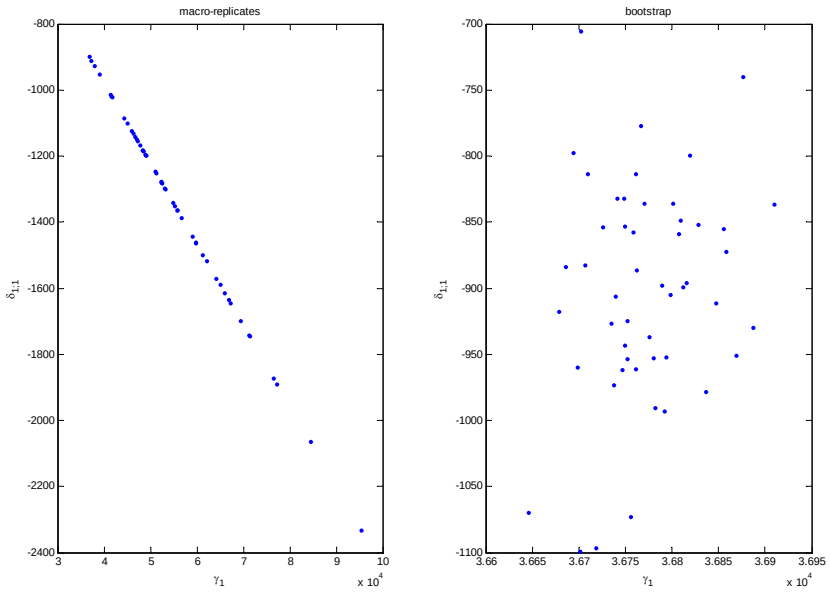


Figure 5.29: Scatterplot of $\widehat{\gamma_1}$ and $\widehat{\delta_{1;1}}$ in macro-replicates (left-hand side) and bootstrap (right-hand side)

demand rate fall outside the initial range, thus requiring $25 \times 36 = 1600$ extrapolated predictions. The two designs are shown in Fig. 5.40.

As Kleijnen and van Beers (2004) observed, Kriging may give very bad predictions in case of extrapolation, even for a valid metamodel. This is confirmed by our experiments: as it can be seen from Fig. 5.41, the approximation is not very good, except for the central region, where the points almost lay on the bisector of the quadrant. These bad results are related to the fact that — as highlighted before — some points required the Kriging metamodel to extrapolate; restricting the scatterplot only to the observations corresponding to demand values belonging to the initial range, we obtain the plot in Fig. 5.42.

These "intermediate" experiments have been mentioned only to justify the seeming limitations introduced by imposing bounds on the demand rate. Therefore, the input data obtained through our space-filling design result in Table 5.14, which also displays the corresponding simulation outputs C_{ij}.

This table gives a Kriging metamodel; applying cross validation gives the scatterplot shown in Figure 5.43.

The fitted Kriging metamodel is used in the following procedure:

Table 5.14: I/O space-filling design from EOQ simulation - Approach 2

Q	a	C
18640.03	18769.65	202575.95
31386.79	17318.21	184511.29
35155.24	5379.43	60903.82
18107.86	8468.38	93011.95
44047.65	2542.48	32724.63
42946.04	7106.25	79489.99
28846.06	5995.76	66778.71
22032.37	16418.89	176436.35
38936.17	19664.18	208542.61
26330.04	15059.09	161403.59
31836.19	8897.71	97106.37
36113.40	2397.15	30185.10
20831.92	752.85	11086.91
27613.03	3827.83	44083.69
22463.79	13876.97	149552.29
29950.48	12445.40	133932.93
17374.52	15599.21	169372.13
39415.71	11269.88	122042.20
33294.33	13377.21	143587.66
23435.46	9717.58	105666.91
41449.15	4305.93	50523.29
15168.58	10985.10	120816.72
40518.27	7487.13	83166.44
37411.17	18082.07	192232.37
24837.19	1347.09	17847.36

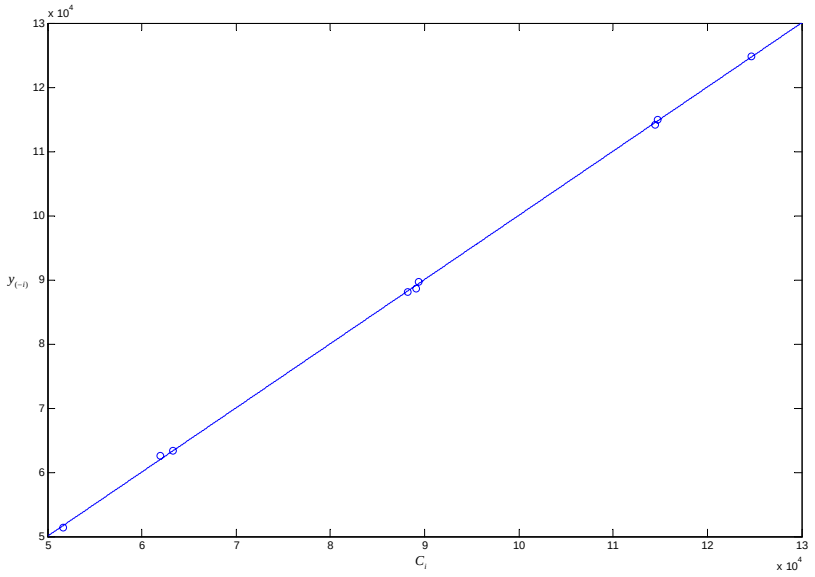


Figure 5.30: Scatterplot of the EOQ regression metamodel for CCD

1. We use LHS to sample $N_a \gg n_a$ values from the distribution of the noise variable a, and a space filling design to select $N_Q \gg n_Q$ values for the control variable Q. We select $N_a = 100$ and $N_Q = 25$. Note that the probability of exceeding the upper bound for a is negligible; nevertheless if this event occurs, we resample.

2. Then we combine the values of Step 1 into $N_Q \times N_a$ input combinations.

3. We compute the Kriging predictions $\widehat{C_{ij}}$ for the combinations of Step 2, using the Kriging metamodel estimated from the small experiment with the simulation model.

4. We use these predictions $\widehat{C_{ij}}$ to estimate the N_Q conditional means and standard deviations of the cost C:

$$\widehat{C_i} = \frac{1}{N_a} \sum_{j=1}^{N_a} \widehat{C_{i,j}} \quad (i = 1, \ldots, N_Q)\,, \tag{5.5.3}$$

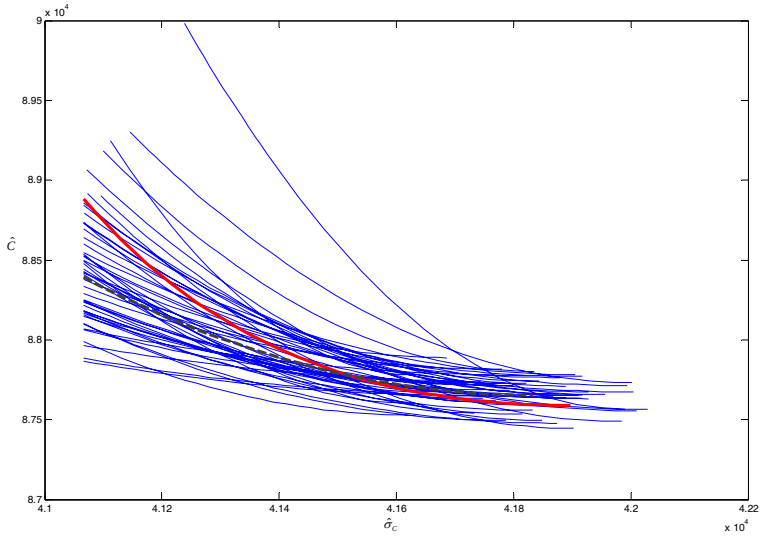


Figure 5.31: Bootstrapped Pareto frontiers, original estimated frontier (dashed curve) and true Pareto frontier (heavy curve) based on CCD

$$\widehat{\sigma_i} = \frac{1}{N_a - 1} \sum_{j=1}^{N_a} \left(\widehat{C_{ij}} - \widehat{C_i} \right)^2 \quad (i = 1, \ldots, N_Q) \,, \tag{5.5.4}$$

which are analogous to (5.5.1) and (5.5.2) but do assume a valid metamodel.

5. Finally, we fit two Kriging metamodels, namely one to the N_Q means and one to the N_Q standard deviations.

The Kriging models resulting from Step 5 are displayed in Figs. 5.44 and 5.45.

The scatterplots for these two metamodels look very good: all points are near the 45° line, so we do not display these two figures. Cross-validation gives relative prediction errors that are very close to zero; e.g., $\widehat{y_{1,(-1)}}/\widehat{C_1} = 0.9999998998$ and $\widehat{y_{2,(-24)}}/\widehat{\sigma_{24}} = 1.0000000089$.

We solve the constrained optimization problem, using `fmincon`. Next we vary the threshold T, albeit over a range that differs from the previous range to get interesting results, namely neither unconstrained nor infeasible. The resulting Pareto frontier resembles Figure 5.39 so we do not display it.

To *verify* our approach, we run the simulation model itself (not the Kriging metamodel) for the big $N_Q \times N_a = 25 \times 100$ design. Notice that we can perform this

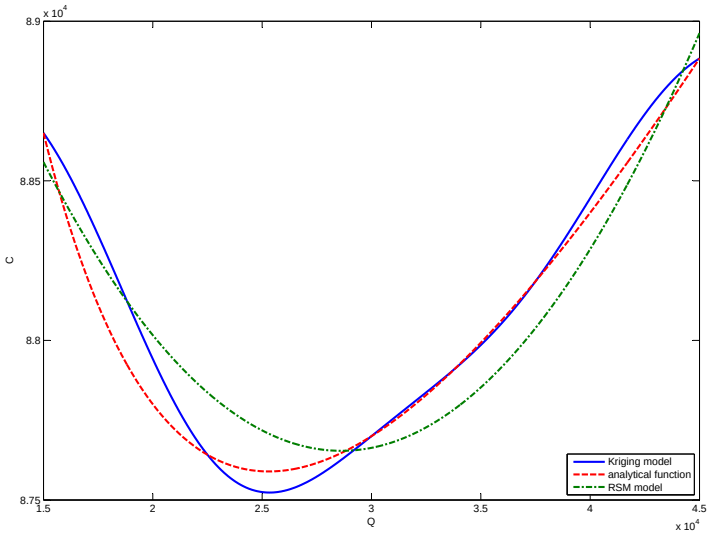


Figure 5.32: Kriging metamodel of EOQ total cost.

simulation because the EOQ simulation is inexpensive; i.e., these 2500 simulation runs take only 20 minutes on a PC with Windows XP, 1 GB RAM and 2.13 GHz. Next, we compute the sample mean and standard deviation using (5.5.1) and (5.5.2) replacing n_a and n_Q by N_a and N_Q, respectively. Then we fit Kriging metamodels to these means and standard deviations (cross-validation gives relative prediction errors very close to zero; e.g., $\widehat{y_{(-1)}}/\widehat{C_1} = 0.9999977251$ and $\widehat{y_{(-24)}}/\widehat{\sigma_{24}} = 1.0000000851$). The resulting Pareto frontier is compared with the Pareto frontier resulting from our less expensive approach that uses (5.5.3) and (5.5.4); the two curves are shown in Figure 5.46.

Bootstrapping the estimated Pareto frontier

The estimated Pareto frontier is built on the random simulation outputs C_{ij}, so we further analyze this frontier. As we explained in Section 4.2, we use *distribution-free bootstrapping*. This bootstrap gives Kriging metamodels for $\widehat{C_b^*}$ and $\widehat{\sigma_{C_b^*}}$ ($b = 1, \ldots, B$); they are shown in Figs. 5.47-5.48 for the two approaches discussed before.

We now consider the 'original' (i.e. not bootstrapped) Pareto frontier: for each threshold value T we have an optimal solution, $\widehat{C^+}$ and $\widehat{s(C)^+}$, which give the optimal order quantity $\widehat{Q^+}$. Then, we use the bootstrapped Kriging metamodels to predict the mean cost and standard deviation for each optimal decision variable $\widehat{Q^+}$. In

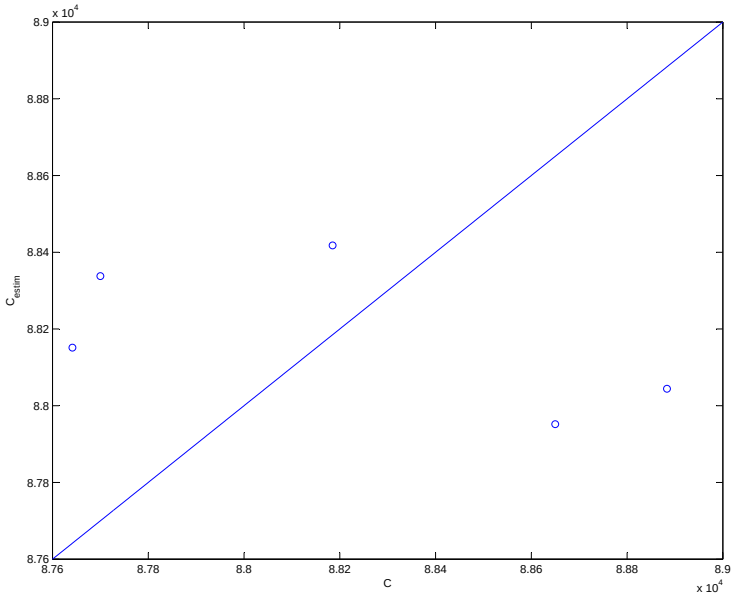


Figure 5.33: Scatterplot for Kriging metamodel of EOQ total cost.

order to show the results through clear figures, we select only two values of the threshold T, corresponding to opposite situations, when the managers are either risk-averse (small T) or risk-seeking (high T); the plots obtained are shown in Fig. 5.49, where the 'star' denotes a point belonging to the 'original' Pareto frontier and the 'dots' are the bootstrapped predictions. On the same plot, it is also depicted an ellipse representing a 90% confidence region for the bootstrapped optimal solutions, according to the approach proposed by Kleijnen and Gaury (2003) and outlined in Sect. 5.3. We point out that the red star is not expected to be inside the elliptical region, since such ellipse is computed based on the bootstrapped estimates and does not represent a confidence region for the optimal solution belonging to the original Pareto frontier (i.e. the red star). We further observe that we compute the optimal solution represented by the red star through solving the optimization problem in Eq. 5.3.7, given T, adopting a Kriging metamodel based on specific estimates of its parameters; the elliptical confidence region, instead, accounts for possible variability in the metamodel parameters. To confirm that the hypothesis of bivariate normal output is satisfied, we also show the chi-square plot in Fig. 5.50; we depict only the one corresponding to $T = 40000$, since the plot associated to $T = 41560$ does not differ so much.

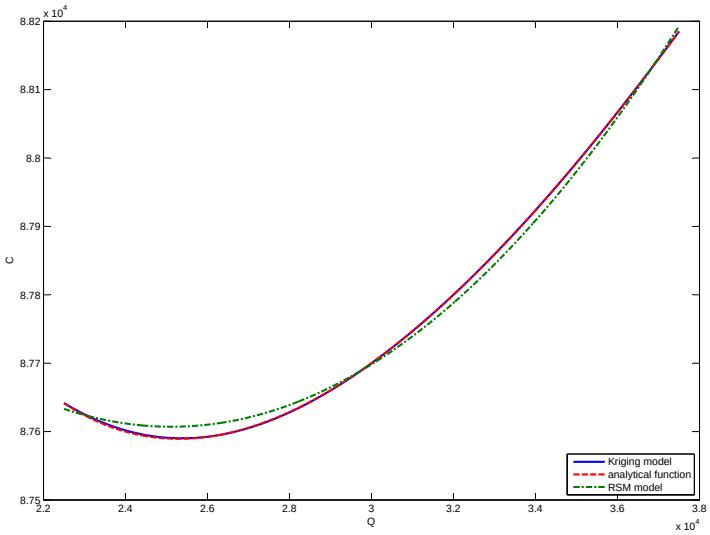


Figure 5.34: Kriging metamodel of EOQ total cost for smaller Q-range.

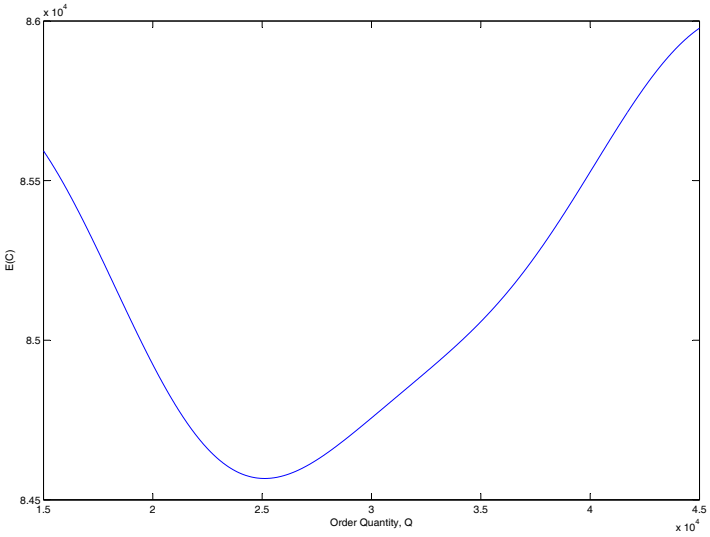


Figure 5.35: Kriging metamodel for the mean cost.

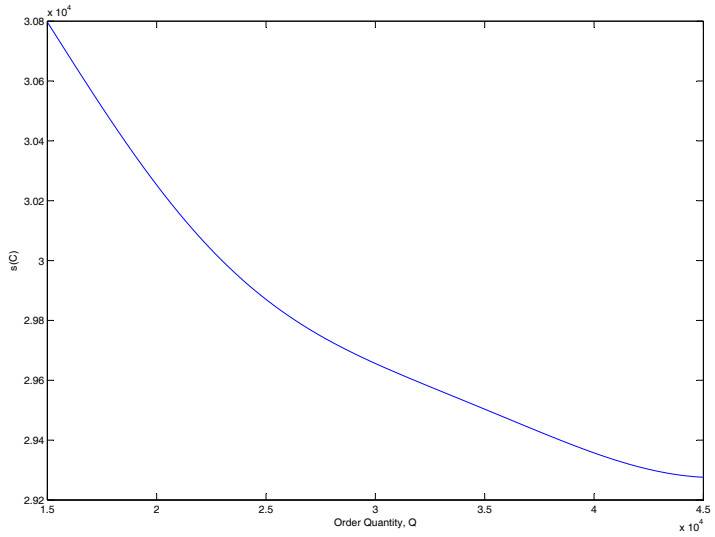


Figure 5.36: Kriging metamodel of the standard deviation of the cost.

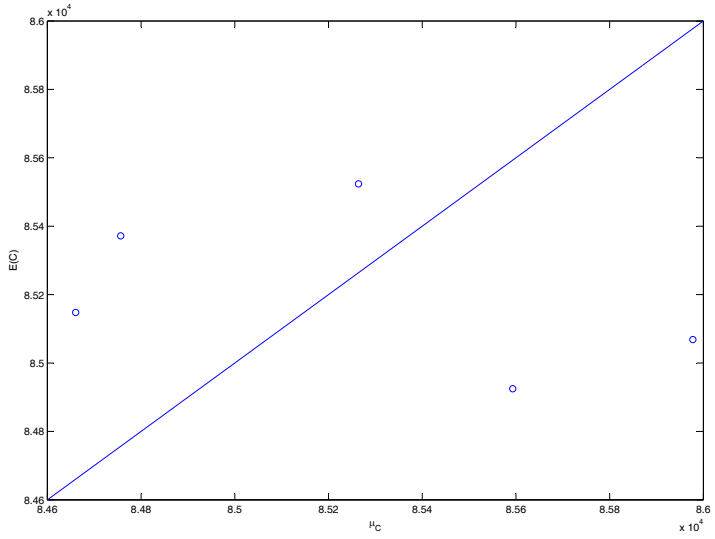


Figure 5.37: Scatterplot of Kriging metamodel for mean cost.

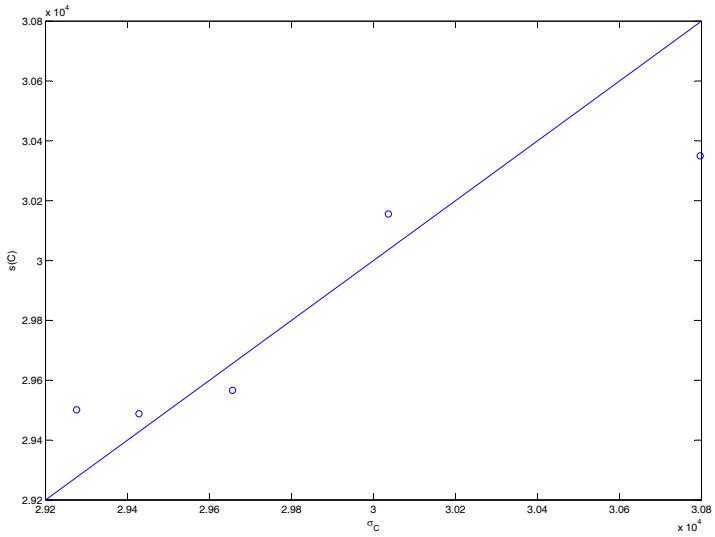


Figure 5.38: Scatterplot of Kriging metamodel for standard deviation of the cost.

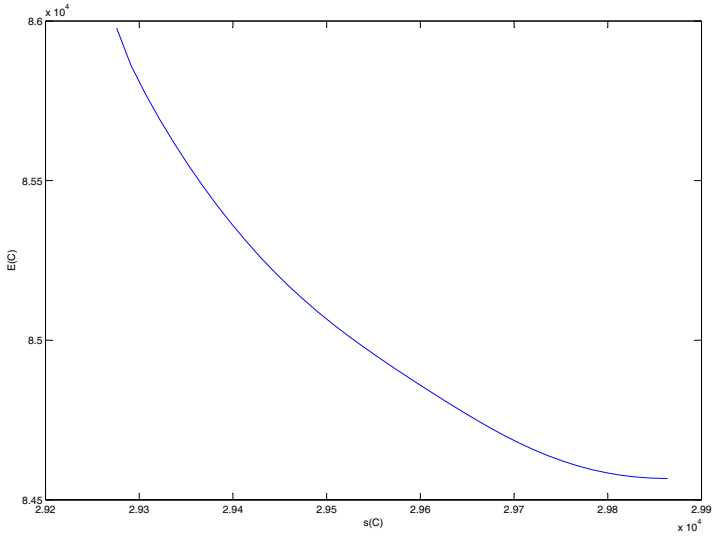


Figure 5.39: Estimated Pareto frontier for EOQ based on Kriging metamodels.

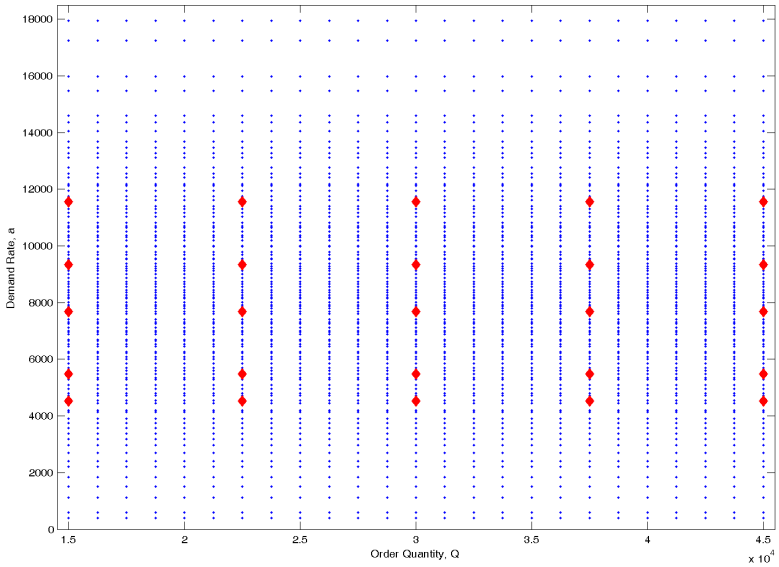


Figure 5.40: Initial design (red diamonds) and points for predictions (blue dots).

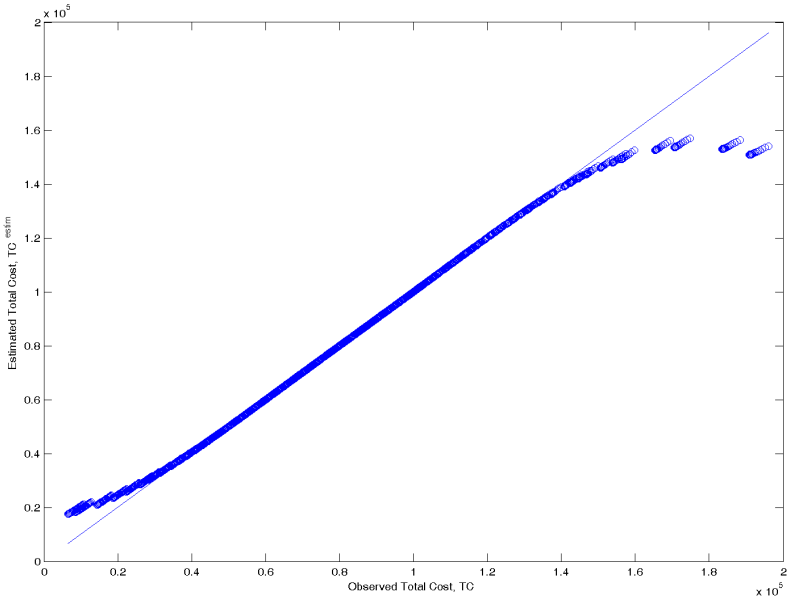


Figure 5.41: Scatterplot of Kriging predictions for total cost versus observed values.

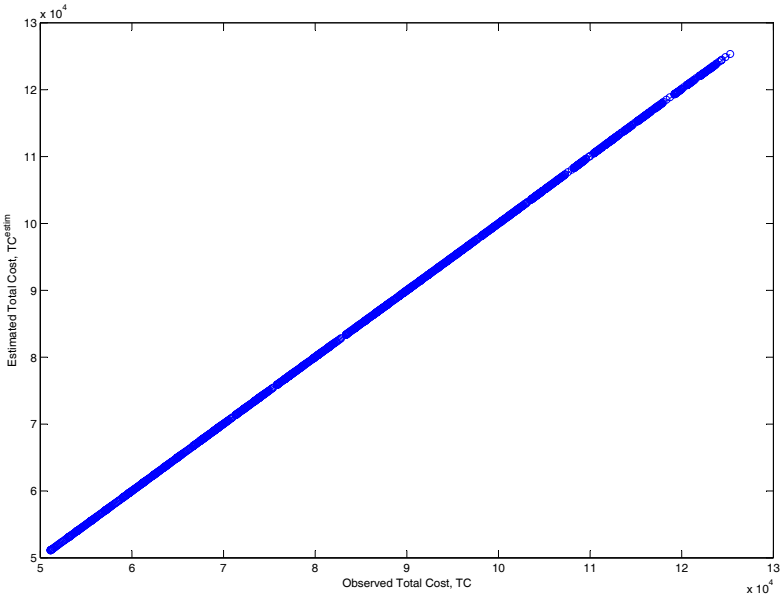


Figure 5.42: Scatterplot of Kriging predictions for total cost versus observed values (removing extrapolations).

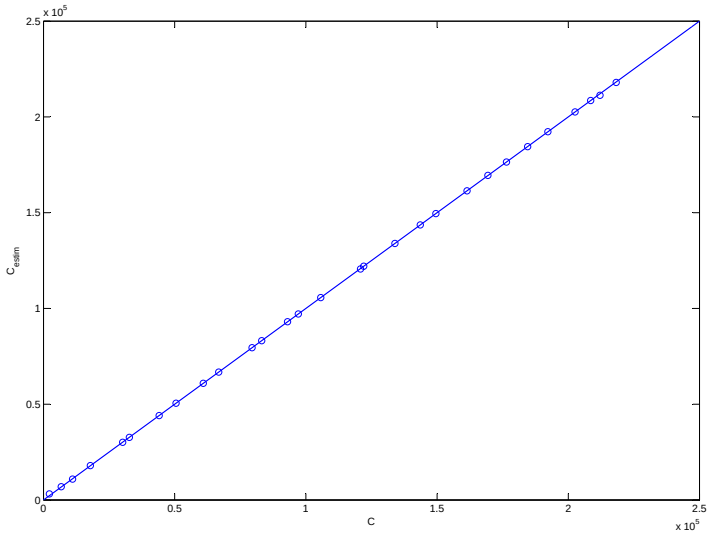


Figure 5.43: Scatterplot for Kriging metamodel of EOQ total cost - Approach 2.

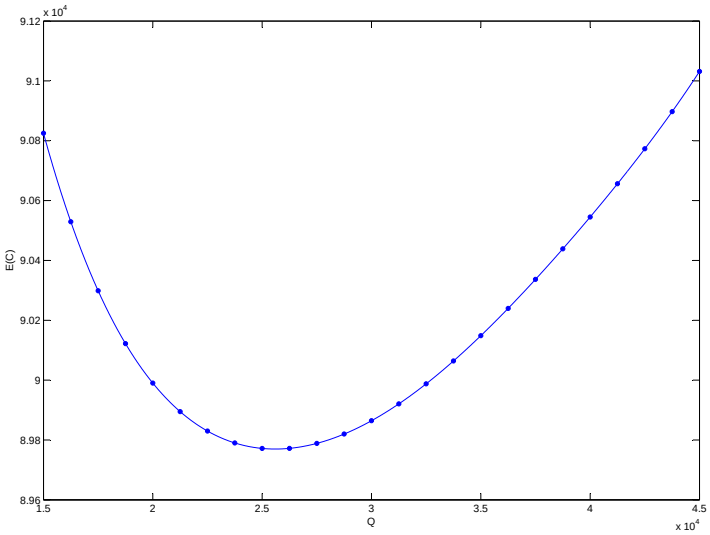


Figure 5.44: Kriging metamodel for mean cost based on Kriging predictions

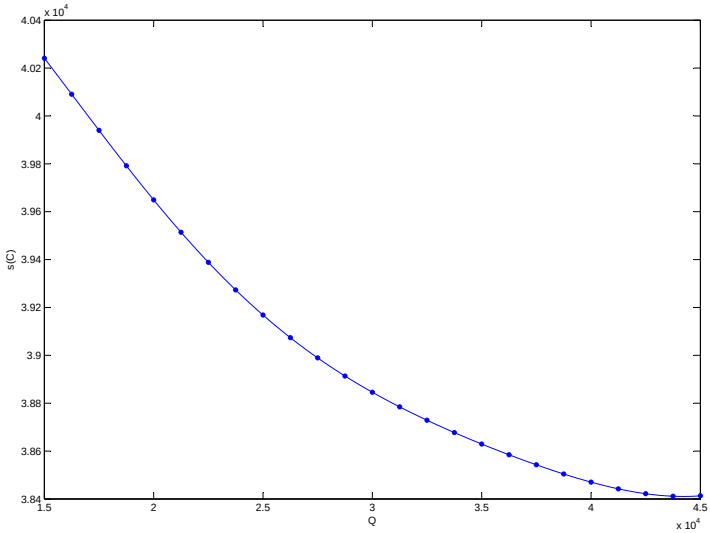


Figure 5.45: Kriging metamodel for the cost's standard deviation based on Kriging predictions

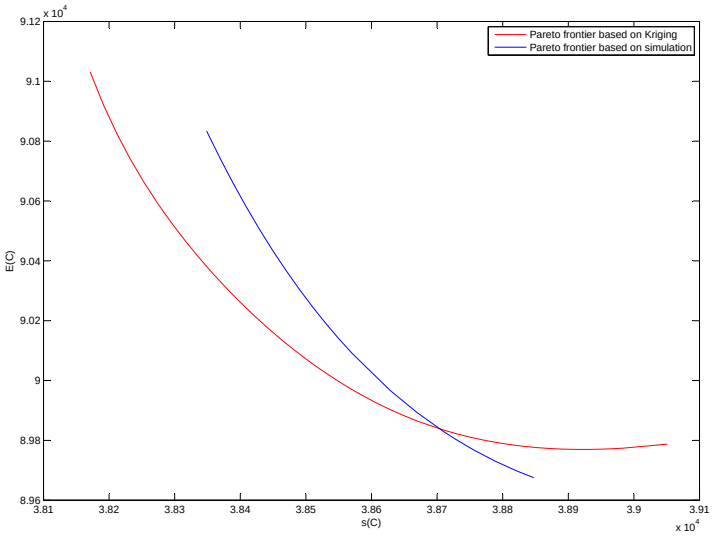


Figure 5.46: Estimated Pareto frontiers based on simulation (blue curve) and on Kriging metamodel (red curve).

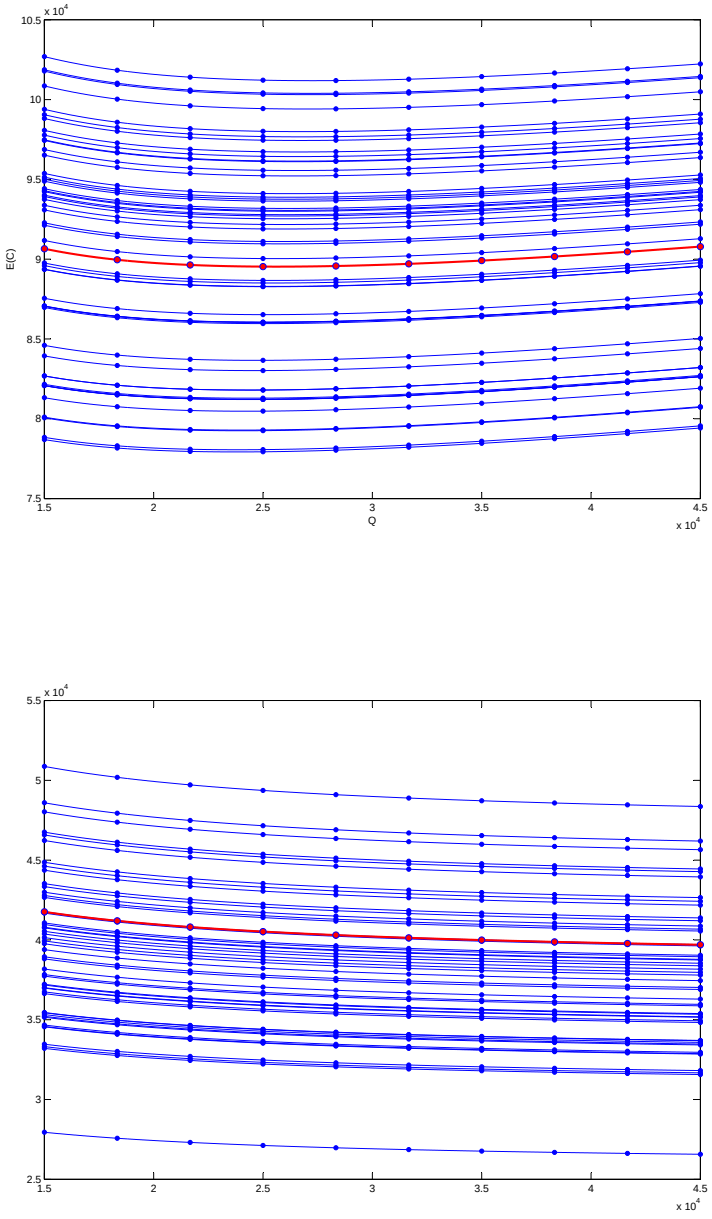


Figure 5.47: Bootstrapped Kriging metamodels for the estimated mean (above) and standard deviation (below) of the total cost — Approach 1.

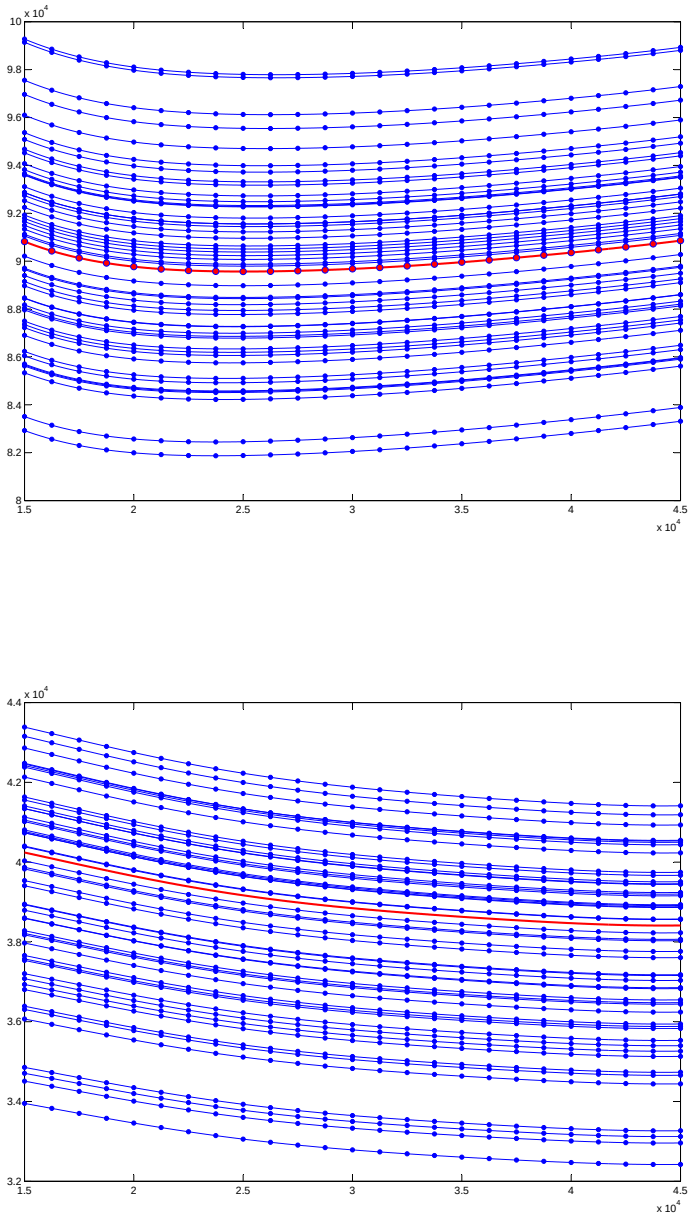


Figure 5.48: Bootstrapped Kriging metamodels for the estimated mean (above) and standard deviation (below) of the total cost — Approach 2.

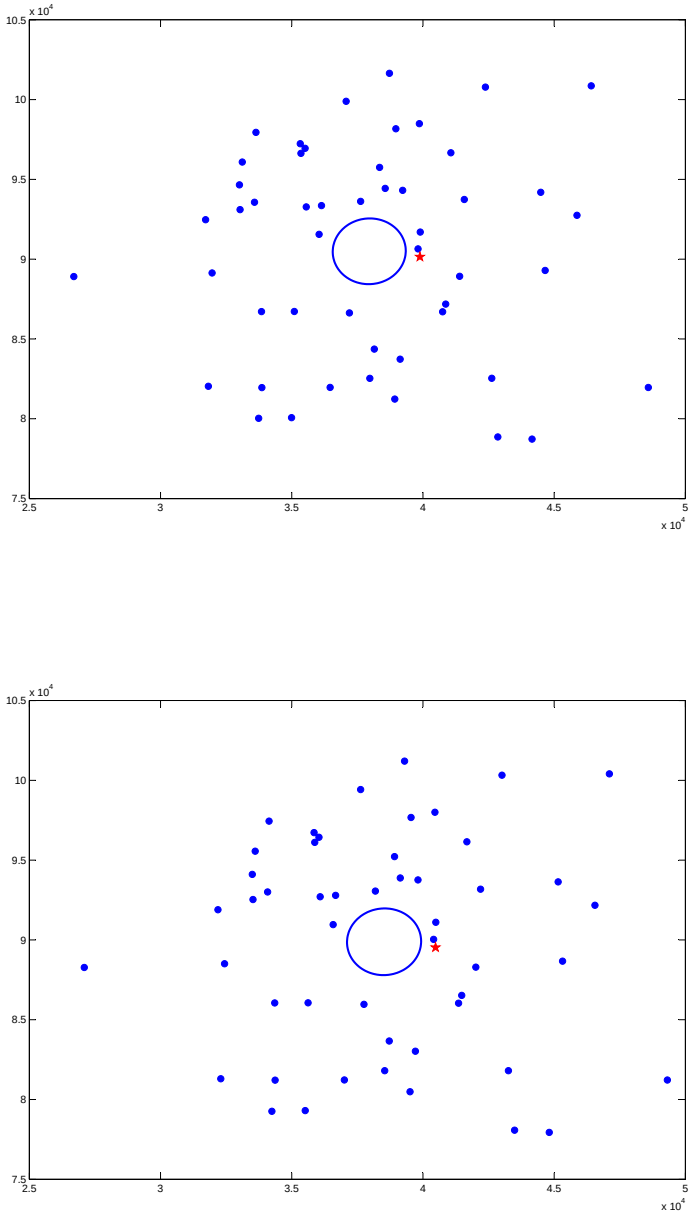


Figure 5.49: Bootstrapped estimations of mean cost and standard deviation and 90% confidence region, based on Kriging metamodels, for $T = 40000$ (above) and $T = 41560$ (below) — Approach 1.

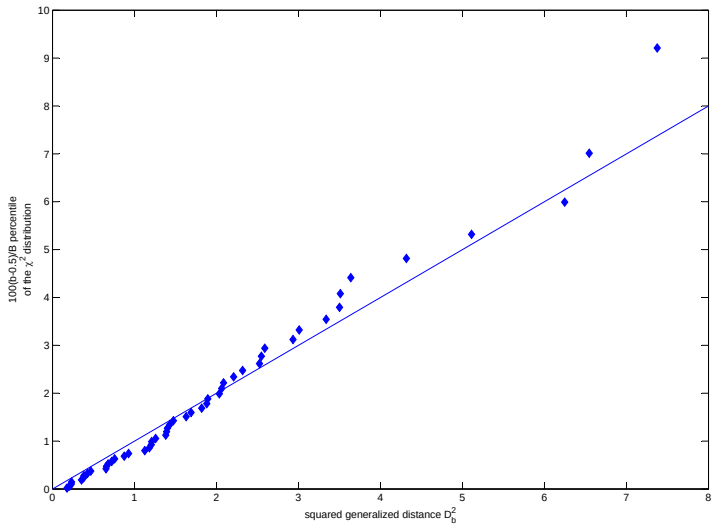


Figure 5.50: Chi-square plot of the bootstrapped estimates of mean cost and standard deviation, based on Kriging metamodels, for $T = 40000$ — Approach 1.

It should be pointed out that the results referred to as Approach 1 are not directly obtained to what discussed before; in fact, in order to apply our bootstrap procedure, we need to take a bigger number of observations for each value of the order quantity. Therefore, we repeated all the experiments in Approach 1 taking $n_a = 25$ values for the demand rate. The results for a small and a high threshold T based on Approach 2 are depicted in Fig. 5.51, together with the corresponding 90% confidence regions. The chi-square plot referred to $T = 38580$ is shown in Fig. 5.52, confirming that the hypothesis of bivariate normality is acceptable.

We also experimented with smaller uncertainty on the demand rate, namely choosing $\sigma_a = 0.10\mu_a$. We performed all the steps described so far, both for Approach 1 and for Approach 2; although we will not show all the details on the results we obtained, we remark some implications of reducing the uncertainty on the noise factors. First of all, it is reflected in a reduced variability in the metamodels for both the expected total cost and its standard deviation: looking at Figs. 5.53-5.54 it is evident — especially focussing on the scale for the y-axis — that the spread of the bootstrapped Kriging metamodels is reduced, for both the approaches.

The described behaviour is also confirmed by computing the bootstrapped estimations of the mean and standard deviation of the total cost corresponding to the optimal Q-values producing the 'original' Pareto frontier. Again, we considered two

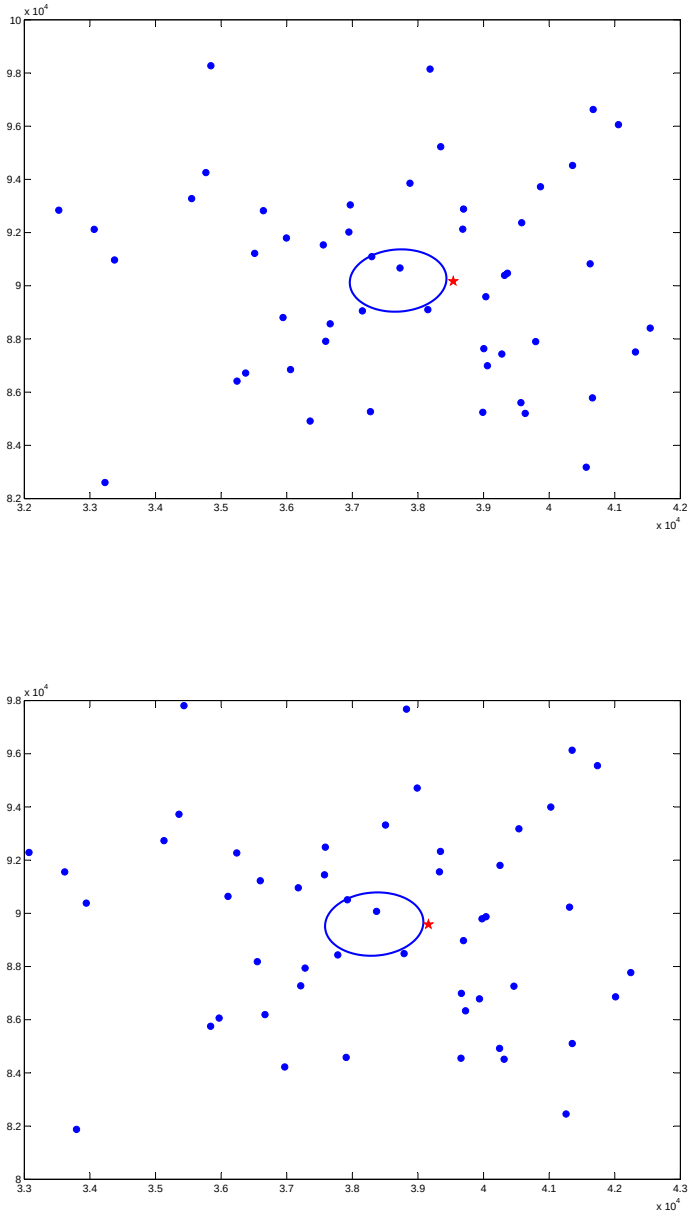


Figure 5.51: Bootstrapped estimations of mean cost and standard deviation and 90% confidence region, based on Kriging metamodels, for $T = 38580$ (above) and $T = 40060$ (below) — Approach 2.

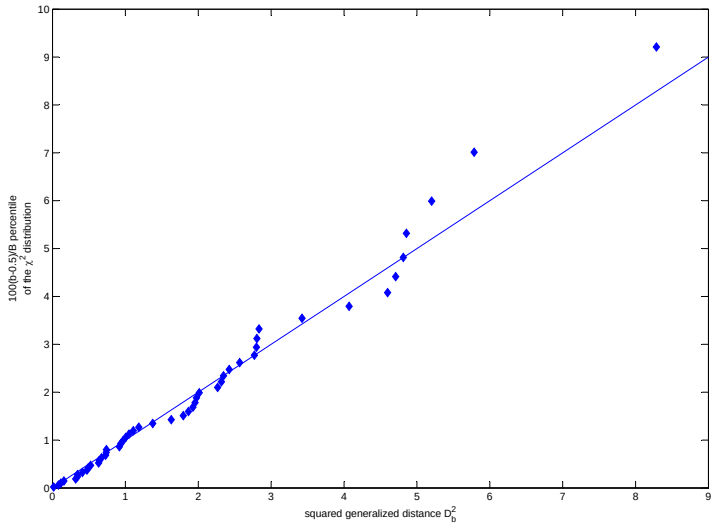


Figure 5.52: Chi-square plot of the bootstrapped estimates of mean cost and standard deviation, based on Kriging metamodels, for $T = 38580$ — Approach 2.

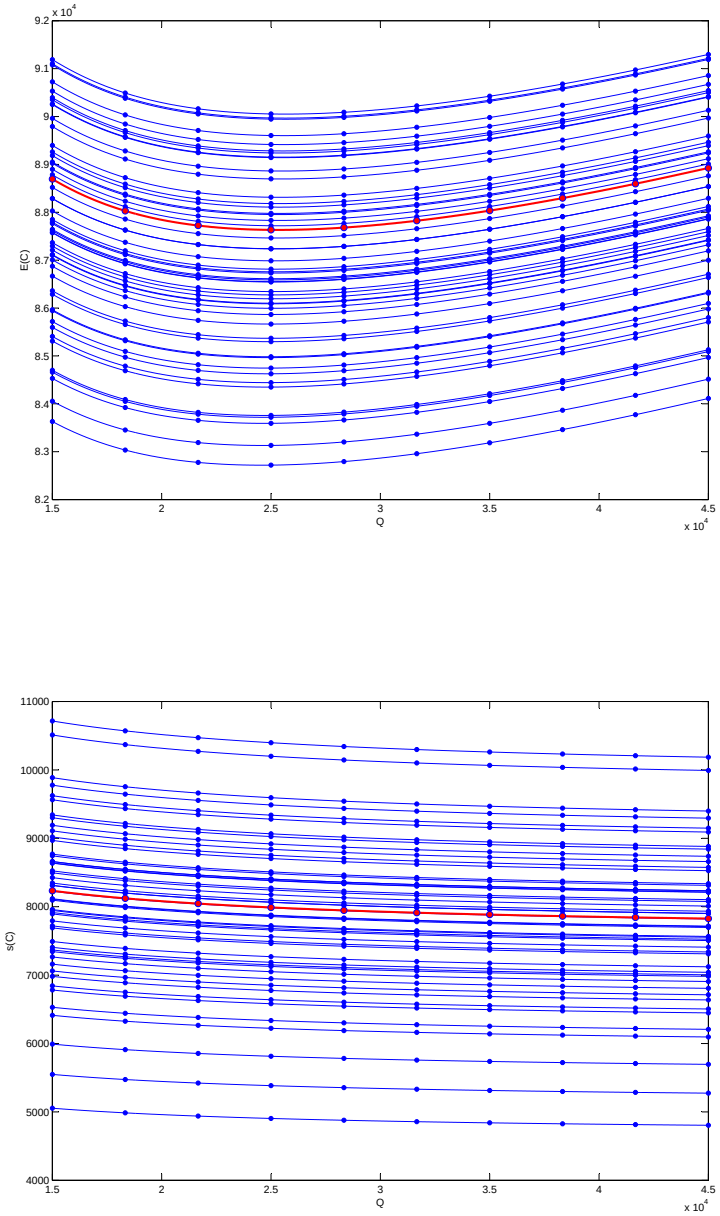


Figure 5.53: Bootstrapped Kriging metamodels for the estimated mean (above) and standard deviation (below) of the total cost, with smaller uncertainty — Approach 1.

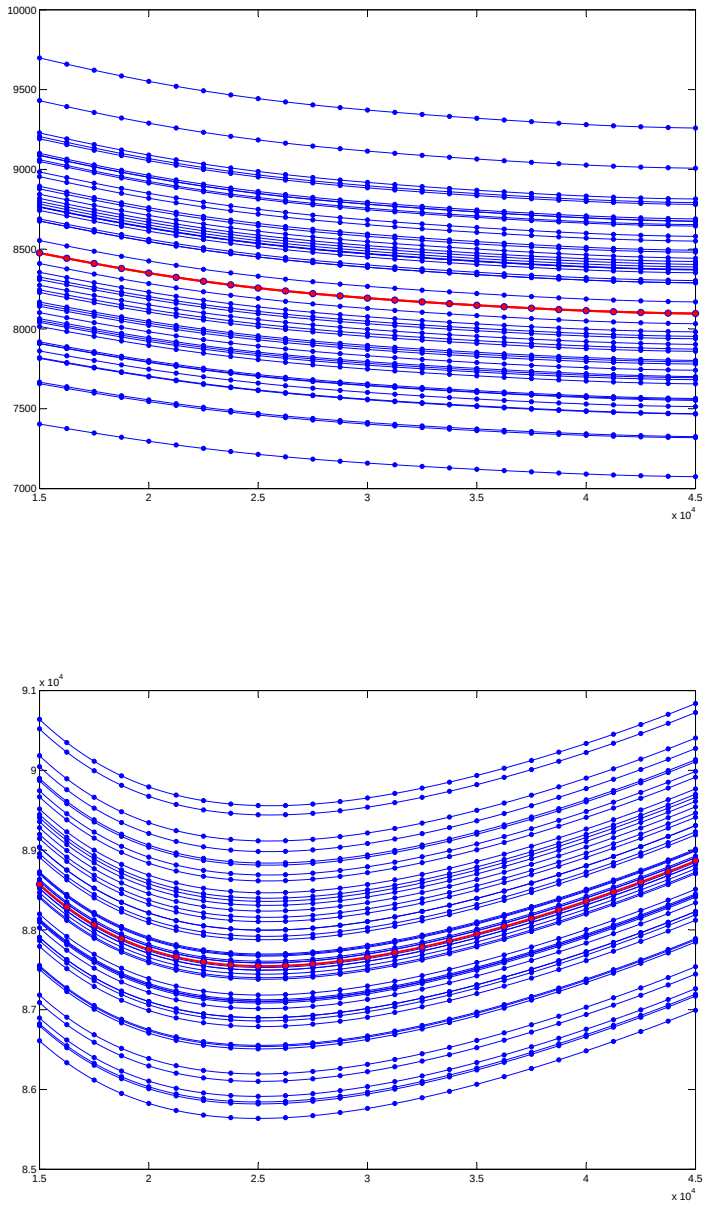


Figure 5.54: Bootstrapped Kriging metamodels for the estimated mean (above) and standard deviation (below) of the total cost, with smaller uncertainty — Approach 2.

extreme cases for both the approaches, selecting a small and a big value for the threshold T; the plot we obtained are depicted in Figs. 5.53 and 5.54, respectively.

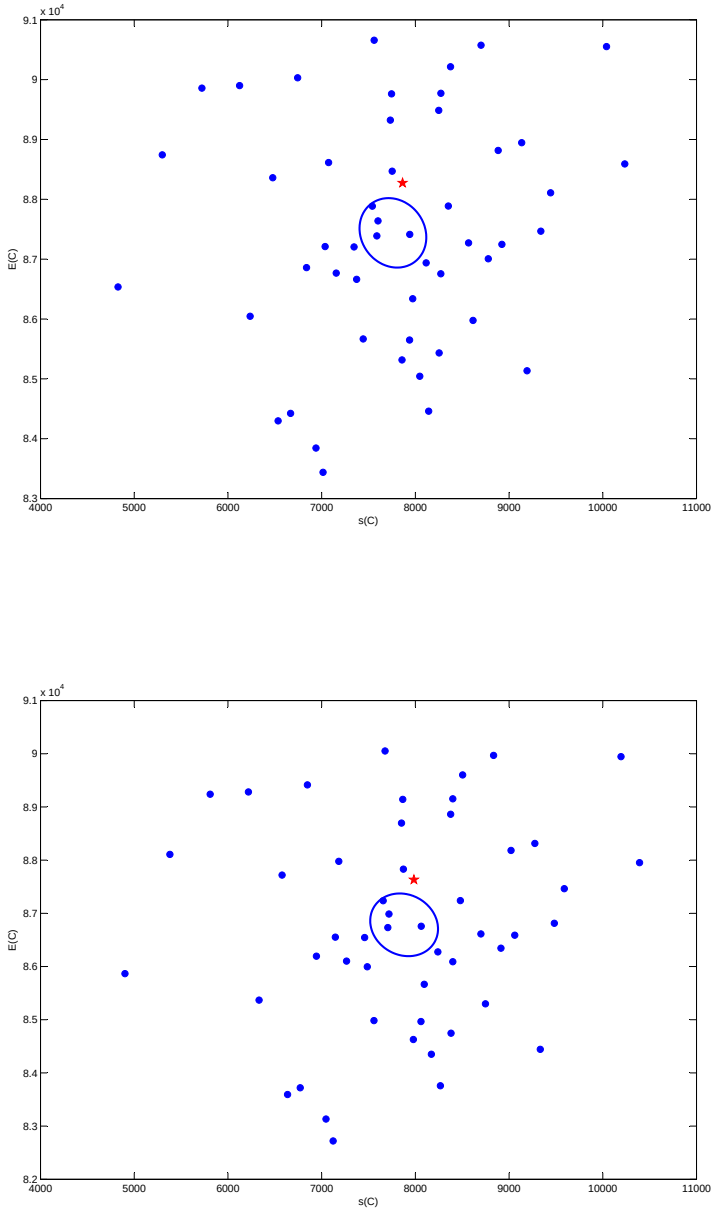


Figure 5.55: Bootstrapped estimations of mean cost and standard deviation and 90% confidence region, based on Kriging metamodels, for $T = 7860$ (above) and $T = 8200$ (below) — Approach 1.

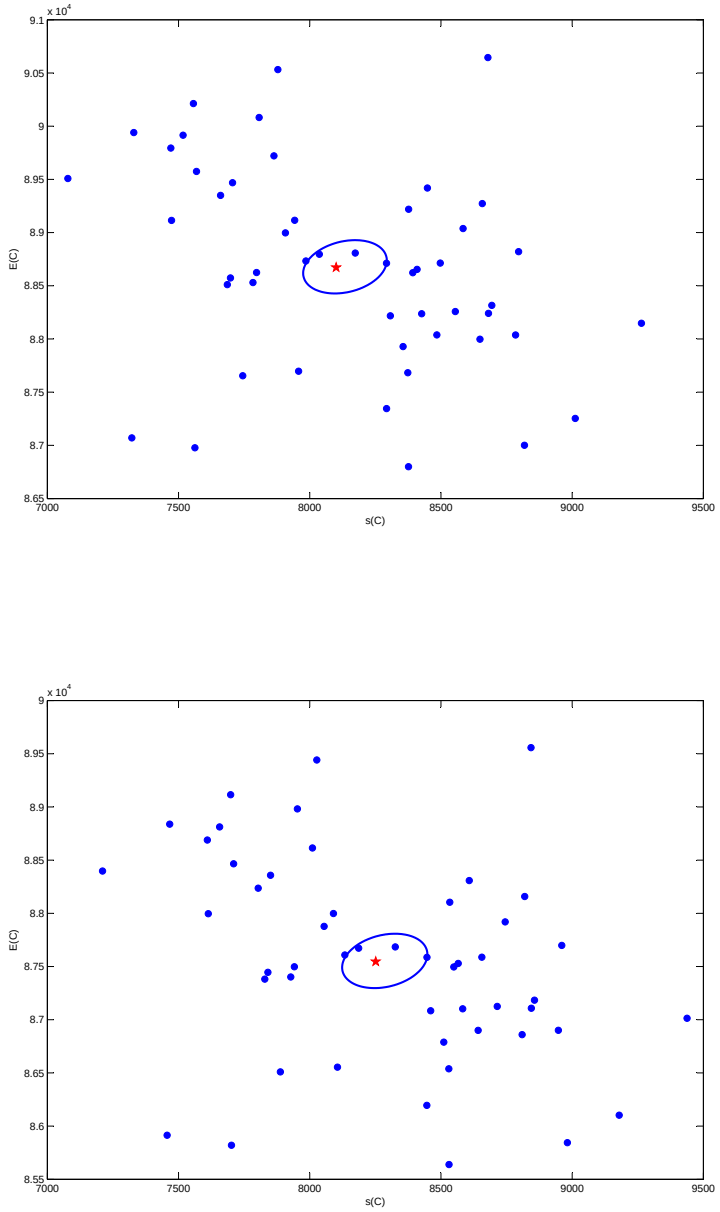


Figure 5.56: Bootstrapped estimations of mean cost and standard deviation and 90% confidence region, based on Kriging metamodels for $T = 8100$ (above) and $T = 8460$ (below) — Approach 2.

Chapter 6

(s, S) Inventory Model

The (s, S) inventory model has been chosen as another example to which apply the proposed methodology. The model we refer to has been already described in Section 2.2, so we will omit all the details on the case study description; nevertheless, we point out that this model is not a deterministic model, as the EOQ model is, being instead stochastic. This implies that it includes some random input components (namely, the demand size and the orders' lead times) which cause the outputs of interest to be random variables themselves, which therefore have to be treated as estimates of the true responses of the model.

The remainder of the chapter is structured as follows: first some issues related to the implementation of the simulation model are discussed in Section 6.1, then the experimental results on the (s, S) model are presented. More specifically, Section 6.2 concerns the results for the classic simulation-optimization approach using either regression or Kriging metamodels; the results of applying the classic simulation-optimization approach to the (s, S) inventory model with a service level constraint are discussed in Section 6.2.1. Then, Section 6.3 and 6.4 describe the results for the robust simulation-optimization, according to what discussed in Chapters 3 for the RSM and 4 for Kriging.

6.1 Modeling Implementation Issues

Before presenting the computational experiments we have carried out and the results we obtained, it is worth to highlight two modeling implementation issues that have to be dealt with. More specifically, one is related to the particular variant of the model that we decide to consider and the other concerns the use of a variance-reduction technique, namely Common Random Numbers (CRN). Each of them is discussed in the following subsections.

6.1.1 Order Crossing

As discussed in Section 2.2, we drop one major assumption in the analysis of the (s, S) inventory system, which considers the probability distribution of the lead times such that ordered items are received in the same sequence as the orders have been placed. This implies that our model allows for order crossing over the periods: although this aspect makes the model more realistic — e.g. when there are multiple suppliers — the mathematical analysis of the model becomes more complicated and there are no alternatives but simulation. During the verification process — which is always a necessary step to perform before going on with any procedure based on the simulation model (see Law, 2007) — we explicitly check that order crossing indeed occurs in some runs.

The randomness in the lead times implies that no analytical solution exists for this problem; therefore, we compare our results for the classic optimization with the outcome of OptQuest, a well-known commercial software by OptTek Systems, which treats the simulation model as a black box; its algorithms are based on tabu search, scatter search, integer programming and neural networks (Rockwell Automation, 2004). It should be pointed out, however, that this solver takes advantage of directly interacting with the underlying — possibly costly — simulation model, whereas we solve the optimization problem using a metamodel: the latter approach is usually less time-consuming, but the former can provide more accurate results.

6.1.2 Replications and Common Random Numbers

Since we are now working with a stochastic model, *replications* are needed (Law, 2007): this means that we make several independent runs, each using the same initial conditions but separate sets of random numbers. This makes the simulation results from different runs independent from each other, given the same scenario or input combinations. When we simulate a different scenario, in order to have a "fair" comparison of the corresponding performance, we should keep "similar" environmental conditions so that any differences observed in the performance of the system can be directly connected to the different input combinations rather than justified through possible changes in the environmental factors. The environmental factors we are referring to in the (s, S) inventory model are the demand size and the lead time. This desirable condition might be achieved through the use of Common Random Numbers (CRN). We will not detail this technique, for which we refer to (e.g.) Law (2007), but we want at least to justify the use of this technique and the way we apply it to our experiments. At this aim, consider two possible scenarios, which respectively lead to the performance denoted as w_{1j} and w_{2j} on the j-th independent replication, and we are interested in comparing their mean responses, thus estimating $\xi = E(w_{1j}) - E(w_{2j})$. If we have m replications of each scenario and denote $v_j = w_{1j} - w_{2j}$, for $j = 1, \ldots, m$, then $E(v_j) = \xi$ so

$$\overline{v} = \frac{\sum_{j=1}^{m} v_j}{m}$$

is an unbiased estimator of ξ. Since the v_j's are i.i.d. random variables,

$$Var(\overline{v}) = \frac{Var(v_j)}{m} = \frac{Var(w_{1j}) + Var(w_{2j}) - 2Cov(w_{1j}, w_{2j})}{m}.$$

If the simulations of the two different scenarios are done independently, i.e. using different random numbers, then w_{1j} and w_{2j} will be independent, so $Cov(w_{1j}, w_{2j}) = 0$. On the other hand, if we could introduce a positive correlation between w_{1j} and w_{2j} then we will get $Cov(w_{1j}, w_{2j}) > 0$, which reduces the variance of the estimator $\overline{v}$. This is exactly the purpose of using CRN, which let us obtain such positive correlation by using the same random numbers to simulate all scenarios. Notice that this is possible because of the deterministic reproducible nature of random number generators.

An important topic in using CRN concerns synchronization of the random numbers across the different scenarios on a particular replication: this means that we should use a specific random numbers for the same purpose in all the scenarios. Losing synchronization among the streams can produce very bad results, thus destroying any potential benefit of using CRN. In order to achieve this purpose, we adopt *stream dedication*, i.e. we use one random number stream for each random variable involved in the simulation model: in our (s, S) model we will use one stream for the demand size and another independent separate stream for the lead time.

A final remark is related to the correctness of achieving complete synchronization of all sources of randomness: referring to the (s, S) inventory model, it appears reasonable and highly desirable for what discussed so far that different input combinations (s, S) will face *exactly* the same demand in each period n of the simulation run, within the same replication. We probably would like to do something similar for the order lead time. However, due to the different values of s and S, orders will generally be placed at different times, so the total number of orders placed during the N observed periods will also differ. As a consequence, it might be difficult to keep the two streams synchronized throughout the N periods; moreover, this can be even nonsense, so it can be better to just generate the order lead time values independently across the different input combinations. However, we will use a common stream of random numbers for the lead time, since we expect it not to 'damage' our results, though being aware that it will not achieve the purpose of variance reduction.

As for the number of replications to run for each input combination, we keep it fixed in our experiments and take $m = 10$. The choice of this value is supported by the verification that the half-length $\ell_h(m, \alpha)$ of the $100(1 - \alpha)$ percent confidence interval for the average simulation outputs $w_h(\mathbf{d}_i)$ is within $\gamma\%$ of the true mean for all outputs h and for all input combinations $\mathbf{d}_i$. Therefore, based on Law (2007), if

$$\forall h : \frac{\ell_h(m, \alpha)}{|\overline{w_h(\mathbf{d}_i)}|} \leq \frac{\gamma}{1 + \gamma} \tag{6.1.1}$$

then m realizes the relative precision γ with $0 < \gamma < 1$. The half-length of the

$100(1-\alpha)$ percent confidence interval is given by

$$\ell_h(m,\alpha) = t_{m-1;1-\alpha/2}\sqrt{\frac{\widehat{var(w_h(\mathbf{d}_i))}}{m}} \tag{6.1.2}$$

where $t_{m-1;1-\alpha/2}$ denotes the $(1-\alpha/2)$ quantile of the t_{m-1} distribution. The average simulation output is given by Eq. 3.2.21, and the estimated variance is given by Eq. 3.2.24.

6.2 Classic Simulation-Optimization using metamodels

In this section we apply a classic simulation-optimization approach based on regression metamodels first and Kriging metamodels later on, to determine the combination of controllable input variables leading to the minimum expected average total cost. At this stage we only take the *aleatory* uncertainty into account: we restrict our attention on the stochastic nature of the model due to some random variables (namely, demand size and order lead time), but we assume that the parameters guiding the corresponding probability distribution are known and equal to some constant values (see also Kleijnen, 1983b, 1994). Notice that here — despite the uncertainty on the demand size and on the lead time — we cannot provide a robust solution for the optimization problem, since the source of uncertainty is intrinsic to the system itself and cannot be removed.

Although interested in determining the optimal policy in terms of the reorder point s and the order-up-to level S, we slightly reformulate our model such that the variables to optimize are the reorder point s (as before) and the difference $d = S-s$. It is easily understandable the reason for introducing this transformation when thinking of a design matrix we are going to create: in fact, if we generate a DoE in the design space of (s, S), we might obtain *infeasible* combinations whenever $S < s$, which is clearly nonsense; generating a design on s and d avoids this problem.

We refer to Section 3.1.1 for the description of the steps we will perform; in particular, see Fig. 3.1. When fitting a Kriging metamodel, the scheme we follow is basically the one sketched in Fig. 3.1, apart for the specific metamodelling technique; moreover, when working on Kriging, step 4 in Fig. 3.1 — devoted to test the regression coefficients for significance — cannot be performed.

As a preliminary experiment, assuming not to have any information about the most promising experimental area, we start from a large experimental area, namely taking $500 \leq s \leq 1250$ and $1 \leq d \leq 500$: then we simulate 25 levels for s and 12 levels for d over this region, thus obtaining a bi-dimensional grid of 300 input combinations.

It can be useful for further analyses to specify the output values obtained by simulating the 300 input combinations of s and d; the I/O data are represented in Table 6.1.

Table 6.1: I/O data for the (s, S) inventory model, using Full Factorial DoE with 300 input combinations

		$d = S - s$									
		1	46.3636	91.7273	137.0909	182.4545	227.8182	273.1818	318.5455	363.9091	409.2727	454.6364	500
s	500	648.8281	661.2877	674.1526	687.0192	699.2021	686.7426	659.5936	639.2441	625.0094	618.8176	616.9391	621.5366
	531.8182	658.7361	670.3716	683.2036	695.8829	695.4632	668.1785	644.5402	627.6503	616.3174	612.8163	612.8531	618.468
	563.6364	667.7662	679.5428	692.3241	702.2186	677.2021	652.0065	631.7647	618.0014	611.9743	609.0122	612.1771	618.236
	595.4545	677.5808	688.6897	701.1988	689.1941	660.4217	639.6004	622.2852	612.9903	607.8811	608.5428	614.3209	622.3152
	627.2727	688.214	698.5458	702.1714	676.668	646.5258	627.8802	616.3031	610.5189	607.9	611.1022	619.3281	627.8151
	659.0909	701.9249	712.8668	697.6933	664.0543	636.6351	621.7528	612.3152	610.393	613.2686	618.9519	626.9623	637.2606
	690.9091	722.4269	740.7666	693.8366	654.8759	629.3396	619.0847	612.605	614.223	618.4655	626.5836	637.8308	648.9742
	722.7273	1022.994	764.8133	692.0025	650.9655	628.0665	618.3219	616.6041	620.0349	626.4028	638.9484	649.812	662.9522
	754.5455	1459.042	789.8041	692.7023	650.0738	629.2434	620.9205	623.4504	629.3621	637.7742	650.8991	662.4721	677.5698
	786.3636	1918.078	809.7967	696.6753	652.5878	633.4554	629.4911	632.6803	639.2371	650.3161	664.9437	679.9729	694.369
	818.1818	2455.587	832.6622	704.7335	660.0522	642.0325	638.5171	645.5799	653.3267	666.0017	681.5822	696.4464	714.2693
	850	2961.55	854.7526	713.9279	668.834	654.0835	652.1394	658.5897	669.053	682.3922	698.4965	716.2496	734.0972
	881.8182	3575.944	876.5349	727.3078	681.9485	667.6538	667.2932	676.0402	686.6761	701.1155	717.4732	736.0597	755.4954
	900	3993.916	887.5806	732.3692	686.2759	673.7158	673.937	682.0878	695.0962	709.7907	726.3743	745.9409	765.5768
	931.8182	4728.66	914.1589	749.4045	703.2552	691.5228	692.2566	701.0988	715.9955	731.8354	748.0263	766.8644	788.4985
	963.6364	5428.914	941.6413	768.2229	721.6248	710.1609	712.9522	723.5265	737.8662	755.1791	771.9124	790.6773	812.1736
	995.4546	6139.146	968.1903	788.9015	742.5272	731.2908	734.0849	745.3526	760.5872	778.1534	796.0548	815.0865	836.9902
	1027.273	6922.445	994.9101	811.2945	764.4432	754.03	758.7611	769.6411	783.7204	803.0343	821.5468	840.4387	862.4875
	1059.091	7873.691	1023.9	835.3252	787.5924	777.9703	782.8258	794.5846	810.0526	828.4839	847.0035	867.5889	888.2494
	1090.909	8794.243	1052.39	860.1276	812.5923	802.663	808.0607	820.3119	836.4119	854.9319	874.6644	894.662	915.3951
	1122.727	9684.689	1080.109	886.2223	839.6326	828.8113	834.7614	847.4301	863.6736	883.2478	902.2558	921.7882	943.5514
	1154.546	10748.13	1111.427	914.0436	865.7559	857.09	862.4267	874.5994	891.1556	909.8342	930.065	950.6786	971.8834
	1186.364	11519.44	1141.508	941.4303	892.9208	884.0321	889.8884	902.8934	919.5095	938.7042	958.8207	979.3686	1001.067
	1218.182	12759.24	1167.288	969.4074	921.3405	912.2834	918.1318	931.2103	948.248	967.5207	988.0856	1008.662	1030.208
	1250	13600.85	1200.837	999.1675	950.3209	941.2269	947.0369	959.9306	977.4214	996.5427	1017.898	1038.325	1059.536

Table 6.2: Regression coefficients for Full Factorial DoE based on 300 input combinations

β_0	β_s	β_{ss}	β_d	β_{dd}	β_{sd}
248.0494	731.1126	427.5785	-877.0716	1761.4316	-1365.4114

We also provide a graphical representation of the I/O data we are going to use to fit both the regression and the Kriging metamodels; see Fig. 6.1.

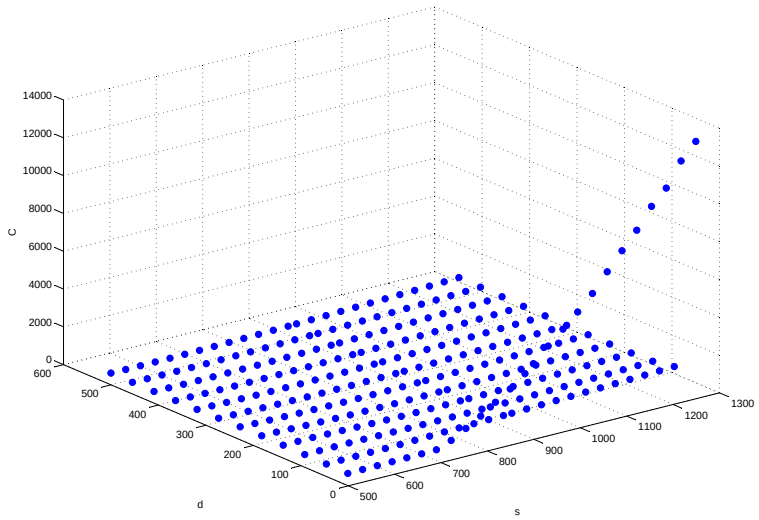


Figure 6.1: 3-D plot of the I/O data for the (s, S) inventory model, using Full Factorial DoE with 300 input combinations

When using RSM, we code the input variables in order to guarantee numerical accuracy in our results by working with a matrix of explanatory variables $\mathbf{X}$ having a reasonably low condition number. We first fit a regression metamodel, having the following structure:

$$y = \beta_0 + \beta_s s + \beta_{ss} s^2 + \beta_d d + \beta_{dd} d^2 + \beta_{sd} sd\,. \tag{6.2.1}$$

The values of the regression coefficients (associated to coded variables) are given in Table 6.2; Fig. 6.2 shows the resulting metamodel.

The t-test helps us to identify non-significant effects, which can be replaced by zero, leading to a reduced metamodel. The results suggest to replace the intercept and the second-order effect for s by zero; all the other effects appear to be important.

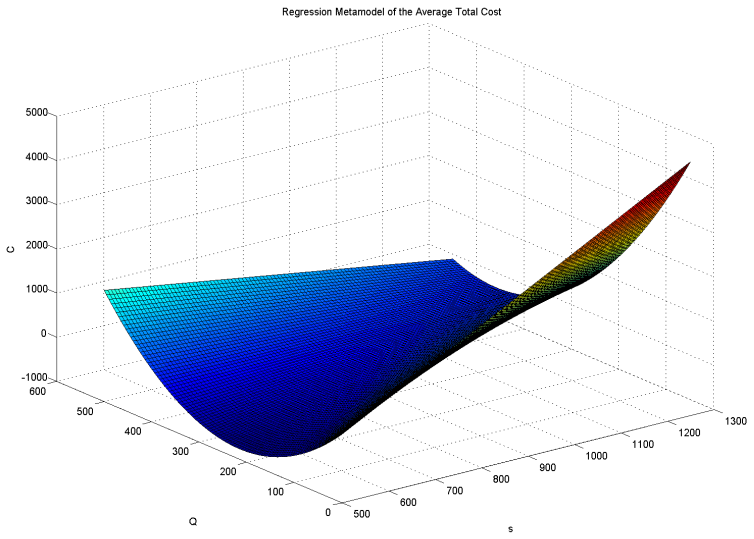


Figure 6.2: Regression metamodel for Full Factorial DoE using 300 input combinations

We do not report the results for the cross-validation procedure, since it would require a table with 300 rows. We only show the resulting scatterplot in Fig. 6.3, from which we conclude that the metamodel is not highly accurate, its relative accuracy being $0.8895 \leq \widehat{y_{(-i)}}/\overline{C_i} \leq 1.1413$. We also test the validity of the metamodel according to the procedure described in Section 4.2, based on the Studentized prediction error (see Eq. 3.2.27): the results obtained lead to the conclusion the metamodel is not valid, with 197 out of 230 that did not pass the test. We mention that — although not often used in practical applications (see Kleijnen, 2008) — we also increase the order of the regression metamodel, to check whether the resulting metamodel provides better results; however, it turns out to be still inaccurate, the average total cost taking some negative values on a given sub-area of the experimental region.

Then, we fit a Kriging metamodel over the same experimental area, starting from the same DoE: the resulting 3-D plot is shown in Fig. 6.4. Although omitting all the cross-validation results, we mention that the relative accuracy is $0.9675 \leq \widehat{y_{(-i)}}/\overline{C_i} \leq 1.0825$ and we show the scatterplot in Fig. 6.5, from which we conclude that the metamodel can be accepted. This conclusion is also supported by the results from the Studentized prediction error, according to Eq. 4.2.6. Comparing Figs. 6.2 and 6.4 and based on the cross-validation results, it is noticeable that the regression model provides poor approximation due to the particular behaviour of the output function, which is instead well reproduced by the Kriging metamodel.

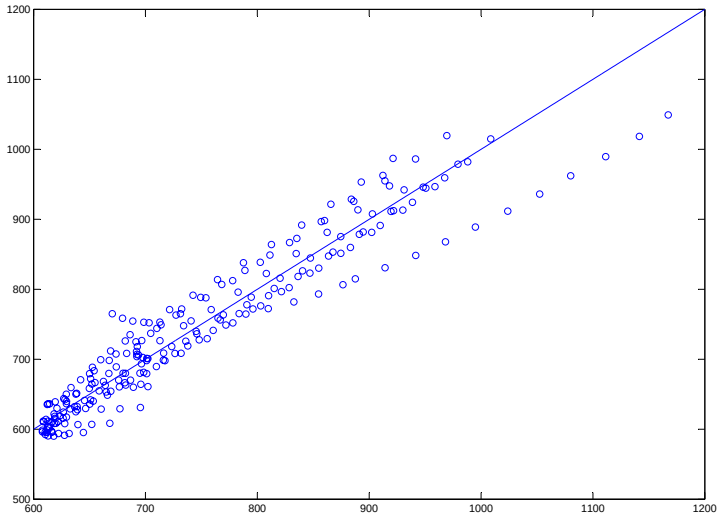


Figure 6.3: Scatterplot of the regression metamodel for Full Factorial DoE using 300 input combinations

As can be noticed by comparing the column corresponding to $d = 1$ in Table 6.1 with each of the next columns, the average total cost is much higher than all the others; this behaviour can be explained by looking at the corresponding values of d which determine the cost in each column: the first column, in fact, is associated to $d = 1$, which means that the order-up-to level is just $s + 1$ units. On the other hand, the demand size follows an exponential distribution with parameter $\lambda_D = 0.01$; therefore, we can compute the probability of receiving a demand for less than one item (remember that in this study we are ignoring integer constraints) as follows:

$$P(D \leq 1) = F_D(D \leq 1) = 1 - e^{-\lambda_D \cdot 1} \approx 0.01\,. \tag{6.2.2}$$

So, most probably (around 99%), the inventory would need to face a demand bigger than 1 unit, which would immediately lead the inventory level below the re-ordering point s causing a new order to be placed, eventually every time period, thus determining higher costs.

The big difference in the magnitude of the cost values associated to $d = 1$ with respect to the observed average costs associated to all the other input combinations gives the erroneous impression that the cost becomes almost constant over the remainder of the experimental area. Therefore, we consider the costs associated to $d = 1$ as outliers and eliminate this point from our design, since the corresponding output values are clearly sub-optimal and uninteresting to be further analyzed, our goal being to minimize the average total cost. This decision is also supported by the

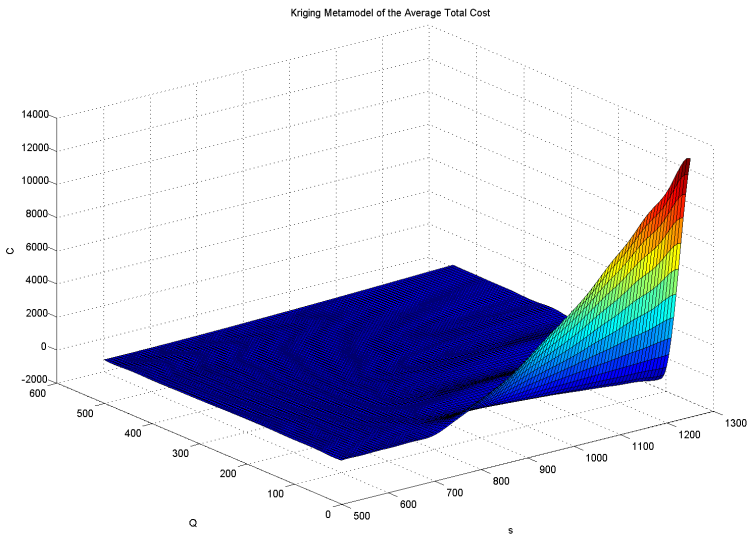


Figure 6.4: 3-D plot of the Kriging metamodel based on Full Factorial DoE using 300 input combinations

Table 6.3: Regression coefficients for Full Factorial DoE based on 300 input combinations, removing $d = 1$

β_0	β_s	β_{ss}	β_d	β_{dd}	β_{sd}
668.7668	185.7851	120.1313	−23.2856	111.7692	21.6447

optimal solution found by minimizing the average total cost using OptQuest; choosing to stop the heuristic search after 100 non-improving solutions, the solution proposed by OptQuest — after 454 simulations — is $s^+ = 571.71$, $d^+ = 397.57$, with an optimal cost equal to $C^+ = 604.97$, thus confirming that the area around $d = 1$ is not promising for our optimization purpose.

After removing all the samples having $d = 1$ from our design, we make a similar plot to the one in Fig. 6.1, resulting in Fig. 6.6. Based on the resulting input data set, we fit both a regression metamodel and a Kriging metamodel; for both the techniques a significant improvement in the accuracy can be observed.

The RSM metamodel fitted on this data set gives the OLS estimates for the regression coefficients reported in Table 6.3; the t-test suggests all regression coefficients to be significant. The metamodel itself is then shown in Fig. 6.7. Although omitting the results for the cross-validation procedure, we show the resulting scatterplot in Fig. 6.8, for which the relative accuracy is $0.9108 \leq \widehat{y_{(-i)}}/\overline{C_i} \leq 1.0679$. Despite the values

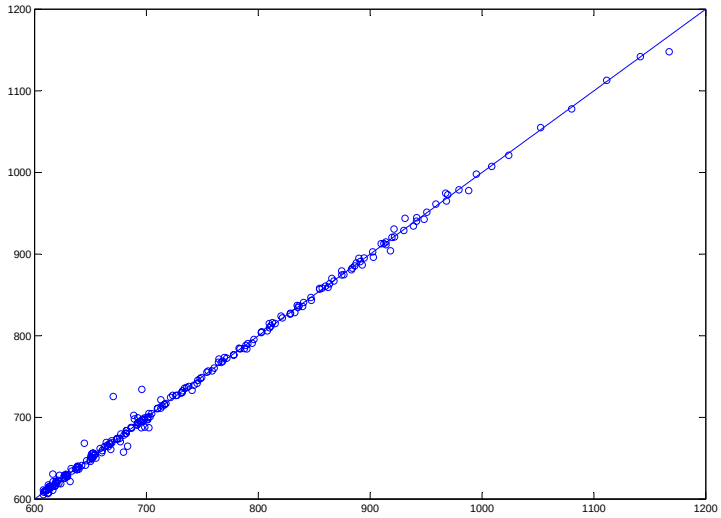


Figure 6.5: Scatterplot of Kriging metamodel for Full Factorial DoE using 300 input combinations

of the relative accuracy based on cross-validation appear to be quite good and close to 1, the regression metamodel does not pass the statistical test based on the Studentized prediction error. This seeming contradiction can be explained as follows: having Eq. 3.2.22 in mind, we recall that we are considering a steady-state simulation, which we achieve after a very long run; this implies that both $\widehat{var(\overline{C_i})}$ and $\widehat{var(\overline{y_{-i}})}$ are small numbers, so their sum is small and — being in the denominator — it makes the ratio big, eventually higher than the threshold $t_{m-1;1-\alpha/2}$, even if prediction errors remain small. For further details on this topic we refer to the specific discussion in Kleijnen (2008).

Fitting a Kriging metamodel on the data set from Table 6.1 after deleting the column associated to $d = 1$ results in the metamodel shown in Fig. 6.9. Again we omit the results for the cross-validation procedure, just showing the resulting scatterplot in Fig. 6.10, for which the relative accuracy is $0.978 \leq \widehat{y_{(-i)}}/\overline{C_i} \leq 1.0355$. Computing the Studentized prediction error as in Eq. 4.2.6 gives similar results to those discussed for the regression model, i.e. not every input combination passes the test (more precisely, only 3 out of 207 points do not pass), implying that the metamodel should not be accepted; however, we do accept it based on the relative prediction errors, which are smaller than 3%.

The previous experiment was indeed time consuming, since it took about 48 hours to be completed, on a PC with Windows XP, 1 GB RAM and 2.13 GHz. However, it gave us useful information to restrict the region of interest, thus focussing on a

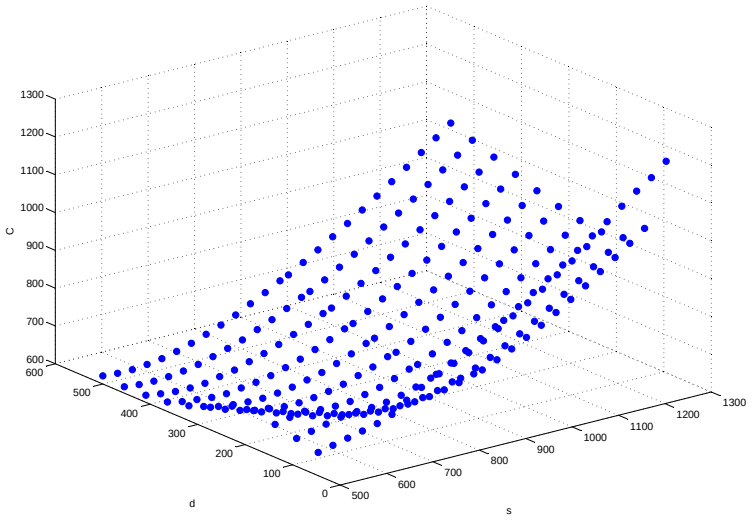


Figure 6.6: 3-D plot of the I/O data for the (s, S) inventory model, based on Table 6.1, removing column $d = 1$

smaller area to look for the optimal solution on the basis of a more accurate (local) metamodel. So we select a maximin LHS (see Fang et al., 2006) based on 25 points over the area delimited by $500 \leq s \leq 1100$ and $50 \leq d \leq 400$, using the on-line tool available on the website www.spacefillingdesigns.nl. A *maximin* design selects points in the design space aiming at maximizing the separation distance, i.e. the minimal distance among pairs of points. Moreover, LHS designs are *non-collapsing*: this implies that — given the levels for each factor — each level occurs exactly once in the design. The importance of such a property becomes evident when it is not known a priori which factors are important: in fact, if one of the design parameters turns out to have (almost) no influence on the simulation outcome, then two design points differing only in the value of this parameter will 'collapse', i.e. they reduce to the same point, which usually is not a desirable situation. Typically a grid might generate collapsing designs.

After simulating the 25 input combinations using the Arena model, we fit a regression model according to (6.2.1), using the design in Table 6.4. The values of the regression coefficients are shown in Table 6.5. A t-test confirms to keep all the factors, being all significant. The resulting metamodel is shown in Fig. 6.11. The regression model provides good accuracy, as Table 6.6 shows through the relative prediction errors. Notice that some input combinations are missing in the table, because they have

Table 6.4: Maximin LHS design using 25 observations

s	$d = S - s$
500	108.33
525	181.25
550	254.17
575	327.08
600	400
625	50
650	122.92
675	195.83
700	268.75
725	341.67
750	64.583
775	137.50
800	210.42
825	283.33
850	356.25
875	79.167
900	152.08
925	225
950	297.92
975	370.83
1000	93.75
1025	166.67
1050	239.58
1075	312.50
1100	385.42

Table 6.5: Regression coefficients for LHS based on 25 observations

β_0	β_s	β_{ss}	β_d	β_{dd}	β_{sd}
641.55	89.853	95.253	-29.716	52.712	31.52

Table 6.6: Cross-validation for RSM based on LHS using 25 observations

i	$\widehat{y_{(-i)}}$	$\widehat{y_{(-i)}}/w_i$
2	674.47	0.964
3	635.4	0.987
4	637.64	1.042
7	690.76	1.021
8	630.77	1.009
9	608.14	0.995
10	623.77	1.004
11	731.57	0.993
12	668.2	1.032
13	632.39	1.002
14	632.71	0.979
15	671.88	0.991
16	729.99	0.978
17	691.89	1.018
18	679.87	0.988
19	706	0.979
20	774.58	1.012
21	776.94	0.985
22	764.85	1.014
23	776.85	0.999
24	833.74	1.016

not been included in the cross-validation set, in order to avoid extrapolation. The corresponding scatterplot is given in Fig. 6.12. Then, we compute the Studentized prediction error, which would suggest that the metamodel is not valid; however, due to the relatively small prediction errors, we accept the regression metamodel.

Using the same design from Table 6.4, we fit a Kriging metamodel, which is shown in Figure 6.13, together with the corresponding contour plot. The cross-validation procedure also gives good results, comparable with those obtained when fitting the regression model: they are represented through a scatterplot in Fig. 6.14 and summarized in Table 6.7. Kriging metamodels is confirmed to provide better accuracy than regression metamodels, even though it is not so striking as it was on the wider experimental region, because regression metamodels usually perform better when the region of interest is small (see (e.g.) Kleijnen, 2008; Montgomery, 2009).

Again, based on the acceptable accuracy provided through the relative prediction errors we accept the metamodel, although the test based on the Studentized prediction error would reject it.

Using the metamodels we have just fitted and validated, we solve the optimization problem of minimizing the *estimated* average total cost ($\widehat{\widehat{C}}$) over the experimental design area of interest, through Matlab's `fmincon`. Finally, we simulate the proposed

Table 6.7: Cross-validation for Kriging based on LHS using 25 points

i	$\widehat{y_{(-i)}}$	$\widehat{y_{(-i)}}/w_i$
2	686	0.980
3	646.11	1.004
4	627.81	1.026
7	681.31	1.007
8	630.6	1.009
9	611.28	1.000
10	624.55	1.006
11	724.11	0.983
12	648.69	1.002
13	628.39	0.995
14	645.15	0.998
15	678.44	1.001
16	747.56	1.001
17	688.24	1.013
18	683.94	0.994
19	733.85	1.018
20	755.53	0.987
21	766.47	0.972
22	755.11	1.001
23	776.82	0.999
24	797.32	0.972

Table 6.8: Optimal solutions estimated through regression and Kriging metamodels.

	s^+		d^+		$\widehat{\overline{C}}^+$		$\overline{C}^+$	
	RSM	KG	RSM	KG	RSM	KG	RSM	KG
R1	636.42	628.47	302.86	340.44	610.45	604.96	610.04	607.28
R2	632.76	617.93	305.56	360.70	610.61	604.89	609.27	606.56
R3	636.76	622.19	304.58	361.31	610.67	604.53	610.01	607.73
R4	637.82	611.36	303.03	365.49	610.76	606.14	610.54	606.72
R5	633.63	612.53	303.29	365.21	612.20	607.31	610.95	608.91

optimal solution, replicated $m = 10$ times, thus obtaining the optimal *observed* average total cost ($\overline{C}$).

In order to make our results less dependent on the particular seed used for CRNs in our simulation run, we consider five macro-replicates, each using a different seed. The whole procedure described before is repeated to fit both the regression and the Kriging metamodel and the optimization process is performed. Table 6.8 summarizes the optimal combinations of the input factors (s, d) found by `fmincon` using either the regression or the Kriging metamodel; then, the predicted optimal cost $\widehat{\overline{C}}^+$ and the corresponding simulated values $\overline{C}^+$) are reported.

Aiming at comparing the two metamodelling techniques, we perform a t-test to check whether the difference between the optimal solutions found by each strategy is statistically significant or not. Therefore the null hypothesis we test is

$$H_0 : \mu_{RSM} - \mu_{KG} = 0\,. \tag{6.2.3}$$

We denote the total number of macroreplicates by R (in our example, we took $R = 5$) while X_i and Y_i refer to the average cost $\overline{C}$ derived from Kriging (KG) and regression (RSM) metamodels, respectively, in the i-th macroreplicate. Then, the null hypothesis will be accepted if

$$|t^\star| \leq t_{R-1;1-\alpha/2}\,, \tag{6.2.4}$$

where

$$t^\star = \frac{\overline{D}}{s(\overline{D})} \tag{6.2.5}$$

$$\overline{D} = \frac{1}{R}\sum_{i=1}^{R}(Y_i - X_i) \tag{6.2.6}$$

$$s(\overline{D}) = \frac{s_D}{\sqrt{R}} \tag{6.2.7}$$

$$s_D^2 = \frac{1}{R-1}\sum_{i=1}^{R}(D_i - \overline{D})^2 \tag{6.2.8}$$

Applying Eq. 6.2.4 to the data in Table 6.8 we conclude that the null hypothesis in (6.2.3) is rejected, with a significance level $\alpha = 0.05$. Kriging is the metamodelling technique which performs better, giving an optimal average total cost of $\overline{C}^{+} = 607.44$ versus $\overline{C}^{+} = 610.16$ based on the regression metamodel, after averaging over the 5 macro-replicates; moreover, Kriging also provides more accurate metamodels, as discussed before.

6.2.1 (s, S) inventory model with service level constraint

This section presents the computational experiments we performed on a possible variation on the (s, S) inventory model, where no shortage costs are considered, accounting for possible out-of-stock through a measure of the service level, as described in Section 2.2.1.

Comparing the average total cost we obtained as one output of the simulation runs with the corresponding output of the previous model (where shortage costs are included) we notice that, depending on how often shortages occur and on the magnitude of the unit shortage costs with respect to the other cost coefficients, it may happen that — when reaching steady-state conditions, as we assume after $N = 30000$ periods — the influence of the shortage costs on the average total cost is reduced.

Similarly to what described in Section 6.2, we fit a (regression or Kriging) metamodel for each output. For this study we further restrict the region of interest to the area delimited by $675 \leq s \leq 1100$ and $140 \leq d \leq 400$, due to some pilot experiments on the experimental region previously chosen which highlighted this smaller area to be more promising.

Given the new experimental region, we construct a design of experiments using a *nested* LHS: nested designs consist of two separate designs, one being a subset of the other, which are useful when dealing with sequential evaluations. This allows to start from a small number of design points and — if it turns out that the resulting metamodel is not accurate enough — then we just need to evaluate an extra set of design points, thus taking advantage of having already evaluated an initial set. For further details refer to Husslage et al. (2005). We first made some pilot experiments starting from a relatively small number of points (namely, 15); however, we noticed that this design provided metamodels which were not enough accurate. So, we increased the number of design points from 15 to 29, which required for only 14 more points to simulate. The resulting design is the one we use in this experiment.

Then, we fit a regression metamodel for each output over this area. The cross-validation procedure applied to the regression metamodel for the total cost gives the scatterplot in Fig. 6.15, supported by relative prediction errors within the range $0.9848 \leq \hat{y}_{1,(-i)}/w_{1,i} \leq 1.0073$, where w_1 denotes the total cost and $\hat{y}_1$ its estimated value. Computing the Studentized prediction errors would suggested not to accept the metamodel, due to one point (out of 25) which did not pass the test in 3.2.27; nevertheless, we do accept it, since we find the relative prediction errors to be wholly satisfactory.

We cross-validate the regression metamodel of the service level as well, leading to the scatterplot in Fig. 6.16 and the corresponding relative prediction errors such that $0.9101 \leq \hat{y}_{2,(-i)}/w_{2,i} \leq 1.2201$, w_2 denoting the service level and $\hat{y}_2$ its estimated value. Even though also for this metamodel there is one point not passing the test on Studentized prediction errors, we decide to accept it.

We show the 3-dimensional plot and the related contour plot for the two metamodels in Figs. 6.17-6.18.

Based on the metamodels we have just fitted, we iteratively solve the following constrained optimization problem, using MATLAB's `fmincon`:

$$\begin{aligned} \min \quad & \widehat{y_1} \\ s.t. \quad & \widehat{y_2} \leq \Gamma \end{aligned} \tag{6.2.9}$$

where Γ denotes a threshold on the service level constraint, taking values from the set $\{0.06, 0.10, 0.15, 0.20, 0.25\}$; this implies that in every iteration we select a value for Γ from that interval and solve the resulting optimization problem. The optimal solutions are reported in Table 6.9: more precisely, we show the optimal input combinations (s^+, d^+), the corresponding estimated values for the total cost $\widehat{y_1}^+$ and service level $\widehat{y_2}^+$, which are compared to the simulated values w_1^+ and w_2^+, respectively.

As already observed in the previous section, we want to derive results that are as much as possible independent on the seed used for generating the random numbers streams; therefore, we take five macroreplicates and repeat the whole procedure applied so far, each time choosing a different seed. The corresponding results are shown in rows labelled as R_2-R_5 in Table 6.9.

Now we fit Kriging metamodels to the total cost and service level outputs, starting from the same design we adopted for building the regression metamodels. The Kriging metamodel of the total cost gives relative prediction errors, based on cross-validation, within the range $0.9685 \leq \hat{y}_{1,(-i)} \leq 1.0059$; the corresponding scatterplot is shown in Fig. 6.19. Although the relative prediction errors do not differ so much from those obtained on the regression metamodel, we notice that the Kriging metamodel of the total cost passes the test on the Studentized prediction errors (see Eq. 4.2.9).

Slightly better results are shown for the Kriging metamodel of the service level, having relative prediction errors $0.9257 \leq \hat{y}_{2,(-i)} \leq 1.1882$ and the scatterplot in Fig. 6.20. Moreover, the accuracy of this Kriging metamodel is certified through the test on the Studentized prediction errors.

The Kriging metamodels of the total cost and service level are respectively depicted in Figs. 6.21-6.22, together with the related contour plots.

Finally we solve the constrained optimization problem in Eq. 6.2.9, based on the two Kriging metamodels, and using the same values for the threshold Γ we selected before. The optimal solutions we obtain are shown in Table 6.10, together with the predicted and simulated values of the two output functions, for each of the five macroreplicates.

In order to compare the results obtained using the two metamodelling techniques (see Tables 6.9-6.10), we perform the test in Eq. 6.2.4, on the null hypothesis that

Table 6.9: Optimal Solutions based on Regression metamodels.

♯ Macro-replicate	s^+	d^+	$\widehat{y_1}^+$	$\widehat{y_2}^+$	w_1^+	w_2^+
	1099.934	206.5844	813.0899	0.06	812.7126	0.0682
	983.2784	278.5672	734.5998	0.1	738.8435	0.1088
$R1$	898.4851	301.6565	686.083	0.15	689.8956	0.1547
	838.2599	308.5755	656.9543	0.2	658.2477	0.2003
	789.7343	310.4617	637.5569	0.25	635.4851	0.245
	1100	222.6205	817.8318	0.06	816.6398	0.0645
	981.9042	277.9287	735.1668	0.1	738.8115	0.1076
$R2$	897.7408	299.6517	686.6008	0.15	689.5749	0.1534
	837.7156	306.7069	657.5173	0.2	658.1527	0.1988
	789.173	309.0545	638.1302	0.25	637.2915	0.2477
	1100.076	237.5336	820.4742	0.0603	818.9041	0.065
	984.3422	278.9105	736.1956	0.1	740.0932	0.1074
$R3$	898.9349	301.7385	686.9218	0.15	691.7657	0.1552
	838.7796	308.0945	657.6125	0.2	658.4372	0.2
	790.413	309.5734	638.1718	0.25	637.4275	0.2476
	1100	230.8388	818.1713	0.06	816.3733	0.0647
	983.42	277.8577	734.3831	0.1	739.1564	0.1089
$R4$	900.0417	297.6824	685.9411	0.15	688.8292	0.1527
	840.8698	303.1186	657.1359	0.2	658.5216	0.2003
	793.0914	304.21	638.0407	0.25	637.2417	0.2471
	1102.287	191.2459	816.008	0.06	815.254	0.0701
	989.6635	278.2089	741.7744	0.1	746.5912	0.1081
$R5$	902.827	302.0406	691.3457	0.15	694.8458	0.1541
	841.6147	309.1239	661.2936	0.2	663.1494	0.2013
	792.4264	311.0787	641.3003	0.25	640.7156	0.2486

Table 6.10: Optimal Solutions based on Kriging metamodels.

♯ Macro-replicate	s^+	d^+	$\widehat{y_1^+}$	$\widehat{y_2^+}$	w_1^+	w_2^+
	1089.493	343.5903	844.3887	0.06	846.0512	0.0594
	993.2601	300.6332	753.4055	0.1	752.8848	0.0998
R_1	899.4497	297.9801	687.1345	0.15	688.7964	0.1538
	840.2203	294.3123	654.7266	0.2	655.2682	0.2031
	789.844	301.3118	636.1993	0.25	635.6792	0.2524
	1052.751	390.7143	836.9825	0.06	835.7697	0.0672
	981.1025	318.4281	752.3388	0.1	752.2652	0.102
R_2	920.2961	254.8813	691.4712	0.15	691.7822	0.15
	860.1897	251.314	658.2201	0.2	657.7207	0.1982
	795.5851	277.1283	634.1674	0.25	633.8311	0.2517
	1100	339.6778	849.6484	0.0603	853.6958	0.0579
	979.8271	328.4263	752.2109	0.1	753.1679	0.101
R_3	888.3206	317.0681	683.9005	0.15	688.5157	0.1595
	828.5839	313.7302	649.9135	0.2	654.5682	0.2067
	822.2448	231.3694	638.8027	0.25	637.8209	0.2462
	1065.055	185.6863	773.2251	0.06	781.8669	0.0826
	971.2554	329.3378	743.9569	0.1	747.8006	0.1067
R_4	889.7075	316.2066	683.6437	0.15	688.086	0.1554
	837.3345	305.5393	654.7059	0.2	657.5555	0.2031
	790.1892	298.3735	635.435	0.25	635.2788	0.2532
	1068.46	194.7875	776.4241	0.06	788.2141	0.0834
	974.6031	336.1392	750.8151	0.1	754.533	0.106
R_5	890.9873	320.5723	690.1346	0.15	692.7968	0.1574
	842.9643	299.5267	660.5789	0.2	661.6827	0.2042
	802.2435	277.7103	638.3657	0.25	638.1493	0.2491

the mean costs in the two methods are not significantly different from each other. Based on this test, we conclude that the null hypothesis is rejected only for $\Gamma = 0.10$, with a significance level of $\alpha = 0.05$: in this case, the optimal solution found on the regression metamodel gives an *observed* cost — averaged over the five macroreplicates — equal to $\overline{C}^+ = 740.69$ whereas for Kriging we have $\overline{C}^+ = 752.13$. Since we are interested in minimizing the total cost as well as satisfying the service level constraint, we should also take this second output into account: so we notice that, for $\Gamma = 0.10$, both the regression metamodel and the Kriging metamodel give an average value for the service level which violates the constraint; the difference between the two means is statistically significant, and this time the Kriging metamodel provides a better service level, averaged over the five macroreplicates. This stands to reason, because the two objectives are usually conflicting; therefore, in that case Kriging is able to keep the service level closer to the target value, but a price must be paid for it, i.e. the cost increases.

6.3 Robust Simulation-Optimization using RSM

After having discussed the classic simulation-optimization, we present the experimental results obtained by applying the robust simulation-optimization combining Taguchi and RSM, which we extensively discussed in Chapter 3.

It is worth to explicitly mention that our robustness approach accounts not only for the *aleatory* uncertainty but also for the *epistemic* uncertainty: at this stage, in fact, we assume that the parameters of the distribution of the random variables (demand size and lead time) are not known, but they can take values within a specified interval, according to a *prior distribution*; in our example we assume that both the mean of the Poisson distribution for the lead time and the mean of the exponential distribution for the demand size follow a Normal distribution, whose mean equals the 'base' value used in classic optimization whereas the variance expresses the variability of the parameter over the specified interval. Nevertheless, all these assumptions are introduced without loss of generality, since the proposed methodology remains applicable even if different probability distributions are taken into account.

We apply the proposed method starting from a design obtained by crossing two LHS designs over the control and noise factors region of interest, respectively: the design involving the control variables (s, d) is a two-dimensional maximin LHS, while we sample the noise factors (λ_D, λ_L) still following a LHS, but taking values from a Normal distribution truncated between $\mu_i - 2\sigma_i$ and $\mu_i + 2\sigma_i$, where μ_i and σ_i denote the mean and standard deviation of the i-th noise factor (either the demand size or the lead time).
We choose $n_{\mathbf{d}} = 25$ and $n_{\mathbf{e}} = 20$, implying that we need to run 500 input combinations, each replicated $m = 10$ times. As discussed in Section 6.1, we use Common Random Numbers (CRN) when simulating replication r ($r = 1, \ldots, m$) of the $n_{\mathbf{d}}$ combinations of the decision variables, to improve the accuracy of the estimated effects of these decision variables. However, we do not apply CRN when simulating the $n_{\mathbf{e}}$

combinations of the environmental variables; in fact, we shall estimate the epistemic variability assuming independent observations.

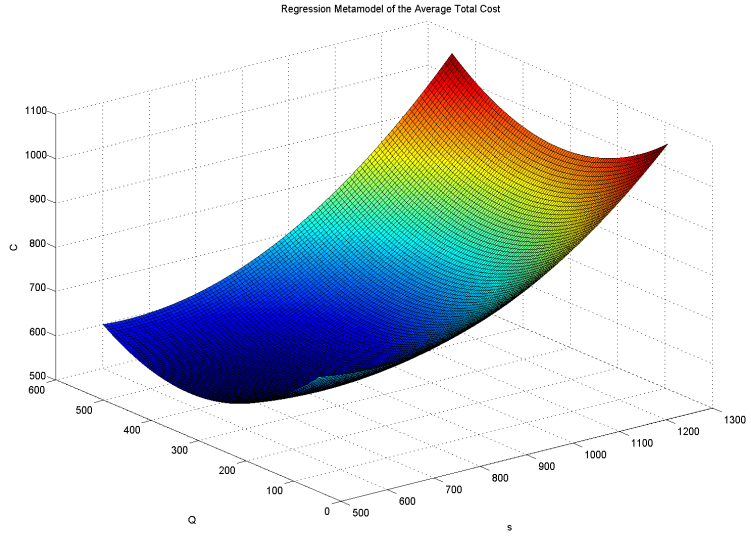
First we fit a regression metamodel for the total cost over the 4-dimensional input space, according to Eq. 3.2.8. Notice that, replications being available, we build the matrix of explanatory variables $\mathbf{X}$ based on all possible scenarios (i.e. combinations of control and noise factors), each replicated m times; this implies that each input combination is repeated m times in the matrix $\mathbf{X}$, thus increasing its number of rows from $n = n_{\mathbf{d}} \cdot n_{\mathbf{e}}$ to

$$N = \sum_{i=1}^{n} m_i = m\, n_{\mathbf{d}} \cdot n_{\mathbf{e}}\,, \tag{6.3.1}$$

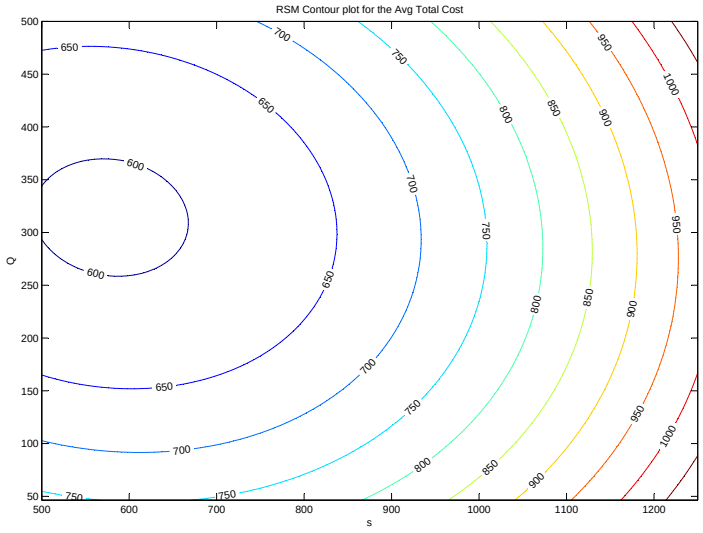
with m_i identical rows; in particular, we keep the number of replications fixed so $m_i = m$ for all i.

The cross-validation of the resulting metamodel produces the scatterplot in Fig. 6.23 and relative prediction errors $0.9276 \leq \hat{y} \leq 1.0844$.

Accepting the metamodel as valid, we use the estimated regression coefficients to derive the metamodels for the mean and standard deviation of the total cost, based on Eqs. 3.2.15-3.2.16. Notice that $\mu_{\mathbf{e}} = [\lambda_{D,nom}, \lambda_{L,nom}]$ and $\Omega_{\mathbf{e}} = diag(\sigma^2_{\lambda_D}, \sigma^2_{\lambda_L})$, because we assume that the two environmental variables (λ_D and λ_L) are independent. The two metamodels, together with the corresponding contour plots, are shown in Figs. 6.24-6.25.



(a) 3-D plot



(b) Contour plot

Figure 6.7: Regression metamodel based on data in Table 6.1, removing column $d = 1$

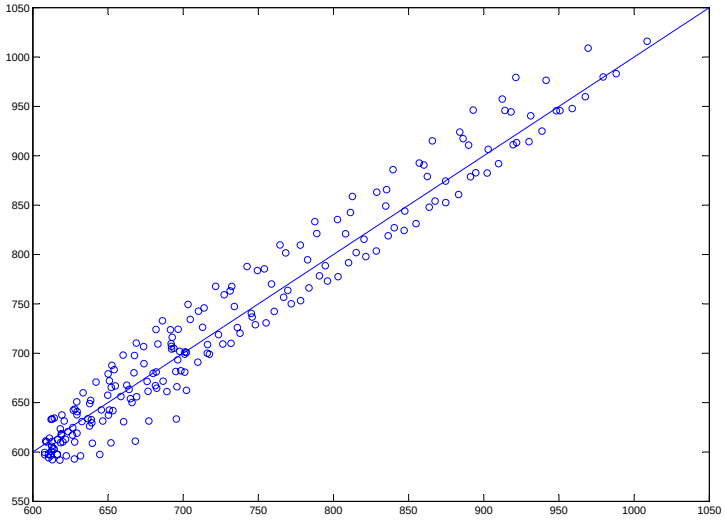
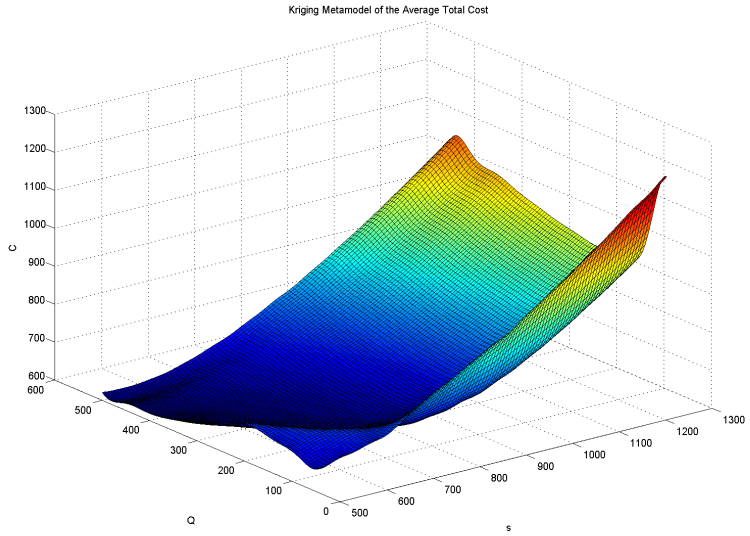
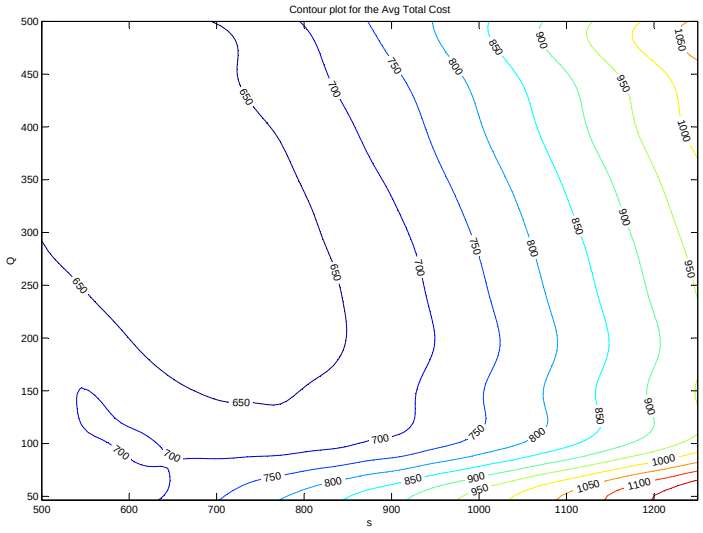


Figure 6.8: Scatterplot of regression metamodel based on data in Table 6.1, removing column $d = 1$



(a) 3-D plot



(b) Contour plot

Figure 6.9: Kriging metamodel based on data in Table 6.1, removing column $d = 1$

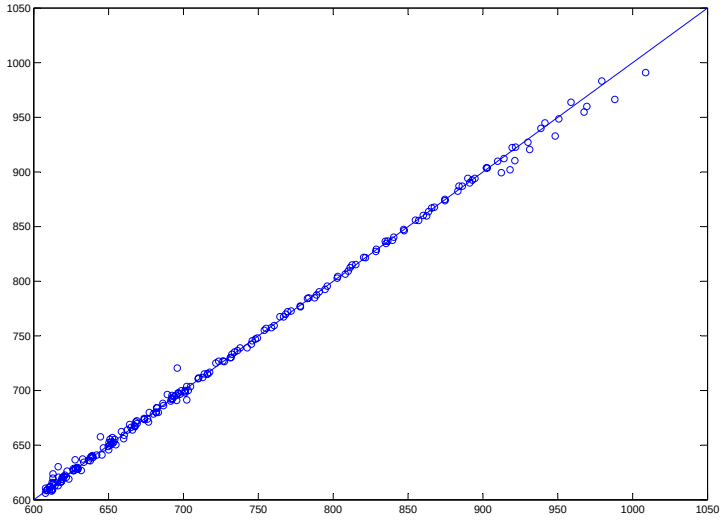
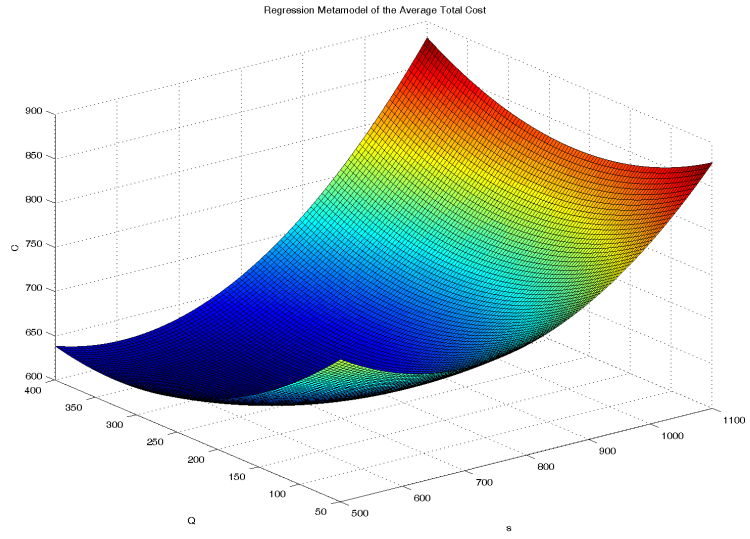
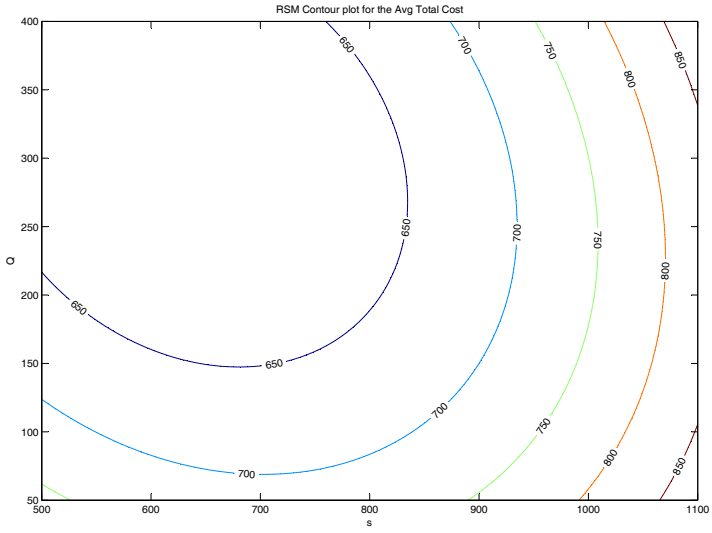


Figure 6.10: Scatterplot of Kriging metamodel based on data in Table 6.1, removing column $d = 1$



(a) 3-D plot



(b) Contour plot

Figure 6.11: Regression metamodel based on LHS using 25 observations

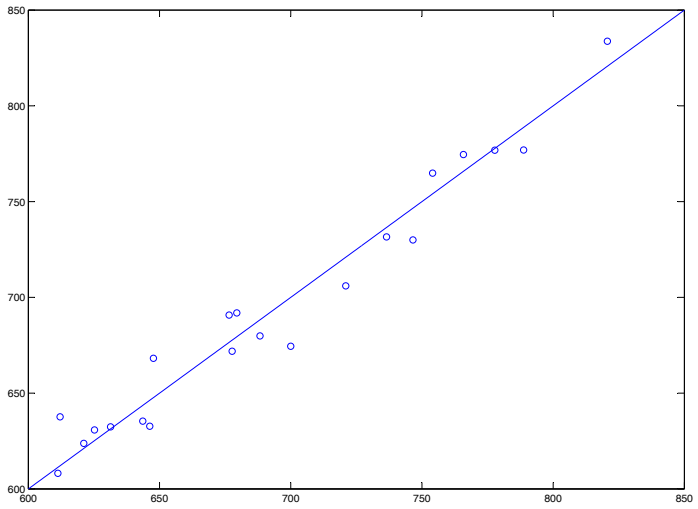
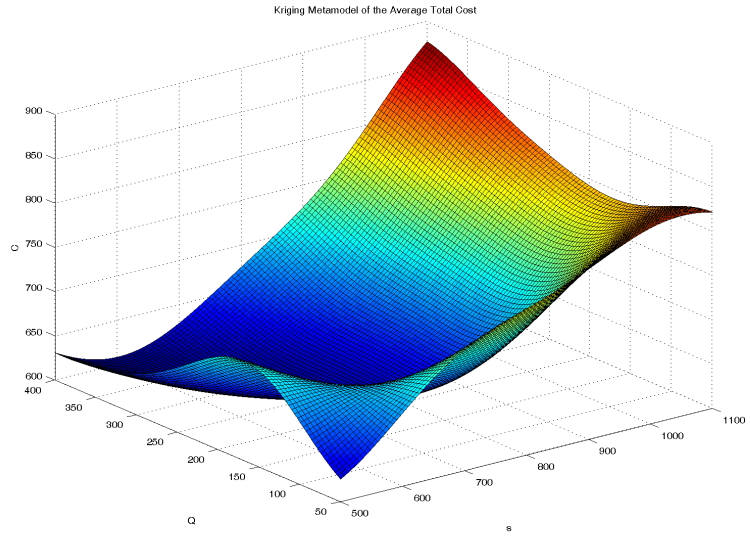
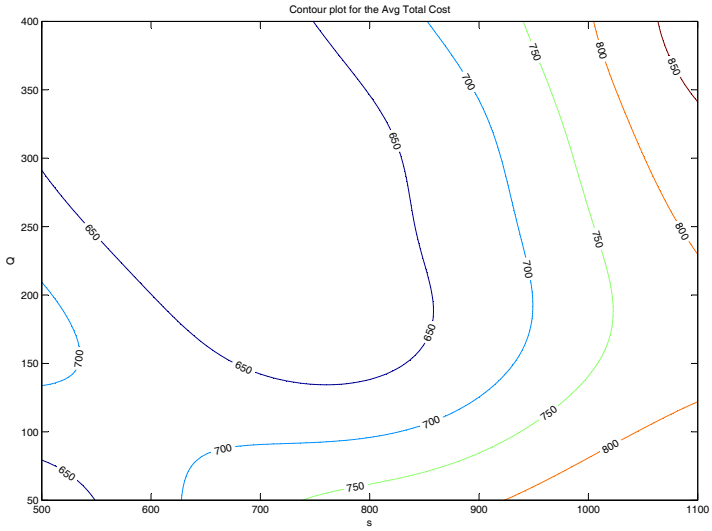


Figure 6.12: Scatterplot of regression metamodel for LHS using 25 observations



(a) 3-D plot



(b) Contour plot

Figure 6.13: Kriging metamodel based on LHS using 25 observations

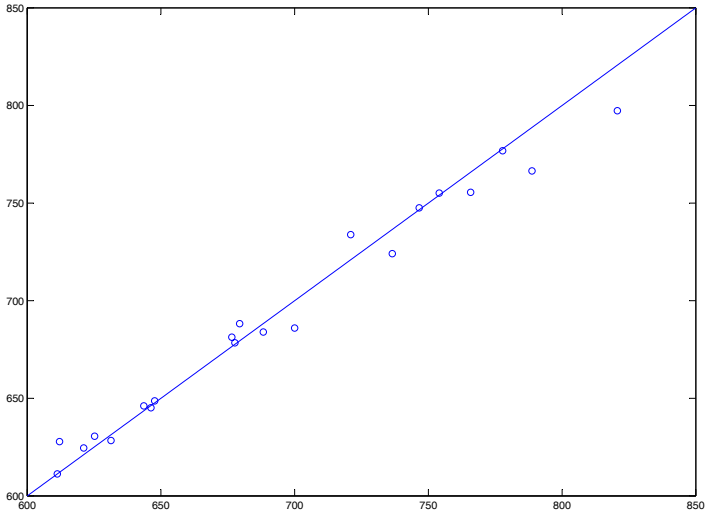


Figure 6.14: Scatterplot of Kriging metamodel for LHS using 25 observations

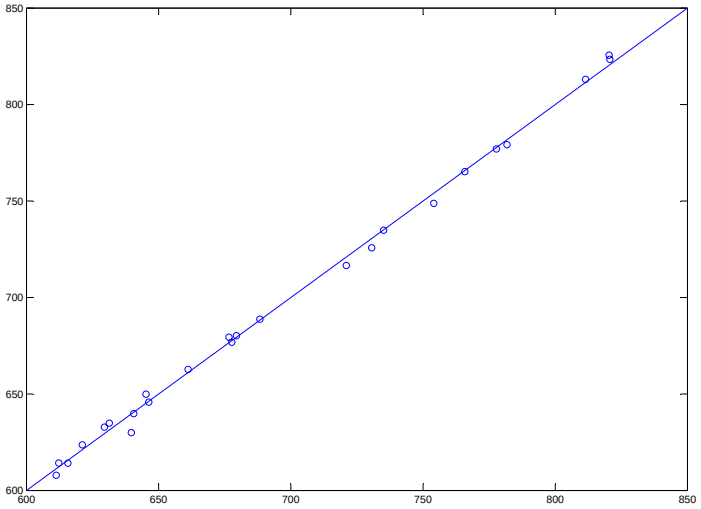


Figure 6.15: Scatterplot of regression metamodel for the total cost

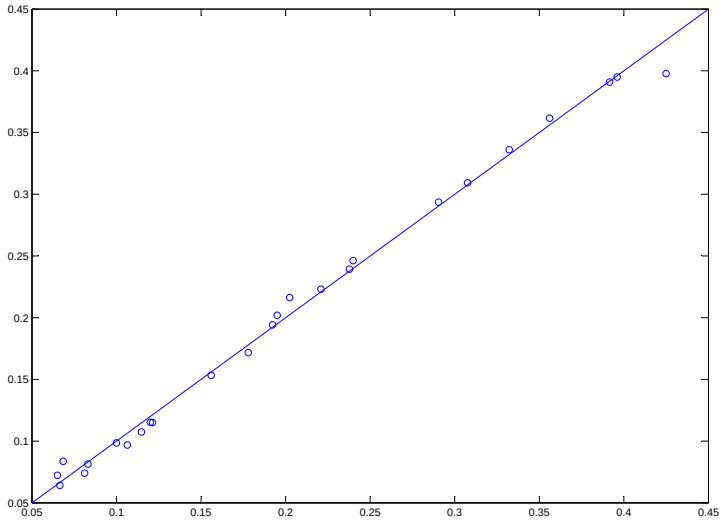
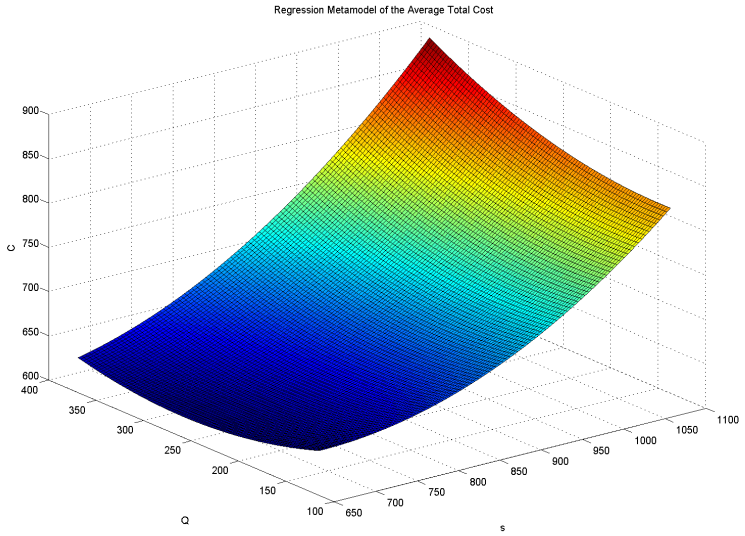
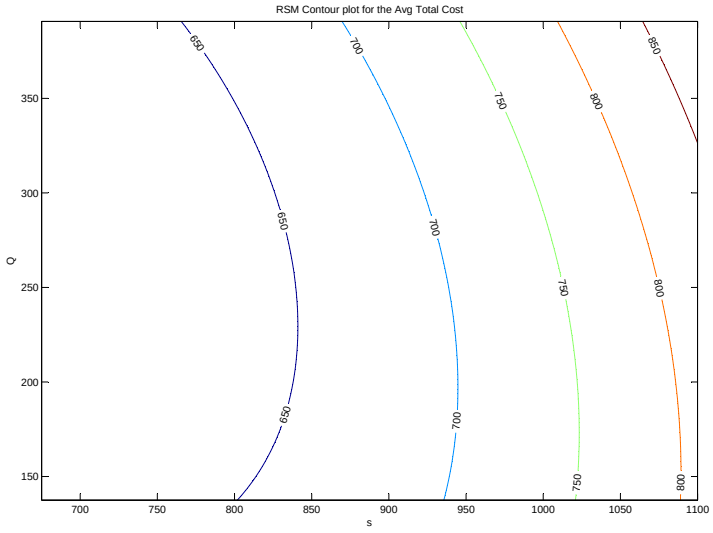


Figure 6.16: Scatterplot of regression metamodel for the service level

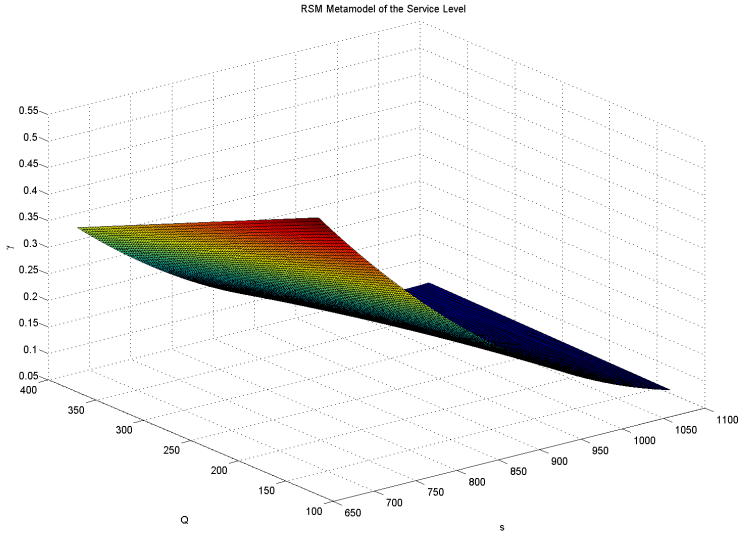


(a) 3-D plot

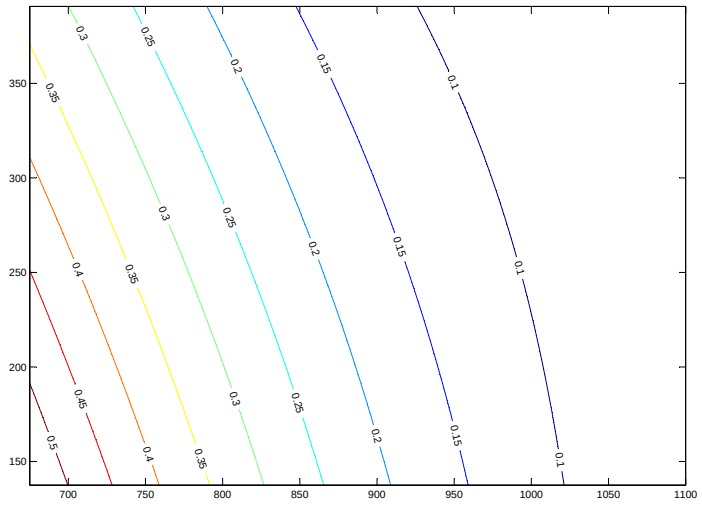


(b) Contour plot

Figure 6.17: Regression metamodel for the total cost



(a) 3-D plot



(b) Contour plot

Figure 6.18: Regression metamodel for the service level

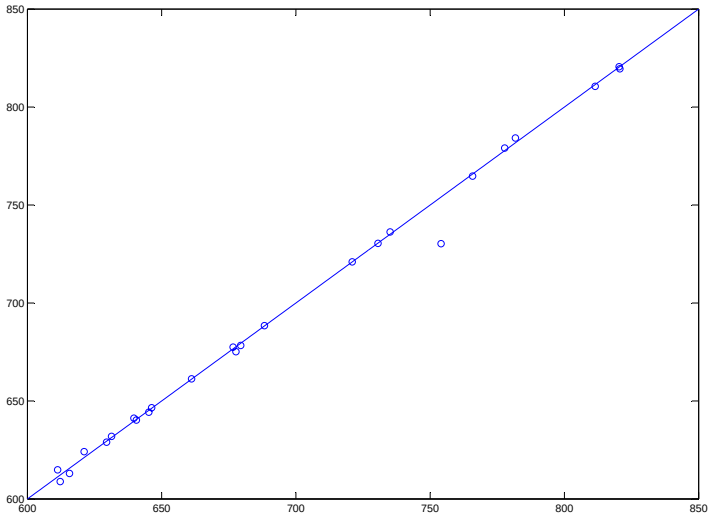


Figure 6.19: Scatterplot of Kriging metamodel for the total cost

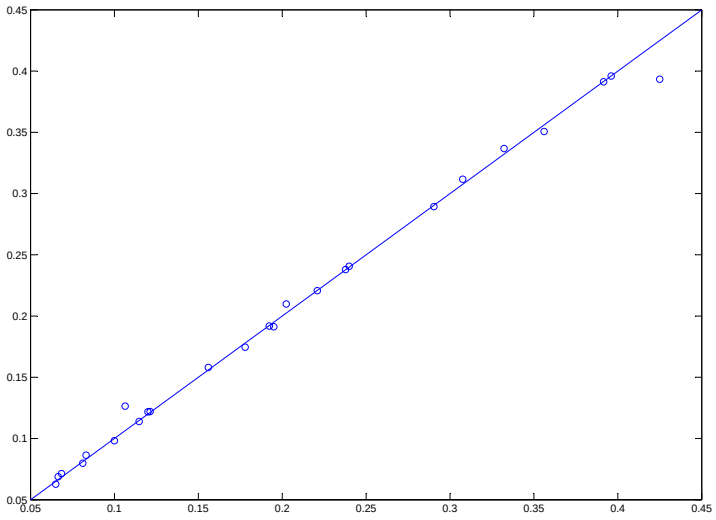
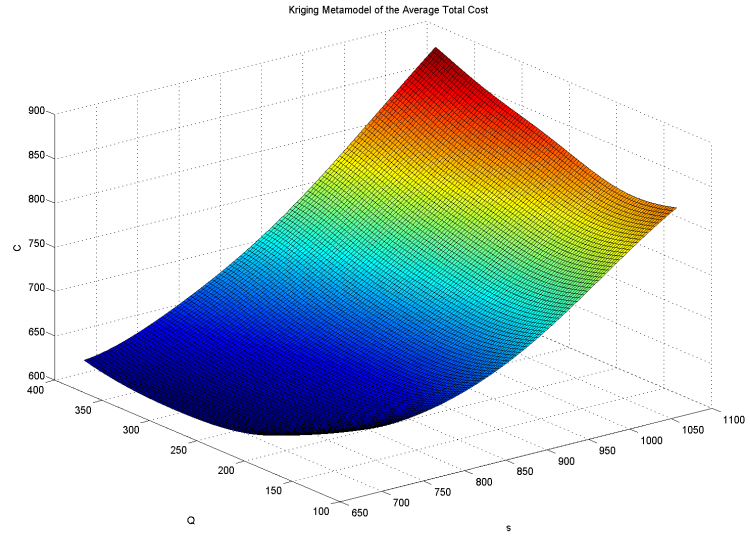
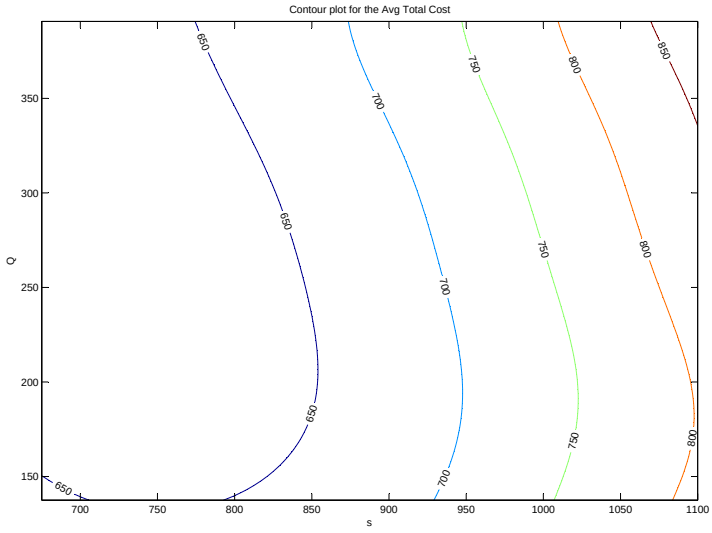


Figure 6.20: Scatterplot of Kriging metamodel for the service level

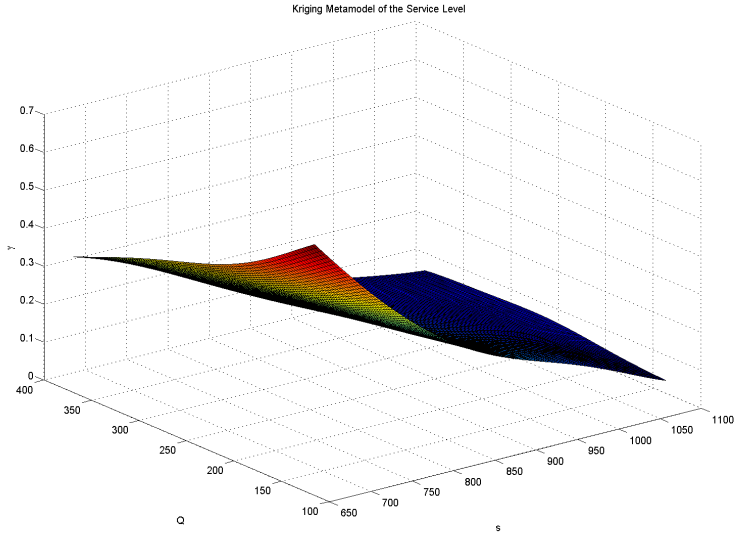


(a) 3-D plot

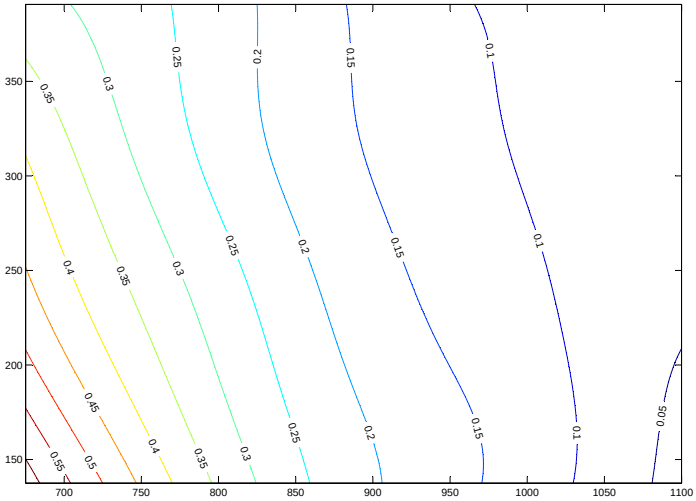


(b) Contour plot

Figure 6.21: Kriging metamodel for the total cost



(a) 3-D plot



(b) Contour plot

Figure 6.22: Kriging metamodel for the service level

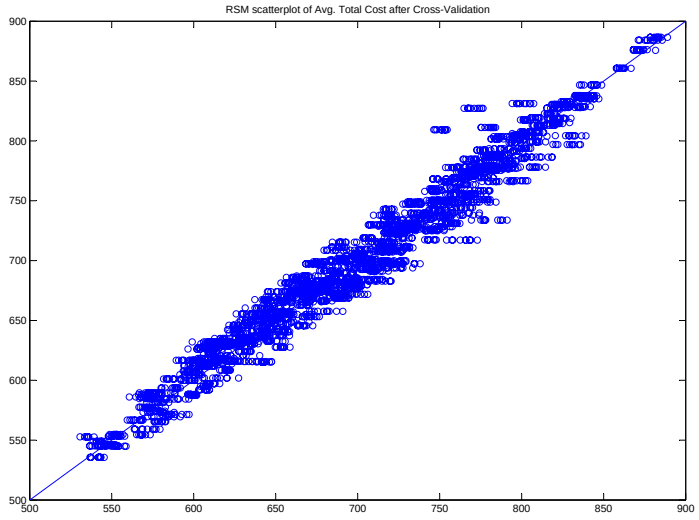
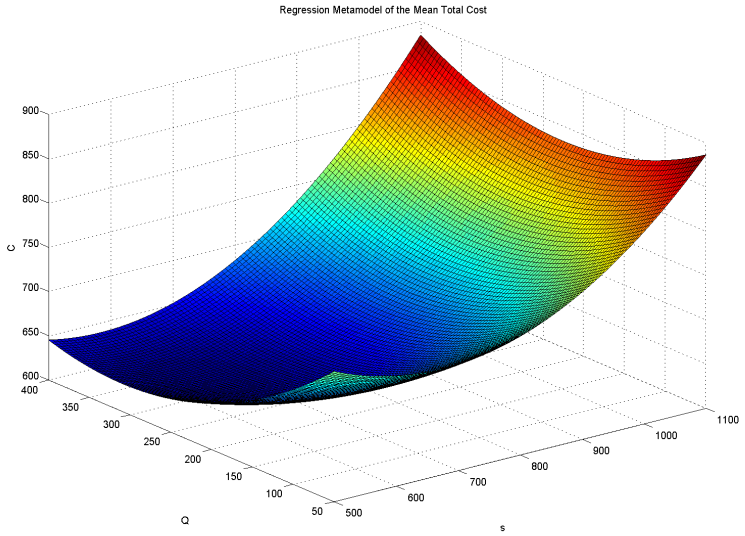
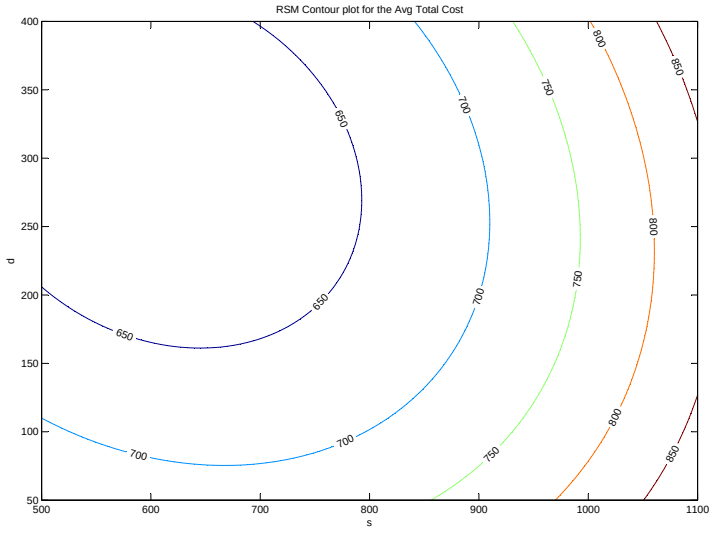


Figure 6.23: Scatterplot of regression metamodel for the total cost



(a) 3-D plot



(b) Contour plot

Figure 6.24: Regression metamodel for the expected total cost

The two regression metamodels are then used to solve the following constrained optimization problem:

$$\begin{aligned} \min \quad & E(C) \\ s.t. \quad & s(C) \leq T \end{aligned} \tag{6.3.2}$$

The threshold T is selected in such a way that feasible solutions always exist, i.e. we will never face an infeasible optimization problem; in particular we choose $T \in (30, 70)$. Moving the threshold in this range and each time solving the resulting optimization problem allow us to derive the estimated Pareto frontier, depicted in Fig. 6.26.

In order to study the variability of the optimal solutions (say (s^+, d^+)) corresponding to the estimated Pareto frontier in Fig. 6.26, we apply distribution-free bootstrap to the output of the simulation model in the following way:

i. Given the 3-dimensional matrix $C = (C_{ijr})$, we resample (with replacement) the n_e combinations of noise factors, given the control factor combination (s_i, d_i) and the r-th replication.

ii. Then, we apply the same sampling across all combinations of control factors (s, d) and all m replications.

iii. We repeat the bootstrap procedure B times; say $B = 100$.

Therefore, for each $b = 1, \ldots, B$, we obtain one matrix of bootstrapped costs, $C^{*b} = (C^{*b}_{ijr})$, to which we can apply the same steps we performed on the original matrix of simulated costs, C. So, we first fit B regression metamodels of the bootstrapped cost matrices, always including replicated scenarios, and then we derive the metamodels for the expected mean and standard deviation. Finally, similarly to what we did in the EOQ example when using distribution-free bootstrap, we evaluate each optimal solution (s^+, d^+) belonging to the original (i.e., not bootstrapped) Pareto frontier on each pair b of bootstrapped metamodels, $E(C^{*b}), s(C^{*b})$. We show the results obtained for two specific values of the threshold, namely $T = 37$ and $T = 67$, in Fig. 6.27. On the same figure, we also show the 90% confidence region, computed on the basis of the bootstrapped estimates of the optimal solutions; the normality assumption has been tested and accepted, as confirmed by the chi-square plot in Fig. 6.28.

It is noticeable that the optimal solution computed on the original model is not fully covered by the 'dotted area' identified by the bootstrapped solutions; this is especially evident in 6.27(a), which correspond the a smaller threshold T on the standard deviation. This might imply that the solution estimated through the original regression model is not so robust to possible variations in the environmental factors — then reflected by variations of the output, which are emulated through distribution-free bootstrap; this is even more evident for stricter constraints, i.e. when we ask for the standard deviation to be smaller.

6.4 Robust Simulation-Optimization using Kriging

Using the same assumptions we made in the previous section, we solve a robust simulation-optimization problem according to the methodology defined in Chapter 4. Referring to Fig. 4.1, we discuss results for the first approach we propose: it allows to fit the Kriging metamodels for the mean and standard deviation of the total cost directly from simulated data; whereas the other approach uses data from the simulation runs in a different way, that is to fit a single Kriging metamodel for the average total cost, then two Kriging metamodels are obtained for the mean and standard deviation of the output of interest, using predictions over the first metamodel.

We select the same design used in Section 6.3. Then we compute the expected value of the total cost as follows:

$$\overline{\overline{C}}_i = \frac{1}{n_{\mathbf{e}} m} \sum_{j=1}^{n_{\mathbf{e}}} \sum_{r=1}^{m} C_{ijr} \,, \tag{6.4.1}$$

where C_{ijr} denotes the simulated total cost, corresponding to the i-th control factor combination, j-th noise factor combination and r-th replication.
Since our final goal is to search for the optimal combination of the decision variables which minimizes the expected total cost, also keeping the standard deviation below a given threshold, we should account for both types of uncertainty; i.e., the total uncertainty should remain below a given threshold. This total uncertainty we estimate through

$$s^2(C_i) = \frac{1}{n_{\mathbf{e}} m - 1} \sum_{j=1}^{n_{\mathbf{e}}} \sum_{r=1}^{m} \left(C_{ijr} - \overline{\overline{C}}_i \right)^2 . \tag{6.4.2}$$

.

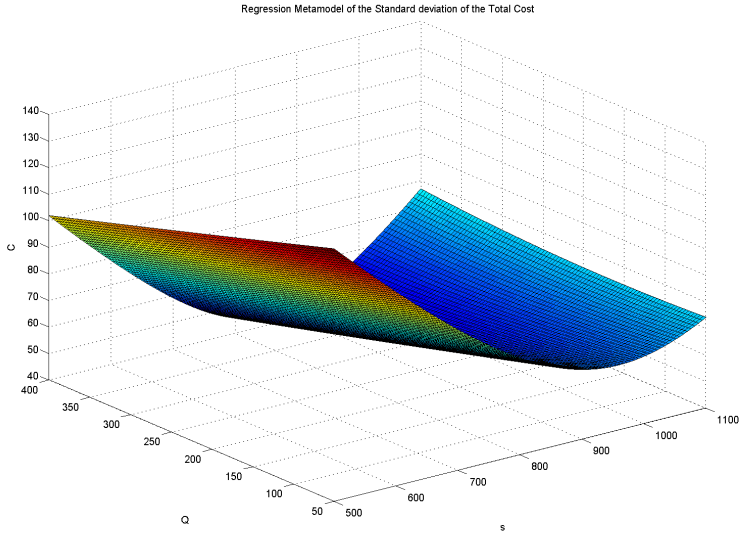
Based on these data, we fit a Kriging metamodels for the expected total cost and a Kriging metamodel for its standard deviation. We validate each of the metamodels using cross-validation: the resulting scatterplots are shown in Figs. 6.29-6.30, respectively; we also computed the relative prediction errors, obtaining $0.9925 \leq \hat{y}_1 \leq 1.0118$ for the expected cost and $0.8616 \leq \hat{y}_2 \leq 1.2324$ for the standard deviation of the cost, which allow us to accept both the metamodels as valid.

A 3-dimensional plot of the two metamodels, together with its contour plot, is reported in Figs. 6.31-6.32.

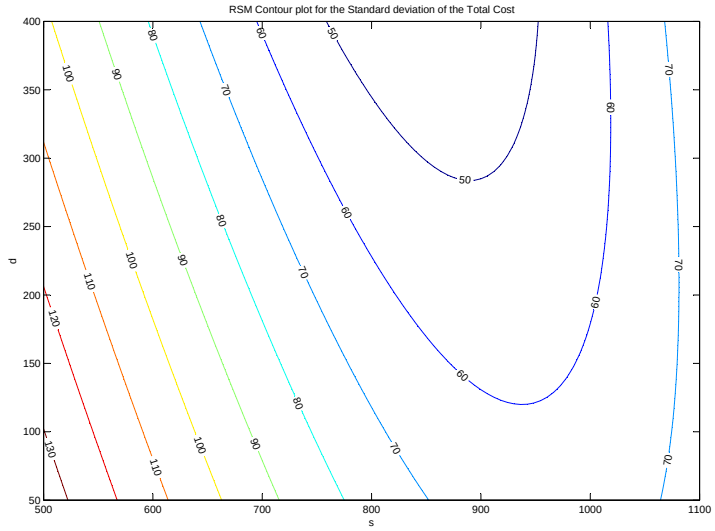
Based on the Kriging metamodels we have just fitted and validated, we solve the constrained optimization problem as formulated in Eq. 6.3.2; adopting the same choice for the threshold values, we estimate the Pareto frontier, which is plotted in Fig. 6.33.

Again we apply distribution-free bootstrap to the output of the simulation model as described in Section 6.3; then we compute the mean and standard deviation of the bootstrapped cost matrices, based on Eqs. 6.4.1-6.4.2. Fitting the Kriging metamodels over these bootstrapped data gives Fig. 6.34.

Using this set of B pairs of bootstrapped metamodels, we compute the predicted mean and standard deviation for each optimal solution (s^+, d^+) belonging to the original Pareto, obtaining $E(C^{*b}), s(C^{*b})$. We pick the same values for the threshold as we did in Section 6.3, to show the corresponding results in Fig. 6.35, where the 90% confidence region derived from the B bootstrapped optimal solutions is also plotted. The hypothesis of bivariate normality of the bootstrapped output is accepted, based on the results from the test described in Sect. 5.3 which results in the chi-square plot in Fig. 6.36.



(a) 3-D plot



(b) Contour plot

Figure 6.25: Regression metamodel for the standard deviation of the total cost

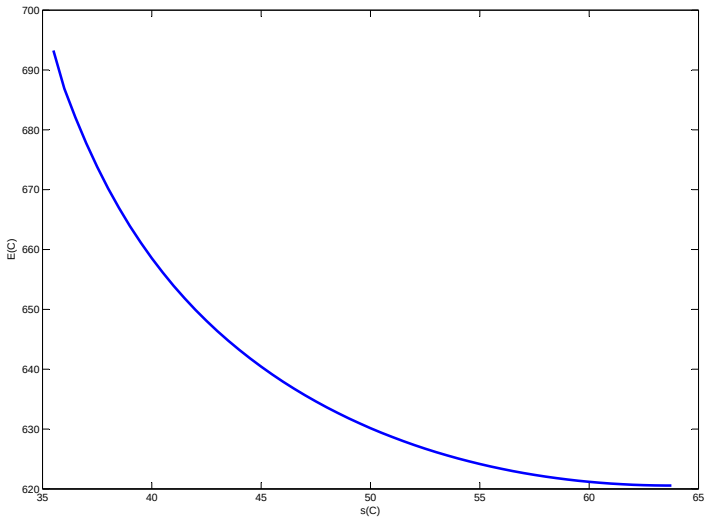
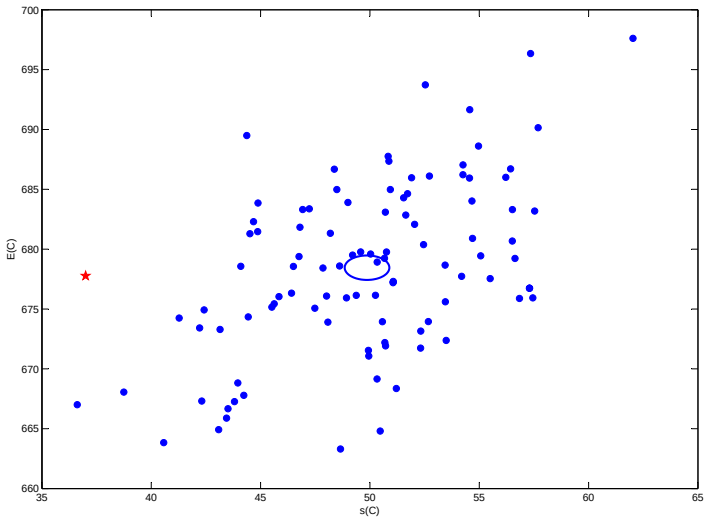
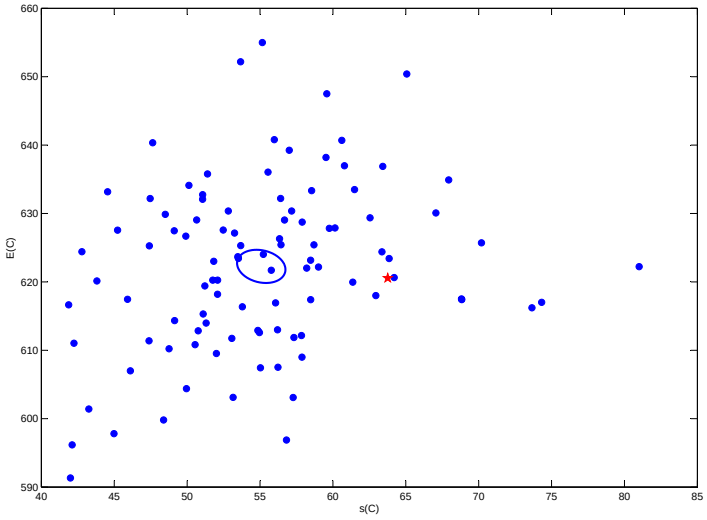


Figure 6.26: Estimated Pareto frontier based on regression metamodels



(a)



(b)

Figure 6.27: Bootstrapped estimations of the optimal solution and 90% confidence region based on regression metamodels of mean cost and standard deviation, for $T = 37$ (above) and $T = 67$ (below).

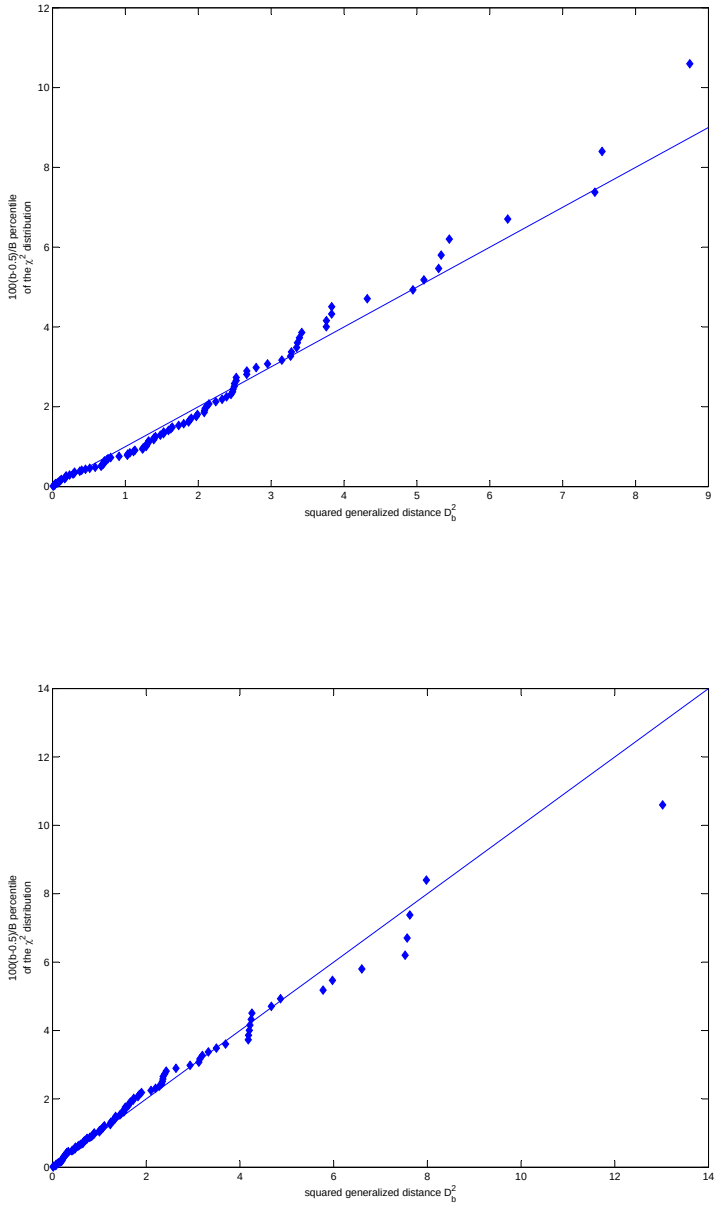


Figure 6.28: Chi-square plot of the bootstrapped estimates of mean cost and standard deviation, based on regression metamodels, for $T = 37$ (above) and $T = 67$ (below).

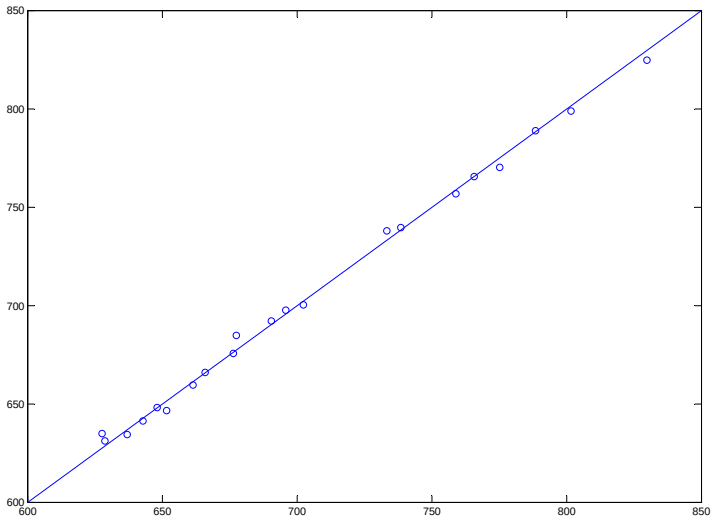


Figure 6.29: Scatterplot of Kriging metamodel for the expected total cost

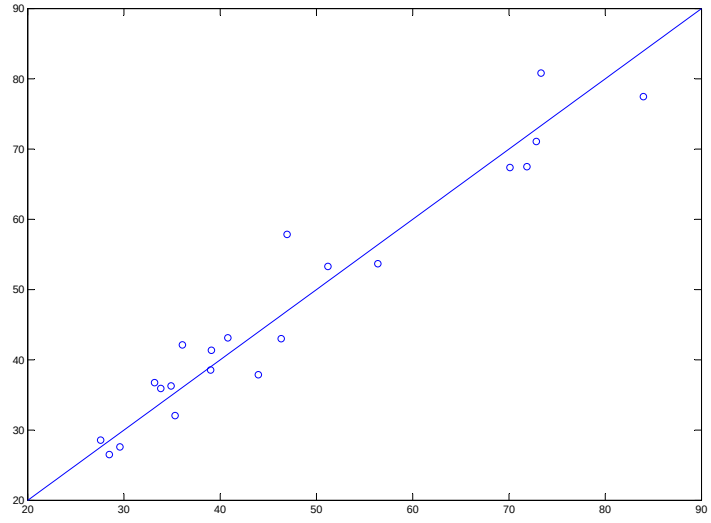
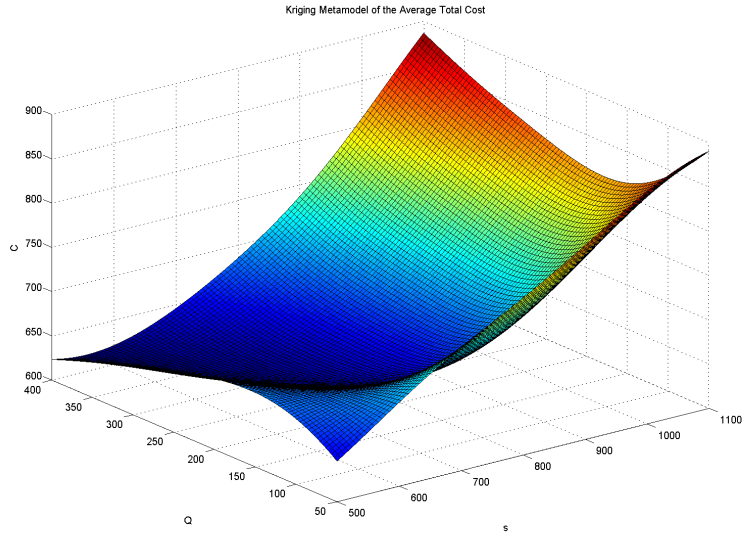
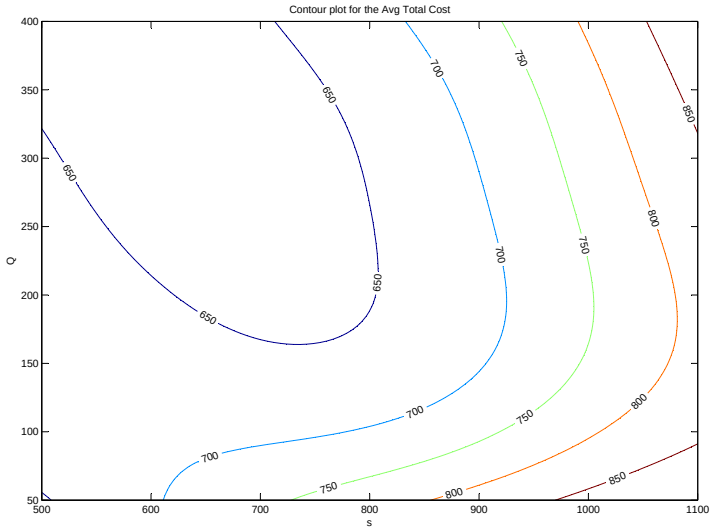


Figure 6.30: Scatterplot of Kriging metamodel for the standard deviation of the total cost

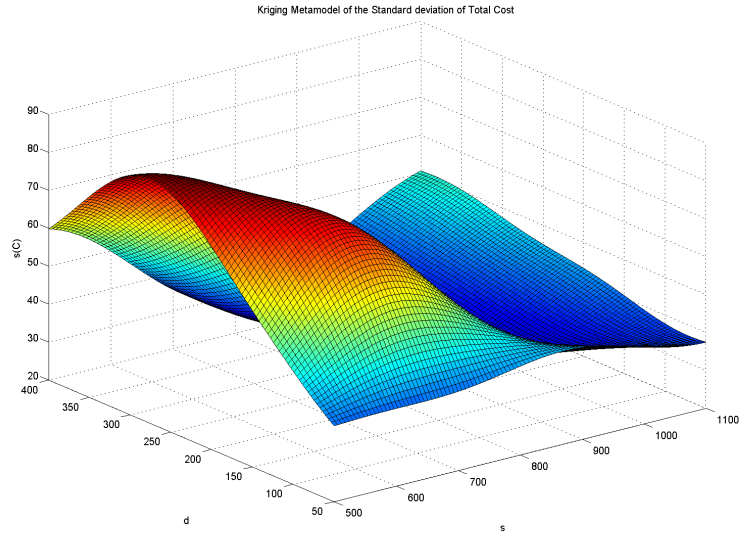


(a) 3-D plot

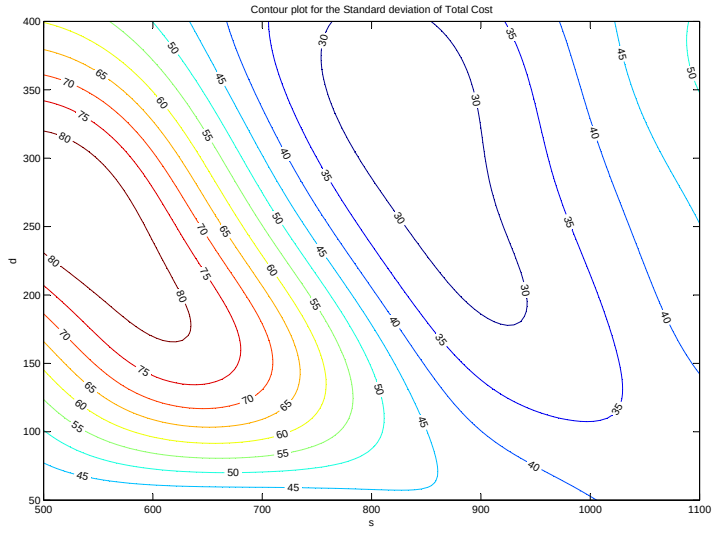


(b) Contour plot

Figure 6.31: Kriging metamodel for the expected total cost



(a) 3-D plot



(b) Contour plot

Figure 6.32: Kriging metamodel for the standard deviation of the total cost

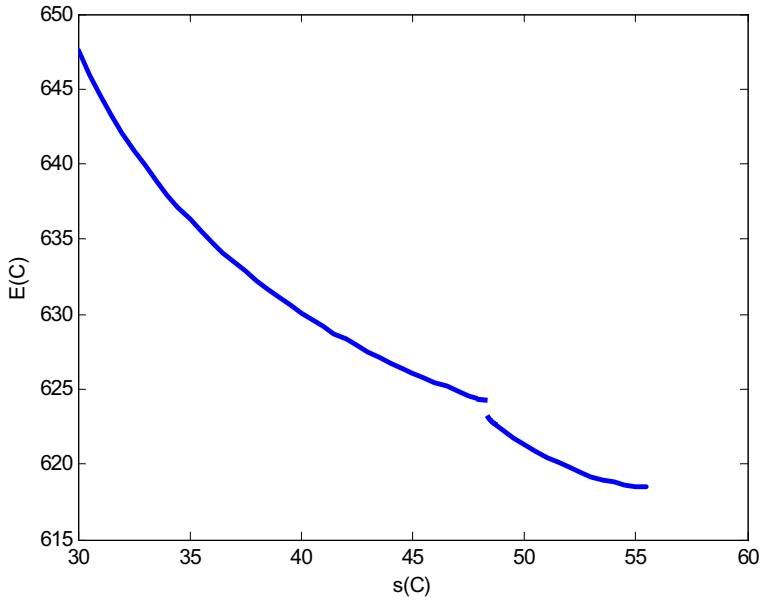
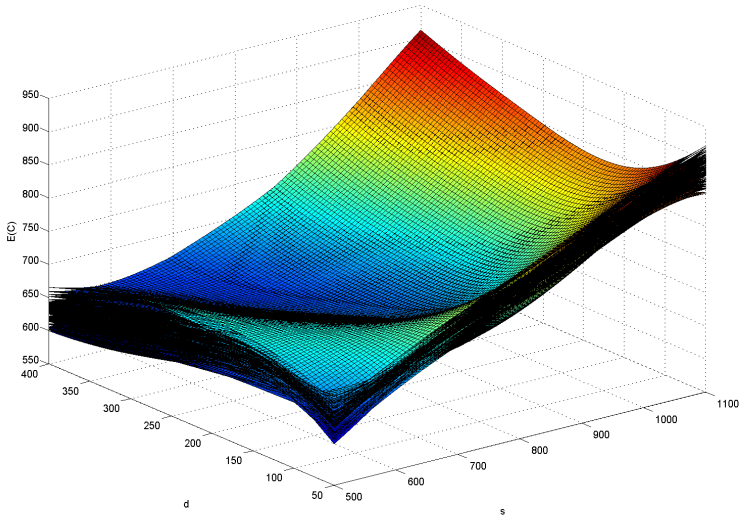
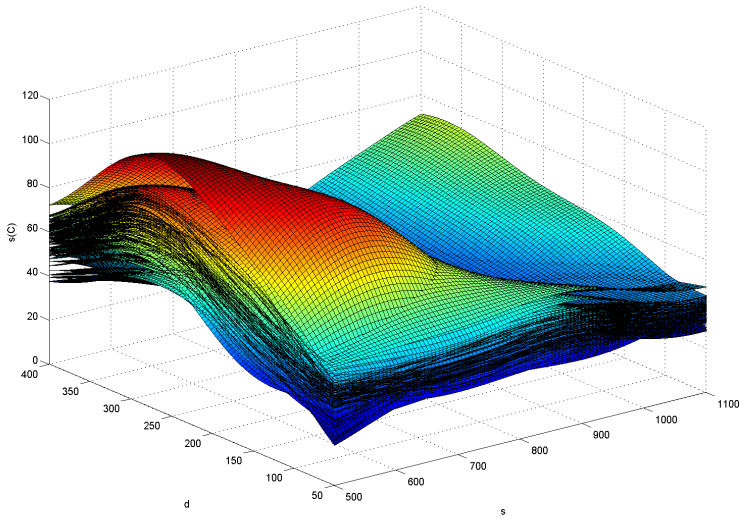


Figure 6.33: Estimated Pareto frontier based on Kriging metamodels



(a) Expected Cost



(b) Standard deviation of the cost

Figure 6.34: Bootstrapped Kriging metamodels

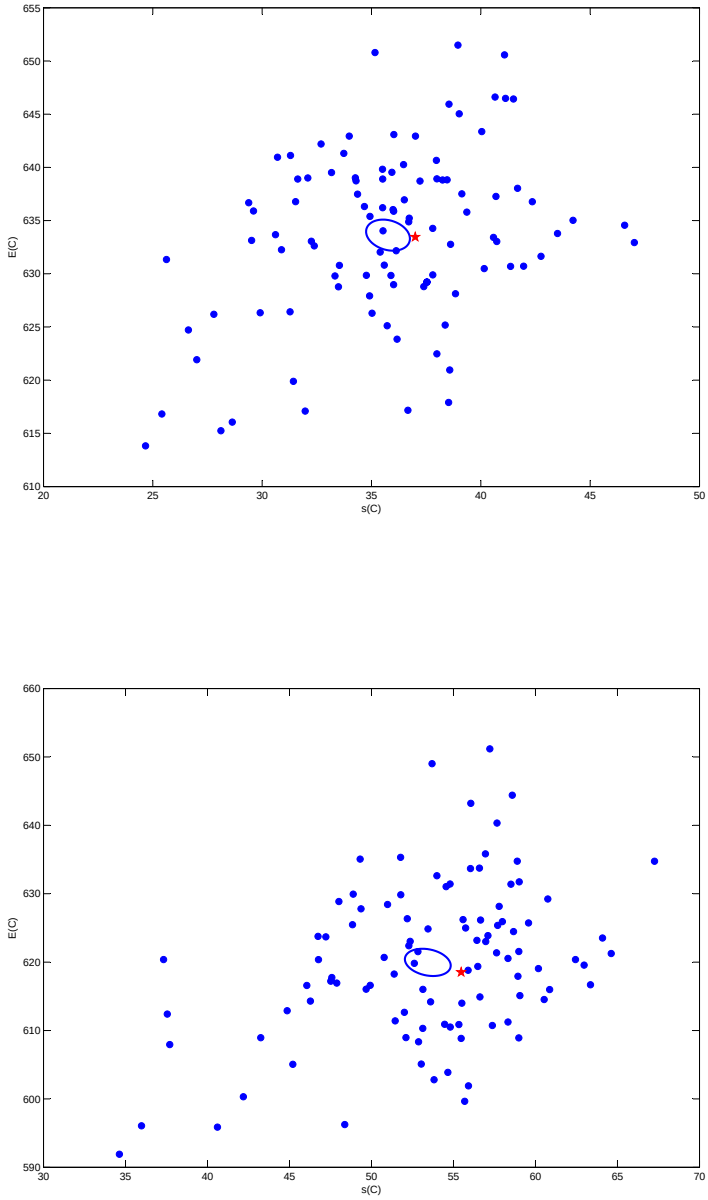


Figure 6.35: Bootstrapped estimations of the optimal solution and 90% confidence region, based on Kriging metamodels, for $T = 37$ (above) and $T = 67$ (below) — Approach 1.

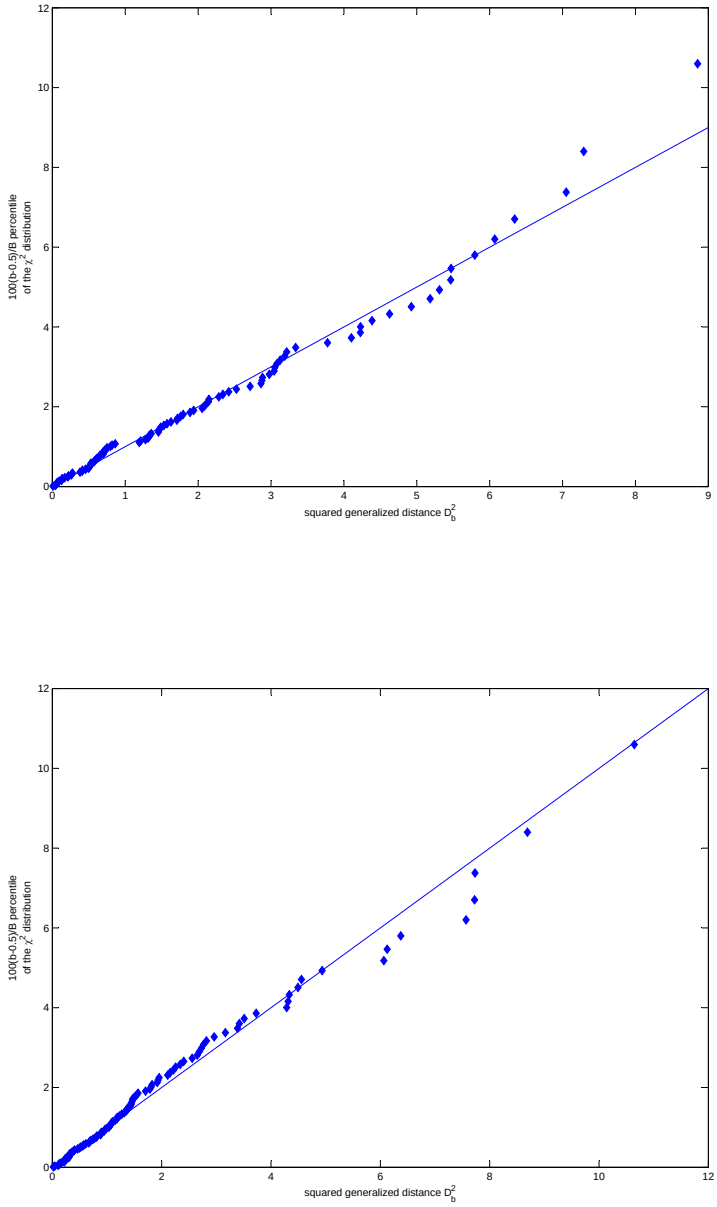


Figure 6.36: Chi-square plot of the bootstrapped estimates of mean cost and standard deviation, based on Kriging metamodels, for $T = 37$ (above) and $T = 67$ (below).

Chapter 7

Discussion and Conclusions

The research work that has led to the present dissertation originated from the study and analysis of metamodels; more precisely, we focussed our attention on analyzing Kriging metamodels, integrating them within an optimization framework. Then, the collaboration with Prof. Jack P.C. Kleijnen, from the CentER Graduate School, Tilburg University, has helped us in sketching a possible outline for the Ph.D. thesis. Such a work started with a wide study of the literature dealing with optimization methods involving some factors affected by uncertainty, usually referred to as *robust* optimization methods. During our study of the state-of-the-art on the topic, we notice that — despite the interest for this research area — there was a lack of methods to deal with robust optimization applied to simulated systems in general and, more specifically, with robust optimization supported by a metamodelling technique, when simulation asks for high computational costs. Therefore, we decided to follow this path, aiming to (at least partially) bridging this gap.

As a preliminary study, we focussed our attention on the approach proposed by Myers and Montgomery, to solve a robust optimization problem, through the use of regression metamodels, to approximate a real (i.e. not simulated) system. Then, their method served as a useful starting point for our research to go on, first of all by extending the robust procedure outlined by Myers and Montgomery to computer simulated systems and generalizing it through overcoming some restrictive assumptions: for instance, Myers and Montgomery identify only two levels for each factor, whereas we conduct a more refined analysis, allowing the input factor to take real values within a given range.

After some experiments on the robust methodology based on regression metamodels, we devoted most of our attention to study, design and implementing a novel approach to robust optimization using an alternative metamodelling technique, namely Kriging.

The approach proposed in this dissertation has been applied to two examples, that we chose as representative of both deterministic and stochastic simulation models:

1. The first application is based on the Economic Order Quantity (EOQ) model, a relatively simple model describing a deterministic system where we artificially introduce some uncertainty on the demand rate so to look for a robust solution w.r.t. possible variations in the parameter values.

2. The second example is based on the (s, S) inventory model, which is also well known in the management literature: it represents a stochastic system where the uncertainty is related to both some random variables characterizing the system and to the parameters defining the probability distribution of such variables.

We point out that, in general, the sources of uncertainty affecting a stochastic simulation model can be identified as belonging to two main classes: the aleatory uncertainty is inherent into the simulation model so being inescapable and it cannot be reduced without redesigning the real system (e.g., finding more reliable suppliers in inventory control, or more reliable machines in production control); the epistemic uncertainty is caused by the lack of information on the uncertain factors, so it can be reduced by collecting more information on those inputs (e.g., uncertain demand rates and lead time in inventory management). Simulation analysts may reduce the uncertainties in their estimates of the aleatory and epistemic uncertainty through 'bigger' sample sizes; i.e., more simulation runs or — in case of steady-state simulations — longer simulation runs. The robust optimization methodology that has been proposed in this Ph.D. thesis aims at accounting for both sources of uncertainty affecting the system. Moreover, it can be useful to further analyze the influence each uncertain factor has on the outputs of the system under study and evaluate to what extent they impact on the final optimal solution and on the variability of the outputs of interest.

The major contributions of the dissertation concerns the following aspects:

- Thorough study of possible source of uncertainty that might affect simulated systems and systematic analysis of the methodologies to solve robust optimization problems accounting for such uncertainty and for the influence it has on the outputs of the system we want to optimize.

- Analysis and integration of Kriging metamodels within a robust simulation-optimization framework for both deterministic and stochastic simulation models, including validation procedures of the approximation models which aim at ensuring an acceptable degree of accuracy.

- Comparison of the proposed approach with methods based on regression metamodels, derived from some techniques already discussed in the literature, but then extended and generalized.

- Wide and interdisciplinary applicability of the proposed methodology in the engineering and management contexts.

The dissertation has been divided into two parts: the first one aiming at describing the methodology proposed, moving from the state-of-the-art in the area of simulation-optimization and robust method to tackle uncertainty; the second one is devoted to

illustrate the results of the framework previously described, with applications to both deterministic and stochastic simulation models.
As far as the methodological part is concerned, some further remarks should be mentioned:

1. Kleijnen (2009) enlights Kriging metamodels to be suitable for approximating the input/output relationship of a (black-box) stochastic simulated system, the final goal being to solve an optimization problem replacing expensive simulation runs by cheap metamodelling predictions. The results of the present dissertation assess that Kriging is a promising approximation technique also in robust simulation-optimization. Moreover, it is also important to highlight that fitting a Kriging metamodel does not require any specific assumption on the output distribution (e.g. normality), as it is the case for regression metamodels, thus providing more flexibility.

2. Based on the results obtained on the two examples, discussed in Chapters 5-6, the evaluation of Kriging performance is twofold:

   (a) From the accuracy point of view, we notice that the validation procedure might reveal low accuracy (eventually failing at all) for metamodels either based on a small number of data samples or fitted over a wide region. Apart from that, the experimental results on (s, S) confirm the ability for Kriging metamodels to provide better approximation of highly nonlinear functions than regression models, as discussed in Sec. 6.2.

   (b) As for the optimization results, we notice that Kriging does not always outperform regression metamodels, depending on the region of interest and the function to be approximated. Notwithstanding this, Kriging appears to give promising results in the robust optimization process we propose; in fact, referring to the (s, S) inventory model, the estimated Pareto frontier based on Kriging is definitely non-dominated by the Pareto frontier estimated from regression metamodels. Moreover, Kriging seems to provide more robust solutions compared to those derived on the regression metamodels, as it emerges through our bootstrap analysis. A final comparison with the optimal solutions found by a well-known commercial software for simulation-optimization (namely, OptQuest) gives encouraging results: we check whether the optimal solutions estimated through Kriging are significantly different from those provided by OptQuest, by performing a t-test for the null hypothesis

   $$H_0 : \mu_{OptQuest} - \mu_{KG} = 0\,, \tag{7.0.1}$$

   similarly to what discussed in Sect. 6.2. The results of the test suggest to accept the null hypothesis with a significance level $\alpha = 0.05$. Moreover, we compute the relative errors on the optimal costs, noticing that they are rather small, being always less than 2%. With respect to this

subject, we point out that such approach has the property of requiring a relatively small number of simulation runs, compared with more classical simulation-optimization approaches which do not use metamodels: on one hand embedding metamodelling techniques has the advantage of saving computational time and resources; on the other hand, the use of surrogates — instead of the original simulation model — can provide less accurate results; therefore, a balance between these two aspects must be achieved.

Further analyses included more rigorous comparisons, through testing how significant is the difference in the optimal solutions between the methods discussed: it turned out that in some cases the results provided by the two approaches are not significantly different, but — when the difference becomes significant — Kriging usually provides better solutions.

3. In this dissertation we discuss some techniques to analyze and compare the robust optimization schemes that have been proposed (see Chapters 5-6), including a method to identify some confidence regions for the results obtained from optimizing the bootstrapped data. The confidence regions provides an idea of the variability of the outputs of each optimization process presented in our examples. More specifically, we are interested in the joint confidence region for the mean and the variance of a distribution, the outputs under study being the mean and the variance of the cost. In the proposed approach, following Kleijnen and Gaury (2003), we apply a method to derive a joint confidence region for a bivariate normal distribution (the outputs of interest in that paper were the work-in-progress (WIP) and a fill-rate measure); this may sometimes require to transform the variables (e.g., through a power transformation) in order to approach normality. An alternative approach might be the one proposed by Mandel and Betensky (2008), which does not necessarily require normality of data, and allows the computation of joint confidence interval for m parameters, using bootstrapping.

4. It is important to notice that the validation procedure highly depends on the Design of Experiments we use to fit the metamodel. In our computational experiments, we consider a static design, selected a priori over the region of interest and never updated while the optimization procedure goes on. An exception is represented by the nested LHS design, which we also adopted, requiring to add some new points to an initial design, in case it did not provide enough accurate metamodels; however, this leaves the optimization process aside anyhow. Other techniques are also proposed in the literature, which can accept to start with non-valid metamodels, whose quality and accuracy can be iteratively improved during the optimization process, by dynamically adding new sample points to the design (see Kleijnen et al. (2008), and Dellino et al. (2009)). On the basis of these notes, we might consider the methodology proposed in this dissertation as a starting point to be integrated in such frameworks.

5. In this work we focus only on continuous optimization, also relaxing some constraints on the decision variables which usually lead to tackle such problems within the area of discrete (or, more in general, mixed integer) optimization. Of course, a first — albeit rough — way to derive integer solutions would be to consider the real-valued optimal solution we obtained and then round it to the closer integer value; even though this is not so rigorous, sometimes it can also give good solutions in practical applications. Nevertheless, we should mention that the proposed approach can be easily extended to deal with integer constraints, provided that the DoE is made of only integer (or discrete) values for the corresponding variables; see Kleijnen et al. (2008), and Kleijnen (2008). In fact, it is important — in a simulation-optimization framework — to simulate scenarios which always guarantee input feasibility, to fit a metamodel over the corresponding I/O data. Then, the approach remains applicable once the Non-Linear Programming (NLP) solver (e.g. `fmincon`) is replaced by an appropriate Mixed-Integer NLP (MINLP) solver, which must be able to optimize black-box functions or metamodels; see, for instance, the IBM projects BONMIN (Basic Open-Source Nonlinear Mixed Integer programming), documented in Bonami and Lee (2007), and COUENNE (Convex Over and Under Envelopes for Nonlinear Estimation), discussed in details in Belotti et al. (2008).

6. Combining the last two remarks, we can develop a more complex and flexible framework, which enables to solve MINLP problems, taking advantage also of strategies to dynamically update the metamodels, this process being ruled both by metamodels' accuracy and through identifying promising regions during the optimization procedure.

Regarding the case studies and the related problem formulations, it is worth mentioning the following aspects as further steps to improve the current research also from this point of view:

1. Use of the Coefficient of Variation (CV) for the constraint on the variability of the (main) objective function (see del Castillo, 2007); it is defined (only for non-zero mean) as $CV = \sigma/\mu$ and it provides a normalized measure of dispersion of a probability distribution.

2. Adoption of different metrics to measure the service level, not necessarily alternative to each other, so that we can study, for instance, the system behaviour over distinct time periods; see Winston and Goldberg (2004).

3. Accounting for different distributions for the demand size and for the lead times; see (e.g.) Gallego et al. (2007).

4. Inclusion of some other constraints on input variables and/or on the output functions; e.g. inventory space constraints, number of shortage periods in the planning horizon.

5. Analysis of other sources of (both aleatory and epistemic) uncertainty, such as those affecting setup times and inventory levels, frequently appearing in real applications.

Bibliography

R. Al-Aomar. A robust simulation-based multicriteria optimization methodology. In E. Yücesan, C.-H. Snowdon, and J. M. Charnes, editors, *Proceedings of the Winter Simulation Conference*, pages 1931–1939, 2002.

T.T. Allen, M.A. Bernshteyn, and K. Kabiri-Bamoradian. Constructing meta-models for computer experiments. *Journal of Quality Technology*, 35(3):264–274, July 2003.

R. Allgor, S. Graves, and P. Xu. The benefits of re-evaluating real time fullfillment decisions. In *Proceedings of 2005 SMA Conference*, Singapore, 2005.

S. Bashyam and M. Fu. Optimization of (s, s) inventory systems with random lead times and a service level constraint. *Management Science*, 44(12):243–256, December 1998.

R. A. Bates, R. S. Kenett, D. M. Steinberg, and H. P. Wynn. Achieving robust design from computer simulations. *Quality Technology & Quantitative Management*, 3(2): 161–177, 2006.

B.M. Beamon. Measuring supply chain flexibility. In *Proceedings of the Eleventh Annual Conference of the Production and Operations Management Society*, San Antonio, Texas, 2000.

P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex minlp. IBM Research Report RC24620, IBM Research Division, August 2008.

A. Ben-Tal and A. Nemirovski. Selected topics in robust convex optimization. *Mathematical Programming*, 112(1):125–158, 2008.

D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

D. Bertsimas and A. Thiele. A robust optimization approach to supply chain management. In *Integer Programming and Combinatorial Optimization*, volume 3064 of *Lecture Notes in Computer Science*, pages 86–100. Springer, 2004.

D. Bertsimas and A. Thiele. A robust optimization approach to inventory theory. *Operations Research*, 54(1):150–168, 2006.

D. Bertsimas, O. Nohadani, and K. M. Teo. Robust nonconvex optimization for simulation-based problems. *Operations Research*, 2007. Submitted.

H. G. Beyer and B. Sendhoff. Robust optimization — a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33–34):3190–3218, 2007.

P. Bonami and J. Lee. *BONMIN Users' Manual*, August 2007.

E. Borgonovo and L. Peccati. Global sensitivity analysis in inventory management. *International Journal of Production Economics*, 108:302–313, 2007.

G.E.P. Box and K.B. Wilson. On the experimental attainment of optimum conditions. *Journal Royal Statistical Society, Series B*, 13(1):1–38, 1951.

W.J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, 2nd edition, 1980.

M.A. Darwish. Epq models with varying setup cost. *International Journal of Production Economics*, 113(1):297–306, 2008.

E. del Castillo. *Process Optimization: a Statistical Approach*. Springer, New York, 2007.

G. Dellino, J.P.C. Kleijnen, and C. Meloni. Robust optimization in simulation: Taguchi and response surface methodology. CentER Discussion Paper 2008-69, CentER - Tilburg University, July 2008a. Submitted to IJPE.

G. Dellino, J.P.C. Kleijnen, and C. Meloni. Robust optimization in simulation: Taguchi and krige combined. Working paper, 2008b.

G. Dellino, P. Lino, C. Meloni, and A. Rizzo. Kriging metamodel management in the design optimization of a CNG injection system. *Mathematics and Computers in Simulation*, 2009. to appear.

D. den Hertog, J.P.C. Kleijnen, and A.Y.D. Siem. The correct kriging variance estimated by bootstrapping. *Journal of the Operational Research Society*, 57(4): 400–409, April 2006.

B. Efron and R.J. Tibshirani. *An introduction to the bootstrap*. Chapman & Hall, New York, 1993.

B. El-Haik and R. Al-Aomar. *Simulation-based lean six-sigma design for six-sigma*. John Wiley & Sons, Hoboken, New Jersey, 2006.

K.-T. Fang, R. Li, and A. Sudjianto. *Design and Modeling for Computer Experiments*. Chapman & Hall, Boca Raton, Florida, 2006.

E. Fleish and C. Tellkamp. Inventory inaccuracy and supply chain performance: a simulation study of a retail supply chain. *International Journal of Production Economics*, 95(3):373–385, 2005.

G. Gallego and A. Scheller-Wolf. Capacitated inventory problems with fixed order costs: some optimal policy structure. *European Journal of Operations Research*, 126:603–613, 2000.

G. Gallego, K. Katircioglu, and B. Ramachandran. Inventory management under highly uncertain demand. *Operations Reseach Letters*, 35:281–289, 2007.

W.H. Greene. *Econometric Analysis*. Prentice Hall, 5th edition, 2003.

F. S. Hillier and G. J. Lieberman. *Introduction to Operations Research*. McGraw Hill, 7th edition, 2001.

B.G.M. Husslage, E.R. van Dam, and D. den Hertog. Nested maximin latin hypercube designs in two dimensions. CentER Discussion Paper 2005-79, Tilburg University, Tilburg, 2005.

R.L. Iman, J.M. Davenport, and D.K. Zeigler. Latin hypercube sampling (program user's guide). Technical Report SAND79-1473, SANDIA National Laboratories, Albuquerque, New Mexico, 1979.

B. Iooss, M. Ribatet, and A. Marrel. Global sensitivity analysis of stochastic computer models with generalized additive models. Working paper, CEA, Cadarache, Saint Paul lez Durance, France, 2007.

R. Jin, X. Du, and W. Chen. The use of metamodeling techniques for optimization under uncertainty. *Structural Multidisciplinary Optimization*, 25:99–116, 2003.

N.L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*, volume 1. John Wiley & Sons, 2nd edition, 1994.

R.A. Johnson and D.W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall International, 1992.

W. D. Kelton, R. P. Sadowski, and D. T. Sturrock. *Simulation with Arena*. McGraw Hill, 4th edition, 2007.

J. P. C. Kleijnen. *Design and Analysis of Simulation Experiments*. Springer Science + Business Media, New York, 2008.

J.P.C. Kleijnen. Cross-validation using the t statistic. *European Journal of Operational Research*, 13(2):133–141, 1983a.

J.P.C. Kleijnen. Risk analysis and sensitivity analysis: antithesis or synthesis. *Simuletter*, 14(1-4):64–72, 1983b.

J.P.C. Kleijnen. *Statistical tools for simulation practitioners.* Marcel Dekker, New York, 1987.

J.P.C. Kleijnen. Sensitivity analysis versus uncertainty analysis: when to use what? In J. Grasman and G. van Straten, editors, *Predictability and nonlinear modelling in natural sciences and economics*, pages 322–333. Kluwer, Dordrecht, The Netherlands, 1994.

J.P.C. Kleijnen. Kriging metamodeling in simulation: a review. *European Journal of Operations Research*, 192(3):707–716, February 2009.

J.P.C. Kleijnen and E.G.A. Gaury. Short-term robustness of production-management systems: a case study. *European Journal of Operational Research*, 148(2):452–465, July 2003.

J.P.C. Kleijnen and J.C. Helton. Statistical analyses of scatter plots to identify important factors in large-scale simulations: review and comparison of techniques. *Reliability Engineering and Systems Safety*, 65(2):147–185, 1999.

J.P.C. Kleijnen and R.G. Sargent. A methodology for the fitting and validation of metamodels in simulation. *European Journal of Operational Research*, 120(1):14–29, 2000.

J.P.C. Kleijnen and W.C.M. van Beers. Application-driven sequential designs for simulation experiments: Kriging metamodeling. *Journal of the Operational Research Society*, 55(9):876–883, 2004.

J.P.C. Kleijnen, W. van Beers, and I. van Nieuwenhuyse. Constrained optimization in simulation: novel approach. Center discussion paper, CentER, Tilburg University, Tilburg, The Netherlands, 2008.

A.G. Kök and K.H. Shang. Inspection and replenishment policies for systems with inventory record inaccuracy. *Manufacturing and Service Operations Management*, 9(2):185–205, 2007.

D.G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 52(6):119–139, December 1951.

A.M. Law. *Simulation Modeling and Analysis.* McGraw-Hill, 4th edition, 2007.

H. L. Lee and S. Nahmias. Single-product, single-location models. In S. C. Graves, A. H. G. Rinnooy Kan, and P. H. Zipkin, editors, *Logistics of Production and Inventory — Handbooks in Operations Research and Management Science*, volume 4, chapter 1, pages 3–55. Elsevier Science, 1993.

K.H. Lee and G.J. Park. A global robust optimization using kriging based approximation model. *Journal of the Japan Society of Mechanical Engineering*, 49(3): 779–788, 2006.

Y. Lee and J.A. Nelder. Robust design via generalized linear models. *Journal of Quality Technology*, 35(1):2–12, January 2003.

S.N. Lophaven, H.B. Nielsen, and J. Søndergaard. DACE: a MATLAB kriging toolbox (version 2.0). Technical Report IMM-TR-2002-12, Technical University of Denmark, Lyngby, August 2002.

M. Mandel and R.A. Betensky. Simultaneous confidence intervals based on the percentile bootstrap approach. *Computational Statistics & Data Analysis*, 52:2158–2165, 2008.

M. Meckesheimer, A.J. Booker, R.R. Barton, and T.W. Simpson. Computationally inexpensive metamodel assessment strategies. *AIAA Journal*, 40(10):2053–2060, October 2002.

K.M. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, Boston, 1999.

G. Miró-Quesada and E. del Castillo. Two approached for improving the dual response method in robust parameter design. *Journal of Quality Technology*, 36(2):154–168, April 2004.

R.C. Mittelhammer. *Mathematical Statistics for Economics and Business*. Springer, 1996.

D.C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 7th edition, 2009.

J. Mula, R. Poler, J.P. García-Sabater, and F.C. Lario. Models for production planning under uncertainty : a review. *International Journal of Production Economics*, 103(1):271–285, 2006.

R. H. Myers and D. C. Montgomery. *Response Surface Methodology: Process and Product Optimization using Design Experiments*. John Wiley & Sons, New York, 2002.

R. H. Myers, A. I. Khuri, and G. Vining. Response surface alternatives to the taguchi robust parameter design approach. *The American Statistician*, 46(2):131–139, 1992.

V.N. Nair, B. Abraham, J. MacKay, J.A. Nelder, G. Box, M.S. Phadke, R.N. Kacker, J. Sacks, W.J. Welch, T.J. Lorenzen, A.C. Shoemaker, K.L. Tsui, J.M. Lucas, S. Taguchi, R.H. Myers, G.G. Vining, and C.F.J. Wu. Taguchi's parameter design: A panel discussion. *Technometrics*, 34(2):127–161, May 1992.

J.J. Neale, B.T. Tomlin, and S.P. Willems. The role of inventory in superior supply chain performance. In T.P. Harrison, H.L. Lee, and J.J. Neale, editors, *The practice of Supply Chain Management: where theory and application converge*, chapter 3, pages 31–60. Springer Science & Business Media, 2004.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2006.

J.T. (chair) Oden. *Revolutionizing engineering science through simulation*. Blue Ribbon Panel on Simulation-Based Engineering Science. National Science Foundation (NSF), 2006.

G.J. Park, T.H. Lee, K.H. Lee, and K.H. Hwang. Robust design: An overview. *AIAA Journal*, 44(1):181–191, January 2006.

T.J. Robinson, C.M. Borror, and R.H. Myers. Robust parameter design: a review. *Quality and Reliability Engineering International Journal*, 20:81–101, 2004.

Rockwell Automation. *OptQuest for Arena — User's Guide*. Rockwell Software Inc. and OptTek Systems Inc., 2004.

R.S. Russell and B.W. III Taylor. *Operations Management along the Supply Chain*. John Wiley & Sons, 6th edition, 2008.

J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, November 1989.

S. M. Sanchez. Robust design: Seeking the best of all possible worlds. In J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, editors, *Proceedings of the Winter Simulation Conference*, pages 69–76, 2000.

T.J. Santner, B.J. Williams, and W.I. Notz. *The design and analysis of computer experiments*. Springer-Verlag, New York, 2003.

B.S. Sarker and E.R. Coates. Manufacturing setup cost reduction under variable lead times and finite opportunities for investment. *International Journal of Production Economics*, 49:237–247, 1997.

J.F. Shapiro. *Modeling the Supply Chain*. Duxbury, Pacific Grove, California, 2nd edition, 2007.

M. Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151, May 1987.

E. Stinstra and D. den Hertog. Robust optimization using computer experiments. *European Journal of Operational Research*, 191(3):816–837, December 2008.

L.P. Swiler and A.A. Giunta. Aleatory and epistemic uncertainty quantification for engineering applications. Technical Report SAND2007-2670C, SANDIA National Laboratories, Albuquerque, New Mexico, July 2007.

G. Taguchi. *System of experimental designs*, volume 1–2. UNIPUB/ Krauss International,White Plains, New York, 1987.

The MathWorks Inc. *Optimization Toolbox User's Guide*. Natick, Massachussetts, 2005a.

The MathWorks Inc. *Statistics Toolbox*. Natick, Massachussetts, 2005b.

The MathWorks Inc. *Symbolic Math Toolbox User's Guide*. Natick, Massachussetts, 2007.

H.C. Tijms and H. Groenevelt. Simple approximations for the reorder point in periodic and continuous review (s, s) inventory systems with service level constraints. *European Journal of Operational Research*, 17(2):175–190, August 1984.

M. W. Trosset. Taguchi and robust optimization. Technical Report 96-31, Dept. of Computational & Applied Mathematics, Rice University, Houston, Texas, 1997.

H. Van Landeghem and H. Vanmaele. Robust planning: a new paradigm for demand chain planning. *Journal of Operations Management*, 20(6):769–783, November 2002.

W.L. Winston and J.B. Goldberg. *Operations Research: Applications and Algorithms*. Duxbury Press, Belmont, CA, 4th edition, 2004.

C.F.J. Wu and M. Hamada. *Experiments: planning, analysis, and parameter design optimization*. John Wiley & Sons, New York, 2000.

G. Yu. Robust economic order quantity models. *European Journal of Operational Research*, 100(3):482–493, 1997.

Q. Zhang, M.A. Vonderembse, and J.S. Lim. Manufacturing flexibility: defining and analyzing relationships among competence, capability, and customer satisfaction. *Journal of Operations Management*, 21(2):173–191, March 2003.

Y. Zhang. A general robust-optimization formulation for nonlinear programming. Technical Report 04-13, Department of Computational and Applied Mathematics, Rice University, Houston, USA, July 2004.

X. Zhao, F. Fan, X. Liu, and J. Xie. Storage-space capacitated inventory system with (r, q) policies. *Operations Research*, 55(5):854–865, September-October 2007.

P. H. Zipkin. *Foundations of Inventory Management*. McGrawHill, 2000.