Carnegie Mellon University

CARNEGIE INSTITUTE OF TECHNOLOGY

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF Doctor of Philosophy

TITLE Interactions of Uncertainty and Optimization: Theory, Algorithms, and Applications to Chemical Site Operations

PRESENTED BY Satyajith Amaran

ACCEPTED BY THE DEPARTMENT OF

Chemical Engineering

NIKOLAOS SAHINIDIS	9/19/14
NIKOLAOS SAHINIDIS, ADVISOR	DATE
LORENZ BIEGLER	9/19/14
LORENZ BIEGLER, DEPARTMENT HEAD	DATE

APPROVED BY THE COLLEGE COUNCIL

VIJAYAKUMAR BHAGAVATULA	9/19/14
DEAN	DATE

INTERACTIONS OF UNCERTAINTY AND OPTIMIZATION: THEORY, ALGORITHMS, AND APPLICATIONS TO CHEMICAL SITE OPERATIONS

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Chemical Engineering

Satyajith Amaran

M.S., Chemical Engineering, Carnegie Mellon University
B.Tech., Chemical Engineering, National Institute of Technology Karnataka

Carnegie Mellon University
Pittsburgh PA

September 2014

Interactions of Uncertainty and Optimization: Theory, Algorithms, and Applications to Chemical Site Operations, Ph.D. Thesis

To Akka, Mom, Dad, and Riteja

ABSTRACT

This thesis explores different paradigms for incorporating uncertainty with optimization frameworks for applications in chemical engineering and site-wide operations. First, we address the simulation optimization problem, which deals with the search for optimal input parameters to black-box stochastic simulations which are potentially expensive to evaluate. We include a comprehensive literature survey of the state-of-the-art in the area, propose a new provably convergent trust region-based algorithm, and discuss implementation details along with extensive computational experience, including examples for chemical engineering applications.

Next, we look at the problem of long-term site-wide maintenance turnaround planning. Turnarounds involve the disruption of production for significant periods of time, and may incur enormous costs in terms of maintenance manpower as well as lost sales. The problem involves (1) the simulation of profit deterioration due to wear and tear followed by the determination of how frequently a particular turnaround should take place; and (2) the consideration of site network structure and turnaround frequencies to determine how turnarounds of different plants may be coordinated over a long-term horizon. We investigate two mixed-integer models, the first of which determines optimal frequencies of individual plant turnarounds, while the second considers maximizing long-term profit through coordination of turnarounds across the site.

We then turn to more conventional methods of dealing with optimization under uncertainty, and make use of a combined robust optimization and stochastic programming approach to medium-term maintenance planning in integrated chemical sites. The nature of the uncertainty considered affects two aspects of maintenance planning, one of which is most suitably addressed through a robust optimization framework, while the other is better handled with stochastic programming models.

In summary, we highlight the importance of considering uncertainty in optimization as well as the choice of approach or paradigm used through chemical engineering applications that span varied domains and time scales.

There is one more thing. It's been emotional.
— Big Chris, *Lock, Stock, and Two Smoking Barrels (1998)*

ACKNOWLEDGMENTS

I have been influenced in profound ways over the last six years with regard to research, culture and language, and ways of life by many people.

I am deeply indebted to Prof. Nick Sahinidis, my research advisor through both my Masters and Doctorate programs. Nick, you gave me the opportunity to work on some very interesting, open-ended, and wide-ranging problems, and have actively supported my professional exposure through conferences and internships. I am privileged to have seen your domain expertise, computing mastery, and amazing intuition up close. Your work ethic, responsiveness, communication, and efficiency are qualities that I aspire to. I appreciate the research freedom you gave me and have enjoyed working in your group very, very much.

I would like to thank the members of my doctoral committee—Prof. Lorenz T. Biegler, Prof. Ignacio E. Grossmann, Prof. Geoffrey J. Gordon, Dr. Bikram Sharda, and Dr. Scott J. Bury. They provided valuable comments that helped both in the evolution of this work, as well as in the quality of the final manuscript. Professors Grossmann and Biegler have been academic role models to me in more ways than one.

I would like to thank NSF award 1033661 and The Dow Chemical Company for supporting my research. I am grateful to Bikram Sharda, John Wassick, and Alex Kalos for providing me with the opportunity to complete a summer internship with The Dow Chemical Company. I really enjoyed working with the dynamic research team at Freeport, Texas.

A number of others have enabled this research. Anshul Agarwal and Max Fahrenkopf were extremely helpful in providing code and data for case studies. Prof. Ted Allen and Prof. K.-H. Chang were kind in sharing their implementations. I am very thankful to Tong Zhang (who hails from a part of China close to Mongolia) who is a collaborator for the work on medium-term turnaround planning. Good luck in your future endeavors, Tong!

I've had some great conversations relating to optimization and PSE with Qi, John Eason, Alison, Max, Pedro Amorim, Pablo, Bruno, Apurva, Yash, Sree, Francisco, Rob, and Luis Miguel—thank you for being willing trampolines for my ideas and questions. I've also consulted on many occasions the resident TEXperts Tor Aksel, Erik Esche, and Siddarth Raman; presentation pundit Alison Cozad; and Linux gurus Arvind Mukundan and Simon Markowski. You introduced and helped me become comfortable with the primary non-programming tools that I use on an everyday basis.

My journey was incredibly smooth thanks to Cindy, Janet, Julie, Justin, Laura, Shannon and Toni, the stalwarts at CMU. You have been the most helpful, efficient, and friendly staff I could have asked for.

I am grateful for having been a part of a very close-knit department. Linlin, German and Tor, I've spent pretty much the entire length of my stay at CMU with the three of you and what a great time it's been. Now I will always need coffee at 3pm. A shout out to Aida, Alison, Deepak, Fernando, Haoqi, Keith, Mustafa, (small) Nick, Panos, Sree, Tong, Yan, Yash, and Zach—my fellow Sahinidis group members for some good times together—be it the mid-afternoon office prattle, the late nights on the upper floors of Doherty Hall, or expeditions to local bars and restaurants ... you know, nothing too far way. Alejandra, Annia, Axel, Bethany, Edna, Ellis, Francisco, Ines, Irene, Jake, Javier (aka J-Law), Jens, John Eason, Juan Morinelly, Jun, Mariano, Markus, Martijn, Max, Monica, Pablo, Qi, Ricardo, Rob, Sumit, Svenja, Vijay, Xue, Zixi, and the rest of the PSE gang—I will always have lots of fantastic memories of my time with you. Salut!

A hat tip to Fuego FC, foosball and table tennis @ happy hour, and my tennis buddies Nick, Pedro, and Takshak. I always had the perfect respite from research through these interactions.

My home away from home was at the Youth Hostel and the Quiz Club@CMU. Full-time residents Varun, Utsav, and Supreeth and part-time residents Aranya, Harini, Samrat, and Erle—you have been brilliant roommates. I've grown, learned, and thoroughly enjoyed living in Pittsburgh because of your company. An attempt to précis all we've been through would be an exercise in futility but for the convenient four-word reduction, *quod non interficet firmat*. Hugs and high-fives to Ajit, Apurva, Arti, Ashwati, Budling, Chitra, Harsha, Janani, Keshav, Lavanya, Maddali, Nishtha,

Ramya, Sandeep, Siddharth Garg, SL, Suha, Suyash, Uday, Vinod, and the rest of the gang. Friday nights will never be the same without you guys.

Anand, Ashwin, Bhavya, Shamitha, Vig, and Vinayak—I am lucky to have friends like you; thank you for all the support.

I am very grateful to Prof. N Krishnamurthy who provided me with immense support, both during my Masters and my PhD. Dear Uncle K., I would not have been able to start, let alone complete this journey without your support.

Riteja, this adventure through my doctorate has coincided with our adventure together. If I am a better person than when I started, it is because of you. This thesis represents a lot of what I have learned in the past few years; yet, it is a piddling amount compared with what I've learned from you. Now that I am finally Ph.*inishe*D., I can't wait to start the next chapter together.

Akka, Mom, and Dad, none of these experiences that I have enjoyed and will treasure could have happened but for your sustained belief in me. I am most grateful for your prayers, support, patience, and love.

CONTENTS

LIST OF FIGURES AND ILLUSTRATIONS

LIST OF TABLES

LIST OF ALGORITHMS

Part I

INTRODUCTION

I just sit at my typewriter and curse a bit.
— P. G. Wodehouse, when asked about his writing technique

1.1 INTRODUCTION

The goal of this thesis is to contribute to the process systems engineering area through novel theory and methods, and their application to domains of relevance in the field. The broad theme of the thesis involves the development of mathematical approaches that consider uncertainty in the context of optimization, and the investigation of both model-free and model-based approaches for various problems in chemical engineering.

The interaction of uncertainty with optimization can take place in multiple ways, three of which are illustrated in Figure 1.1.

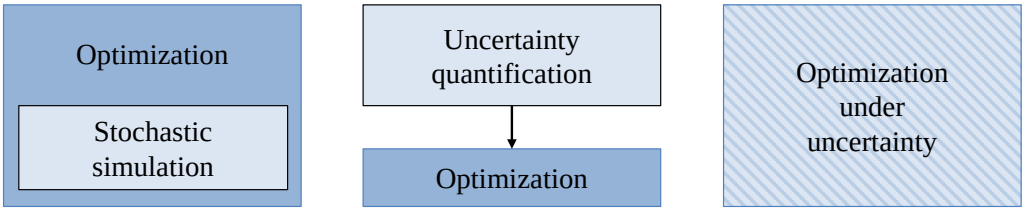


Figure 1.1: Interactions of uncertainty and optimization

The first paradigm illustrates the case of stochastic black-box optimization, where we would like to optimize directly over a stochastic simulation. Here, the simulation model is not available to the optimization routine, and therefore the optimizer has to progress through simulation-based iterations. In addition, many applications require the use of high-fidelity simulations that have embedded uncertainty, describe underlying phenomena in detail, and may be expensive to evaluate, thus necessitating a model-free optimization approach. The adjustment of input parameters to the simulation affects the outputs, and these are typically tied to some sort of performance or economic measure. The goal in this context is to identify optimal input parameter values.

A common way to combine uncertainty and optimization is to perform uncertainty quantification as a pre-processing step to optimization, and this is illustrated through the second paradigm. Outputs from optimization routines are affected by both the parameters as well as the assumptions made in the model. Here, the uncertainty quantification complements the optimization routine by supplying it with good parameter values. In our context, uncertainty quantification may involve analyzing production, reliability, and forecast data in a production network through simulations to determine suitable estimates of cost coefficients, demands, or other parameters.

When we make use of an algebraic description of the model for optimization, an optimizer is typically concerned with manipulating the decision variables to minimize, for example, some objective, while satisfying some feasibility conditions. The constants and parameters that go into the model, however, may have a significant effect on optimality, sensitivity of solution, and feasibility. The third paradigm shown in Figure 1.1 involves robust optimization and stochastic programming approaches, both of which simultaneously consider parameter uncertainty as well as optimization within a single modeling framework.

To motivate the need to incorporate various kinds of uncertainties in chemical engineering optimization problems, we consider integrated chemical site operations. Figure 1.2 illustrates the structure of a large integrated site.

A large-scale integrated chemical site constitutes a number of individual production units that are either connected to each other directly or through buffer storage capacities. These production units supply raw material to other units and produce final products that are ready to be shipped to end users. The tight integration of this network of plants provides synergistic opportunities for sharing raw materials, products, process and business information, domain knowledge, energy, utilities, manpower, safety infrastructure, and transportation.

In addition, integrated sites may also benefit from holistic, long-term, medium-term and short-term maintenance turnaround planning. Maintenance tasks represent disruptions in production, and necessitate intelligent decision-making for production planning, demand satisfaction, and manpower availability.

Maintenance turnarounds involve shutting down plants for inspection, monitoring, cleaning, structural reinforcement and overhaul, and last 1–8 weeks. They

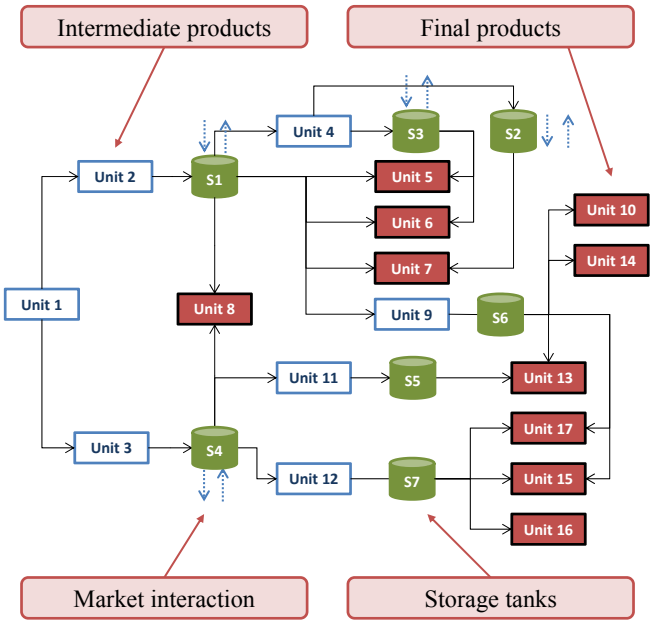


Figure 1.2: Example of an integrated chemical site network

incur substantial costs, and Figure 1.3 shows that in 2008, 408 turnarounds in active projects in the chemical process industry in North America were valued at a cumulative amount of around $1,034MM (North American Chemical Processing Industry Maintenance Turnarounds 2014). Thus, maintenance optimization has potential for substantially reducing operating costs for a chemical plant, and increasing profit by increasing availability, reliability, and production.

On a short-term time-scale, there may be several minor maintenance or upkeep functions that cause production levels to fluctuate on an hourly or daily basis, and this leads to the need for buffer storage to hedge against this uncertainty in production. This production variability may be modeled through a stochastic simulation framework, and a key decision may involve the determination of base stock levels in the buffer inventory tanks. Black-box simulation optimization methods may be used to make such decisions (Section 1.2).

On a medium-term time-scale, planning of maintenance tasks and arranging for maintenance manpower is done several months prior to a turnaround. As observed by Lenahan (1999) and Narayan (2004), a large portion of the turnaround work scope is hidden due to inaccessibility to plant equipment, and the major uncertainty is in the duration of the turnaround. Production planning decisions are commonly addressed by mixed-integer programming formulations, and extensions to these, in

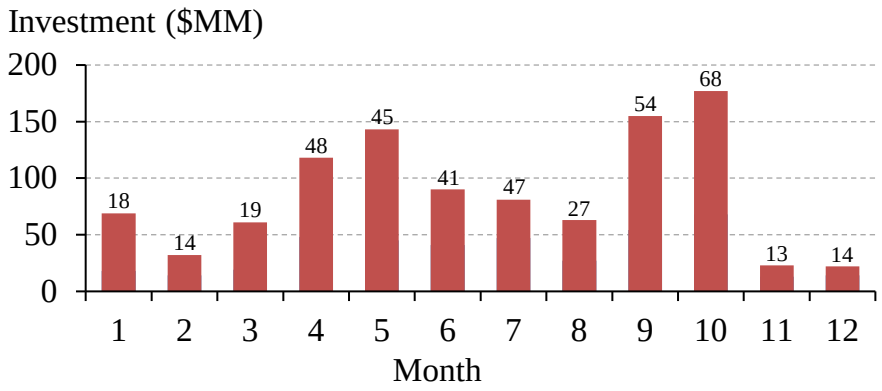


Figure 1.3: Value and number of North American maintenance turnarounds in the chemical processing industry for 2008 (`http://www.industrialinfo.com/media/downloadMedia.jsp?mediaId=344733`, accessed February 21, 2014)

terms of robust optimization (Ben-Tal et al. 2009) and stochastic programming (Birge & Louveaux 2011), may be used to handle parametric uncertainty suitably (Section 1.3).

Turnarounds take place once every few years, and their coordination in an integrated site requires the consideration of a long-term time scale. The frequency with which turnarounds on a particular plant must be performed is a complex issue. The determination of this may require the performance of simulations to characterize rates of wear and tear and the analysis of historical data, in an uncertainty quantification step. Bounds on the frequency of turnarounds can then be used as inputs to a turnaround planning optimization model (Section 1.3).

With this background, we describe the organization of the thesis.

1.2 OPTIMIZATION OVER STOCHASTIC SIMULATIONS: THEORY, ALGORITHMS, AND APPLICATIONS

The first part of the thesis addresses simulation optimization—an area still very much in its nascency. Simulation Optimization (SO) refers to the optimization of an objective function subject to constraints, both of which can only be evaluated through a stochastic simulation. To address specific features of a particular simulation—discrete or continuous decisions, expensive or cheap simulations, single or multiple outputs, homogeneous or heterogeneous noise—various algorithms have been proposed in the literature. As one can imagine, there exist several competing

algorithms for each of these classes of problems. We provide an extensive literature review of the area in Chapter 2.

In Chapter 3, we develop an algorithm for Continuous Optimization via Simulation (COvS) in an unconstrained setting. As the simulation may only be available as a black-box with stochastic outputs, we do not have access to derivative information, making the search for optimal parameters challenging. In addition, having a limited knowledge of the underlying distributions, and dealing simulations that are expensive to evaluate further complicate the problem. We extend prototypical trust region methods for derivative-free optimization (DFO) to the stochastic context, while providing guarantees of global convergence to stationary points.

We discuss our particular implementation of this strategy in Chapter 4. We provide computational experience with this framework, demonstrate that it is competitive with other algorithms on a large test set, and discuss two examples involving inventory optimization and DNA separations.

1.3 SEQUENTIAL AND SIMULTANEOUS STRATEGIES FOR OPTIMIZATION UNDER UNCERTAINTY: MODELS AND APPLICATIONS FOR SITE-WIDE MAINTENANCE

As mentioned earlier, individual plants in an integrated site interact closely, are dependent on each other for raw materials and demand for their products, and have the provision of intermediate storage tanks to help manage inventory at strategic points in the network. Disruptions in the operation of these plants can drastically affect flow of material in the site network. As a result, the choice of sequence and timing of planned periodic turnarounds, which are major disruptions, is important in order to minimize effects on profits and production.

In terms of time scales, a turnaround schedule over a multi-year horizon would be required along with a more detailed schedule for decisions over shorter time scales. An optimal long-term turnaround scheme is needed to plan for required manpower, to schedule turnarounds around seasonal constraints, and to maximize long-term profit margins while balancing financial performance. On the other hand, a fine-grained schedule over a shorter time scale would help manage short-term and medium-term production and inventory decisions and resolve resource (e.g.,

manpower, utilities) conflicts for a set of plants or units that are being maintained around the same time.

In the long-term context, we investigate a discrete-time mixed-integer linear programming (MILP) model to perform turnaround optimization. The objective is to recommend potential schedules in order to minimize losses while satisfying network, resource, turnaround, demand, financial and other practical constraints. We propose general formulations to tackle this problem and study an industrial-size site network under various scenarios over a long-term horizon in Chapter 5.

For the medium-term consideration of turnaround tasks, planning begins 6–9 months prior to the turnaround in order to plan downstream inventory build-up, and to plan for maintenance manpower and equipment. We consider a particular set of turnarounds that occur together, as recommended by a solution to the long-term turnaround planning problem. As mentioned earlier, the key uncertainty lies in the duration of the turnaround. Two aspects that are most affected by this are manpower availability and production planning. Manpower is a scarce resource, and it is important to ensure manpower availability at all times. Production planning is also affected by uncertainty in the duration of turnarounds due to the highly coupled nature of plants in an integrated site. In Chapter 6, we address the manpower issue through a robust optimization formulation, and production planning problem through the consideration of scenarios in a stochastic programming framework. The goal is to maximize site-wide profit margins by coordinating turnarounds and production through sequential and simultaneous strategies.

We summarize the contributions of the thesis in Section 7.1, where we discuss directions for future work in both simulation optimization, the key methodological contribution of the thesis, as well as in turnaround planning and coordination, which is the main application focus in our work.

Part II

OPTIMIZATION OVER STOCHASTIC SIMULATIONS: THEORY, ALGORITHMS, AND APPLICATIONS

Simulation optimization involves the optimization over stochastic simulations such as discrete-event simulations and stochastic differential equation systems. We provide a comprehensive review of current techniques in both discrete and continuous settings, and then develop a novel provably convergent trust region-based method for simulation optimization. We also demonstrate the practical use of the method through the description of an implementation, its success on a large test bed, and its application to two problems from chemical engineering, namely inventory optimization in chemical supply chains, and optimal sizing of obstructions for DNA separation.

2

LITERATURE REVIEW

2.1 INTRODUCTION

Advances in modeling and availability of cheap computational power have enabled the science, engineering, and business research communities to make use of simulations to model phenomena and systems. It is only natural that there be a great interest in manipulating degrees of freedom in the simulations to optimize them.

The term Simulation Optimization (SO) is an umbrella term for techniques used to optimize stochastic simulations. Simulation Optimization involves the search for those specific settings of the input parameters to a stochastic simulation such that a target objective, which is a function of the simulation output, is, without loss of generality, minimized.

As opposed to mathematical programming, SO does not assume that an algebraic description of the simulation is available—the simulation may be available as a black box that only allows the evaluation of the objective and constraints for a particular input. In fact, many SO algorithmic approaches solely depend on such input-output data from the simulation in their search for optimal input settings.

In addition, many large-scale and/or detailed simulations may be expensive to run, in terms of time, money, or resources. As a result, there is also a need to perform few simulations in this search for optimal parameters. Outputs from these stochastic simulations are not deterministic, and usually follow some output distribution, which may or may not vary across the parametric space. This uncertainty or variability in output also adds to the challenge of optimization, as it becomes harder to discern the quality of the parametric input in the presence of this output noise. In addition, when an algebraic description of the simulation is not accessi-

ble, derivative information is usually unavailable, and the estimation of derivatives from the use of finite differences may not be suitable due to noisy outputs and the expensive nature of simulations.

The nature of the stochastic simulations under study will determine the specific technique chosen to optimize them. The simulations, which are often discrete-event simulations, may be partially accessible to us in algebraic form, or may be purely available as an input-output model (as a black box); they may have single or multiple outputs; they may have deterministic or stochastic output(s); they may involve discrete or continuous parameters; and they may or may not involve explicit, or even implicit/hidden constraints.

A very general Simulation Optimization problem can be represented by (P1).

$$\begin{aligned} \min \quad & E_\omega[f(x,y,\omega)] \\ \text{s.t.} \quad & P(g(x,y,\omega) \leq 0) \geq 1-\alpha \\ & h(x,y) \leq 0 \\ & x_l \leq x \leq x_u \\ & x \in \mathbb{R}^n, y \in \mathbb{D}^m. \end{aligned} \tag{P1}$$

The function f can be evaluated through simulation for a particular instance of the continuous inputs x, discrete inputs y, and a realization of the random variables in the simulation, the vector ω (which may or may not be a function of the inputs, x and y). Similarly, the constraints defined by the vector-valued function g are also evaluated with each simulation run. In this formulation, expected values for these stochastic functions are used. There may be other constraints (represented by h) that do not involve random variables, as well as bound constraints on the decision variables.

The relaxation of any of these conditions would constitute a problem that would fall under the purview of SO. Most algorithms focus on problems that either have solely discrete choices, or solely continuous decisions to make. Each constraint may be thought of as representing additional outputs from the simulation that need to be taken into consideration. In addition, there may be bound constraints imposed on decision variables, that may either be available or obtained from domain-specific knowledge. Relatively few existing algorithms attempt to address both discrete and continuous choices simultaneously, although some broad classes of approaches

naturally lend themselves to be applicable in either, and therefore both, settings. Further, the discrete variables may either be binary, integer-ordered, or categorical and lie in some discrete space $\mathbb{D}$.

As can be seen, the formulation P1 is extremely general, and therefore a wide variety of applications fall under the scope of simulation optimization. Various applications of simulation optimization in diverse research fields are tabulated in Section 2.2.

Another common assumption is that f is a real-valued function and g is a real vector-valued function, both of whose expected values may or may not be smooth or continuous functions. The most common objective in SO is to optimize the expected value of some performance metric, but other objective functions may be appropriate depending on the application. For instance, an objective that minimizes risk could be a possible alternative, in which case one would incorporate some sort of variance measure as well into the objective.

This paper is meant to be a survey of available techniques as well as recent advances in simulation optimization. The remainder of the introduction section provides a literature survey of prior reviews, and elaborates on the relationship of simulation optimization to mathematical programming, derivative-free optimization, and machine learning. Section 2.2 provides a glimpse into the wide variety of applications of simulation optimization that have appeared in the literature. Section 2.3 focuses on various algorithms for discrete and continuous simulation optimization, provides basic pseudocode for major categories of algorithms, and provides comprehensive references for each type of algorithm. Section 2.4 provides a listing of available software for simulation optimization and Section 2.5 discusses means to compare their performance. Section 2.6 summarizes the progress of the field, and outlines some current and future topics for research.

2.1.1 *Prior reviews of simulation optimization*

Several review papers (e.g., Meketon (1987); Jacobson & Schruben (1989); Safizadeh (1990); Azadivar (1992); Fu (1994); Carson & Maria (1997); Andradóttir (1998); Azadivar (1999); Swisher et al. (2000); Fu et al. (2000); Fu (2002); Tekin & Sabuncuoglu (2004); Fu et al. (2005); Hong & Nelson (2009); Ammeri et al. (2011); Pasupathy & Ghosh (2013)), books and research monographs (e.g., Spall (2003b);

Rubinstein & Kroese (2004); Kleijnen (2008); Chen & Lee (2010)), and theses (e.g., Angün (2004); Driessen (2006); Deng (2007); Chang (2008); Frazier (2009); Kabirian (2009)) have traced the development of Simulation Optimization.

Meketon (1987) provides a classification of algorithmic approaches for optimization over simulations based on how much information or structure about the underlying model is known. The paper surveys the progress of the field between 1975 and 1987, and focuses on continuous simulation optimization. Andradóttir (1998) provides a tutorial on gradient-based procedures for continuous problems. Carson & Maria (1997) and Azadivar (1999) also give brief outlines of and pointers to prevailing simulation optimization algorithms.

Fu et al. (2000) contains several position statements of eminent researchers and practitioners in the field of simulation, where the integration of simulation with optimization is discussed. The issues addressed include generality vs. specificity of an algorithm, the wider scope of problems that simulation optimization methodologies have the potential to address, and the need for integrating provably convergent algorithms proposed by the research community with metaheuristics often used by commercial simulation software packages.

Of the more recent surveys, Fu (1994) provides an excellent tutorial on simulation optimization, and focuses on continuous optimization problems more than discrete optimization problems. The paper focuses specifically on discrete-event simulations. Fu (2002) provides a comprehensive survey of the field and its scope—the paper outlines the different ways in which optimization and simulation interact, gives examples of real-world applications, introduces simulation software and the optimization routines that each of them use, provides a very basic tutorial on simulation output analysis and convergence theory for simulation optimization, elaborates on algorithms for both continuous and discrete problems, and provides pointers to many useful sources. Fu et al. (2005) provide a concise, updated version of all of this, and also talk about estimation of distribution algorithms.

Tekin & Sabuncuoglu (2004) provide a table that analyzes past review papers and the techniques they focus on. Apart from providing detailed updates on advances in approaches and algorithms, the paper also lists references that attempt to compare different SO techniques. Hong & Nelson (2009) classify simulation optimization problems into those with (1) a finite number of solutions; (2) continuous decision variables; and (3) discrete variables that are integer-ordered. The paper

describes procedures for each of these classes. Perhaps the most recent survey by Ammeri et al. (2011) classifies simulation optimization algorithms and provides a survey of methods as well as applications appearing in the literature between 1995 and 2010.

This work provides an overview of techniques, and briefly outlines well-established methods with pointers to more detailed surveys, while expounding on more recent methods in a concise manner. Though several reviews exist, we catalog the most recent developments—the emergence of derivative-free optimization and its relationship with simulation optimization, the appearance of simulation test-beds for comparing algorithms, the recent application of simulation optimization in diverse fields, the development of and interest in related techniques and theory by the machine learning community and the optimization community, as well as the sheer unprecedented nature of recent interest in optimizing over simulations. A reflection of a surge in recent interest is evidenced by the fact that more than half of the works we reference were published in the last decade. The intent is to not only trace the progress of the field, but to provide an update on state-of-the-art methods and implementations, point the familiar as well as the uninitiated reader to relevant sources in the literature, and to speculate on future directions in the field.

2.1.2 *A note on terminology and scope*

As simulation optimization involves the use of algorithms that arose from widely differing fields (Section 2.3), has relationships to many diverse disciplines (Section 2.1.3), and has been applied to many different practical applications from biology to engineering to logistics (Section 2.2), it is not surprising that it is known by various names in different fields. It has also been referred to as simulation-based optimization, stochastic optimization, parametric optimization, black-box optimization, and Optimization via Simulation (OvS), where the continuous and discrete versions are accordingly known as Continuous Optimization via Simulation (COvS) and Discrete Optimization via Simulation (DOvS). Each algorithmic technique may also go by different names, and we attempt to reconcile these in Section 2.3.

Inputs to the simulation may be variously referred to as parameter settings, input settings, variables, controls, solutions, designs, experiments (or experimental designs), factors, or configurations. Outputs from the simulation are called mea-

surements, responses, performance metrics, objective values, simulation replications, realizations, or results. The performance of a simulation may also be referred to as an experiment, an objective function evaluation, or simply a function evaluation. We will use the term 'iteration' to refer to a fixed number of function evaluations (usually one) performed by a simulation optimization algorithm.

A note of caution while using SO methods is to incorporate as much domain specific knowledge as possible in the use of an SO algorithm. This may be in terms of (1) screening relevant input variables, (2) scaling and range reduction of decision variables, (3) providing good initial guesses for the algorithm; and (4) gleaning information from known problem structure, such as derivative estimates.

Table 2.1 classifies the techniques that are usually most suitable in practice for different scenarios in the universe of optimization problems. Certain broad classes of algorithms, such as random search methods, may be applicable to all of these types of problems, but they are often most suitable when dealing with pathological problems (e.g., problems with discontinuities, nonsmoothness) and are often used because they are relatively easy to implement.

Table 2.1: Terminology of optimization problems

	Algebraic model available	Unknown/complex problem structure
Deterministic	Traditional math programming (linear, integer, and nonlinear programming)	Derivative-free optimization
Uncertainty present	Stochastic programming, robust optimization	Simulation optimization

The possibilities of combining simulation and optimization procedures are vast: simulation with optimization-based iterations; optimization with simulation-based iterations; sequential simulation and optimization; and alternate simulation and optimization are four such paradigms. A recent paper by Figueira & Almada-Lobo (2014) delves into the taxonomy of such problems, and provides a guide to choosing an appropriate approach for a given problem. As detailed by Meketon (1987), different techniques may be applicable or more suitable depending on how much is known about the underlying simulation, such as its structure or associated probability dis-

tributions. We focus on approaches that are applicable in situations where all the optimization scheme has to work with are evaluations of $f(x, y, \omega)$ and $g(x, y, \omega)$, or simply, observations with noise.

2.1.3 *Relationship to other fields*

Mathematical Programming As mentioned earlier, most mathematical programming methods rely on the presence of an algebraic model. The availability of an algebraic model has many obvious implications to a mathematical programming expert, including the ability to evaluate a function quickly, the availability of derivative information, and the possibility of formulating a dual problem. None of these may be possible to do/obtain in an SO setting.

In the case with continuous decisions, derivative information is often hard to estimate accurately through finite differences, either due to the stochastic noise associated with objective function evaluations, or due to the large expense associated with obtaining function evaluations, or both. The inherent stochasticity in output also renders automatic differentiation (AD) (Rall 1981; Griewank & Walther 2008) tools not directly applicable. Moreover, automatic differentiation may not be used when one has no access to source code, does not possess an AD interface to proprietary simulation software, and, of course, when one is dealing with a physical experiment. The lack of availability of derivative information has further implications—it complicates the search for descent directions, proofs of convergence, and the characterization of optimal points.

Simulation Optimization, like stochastic programming, also attempts to optimize under uncertainty. However, stochastic programming differs in that it makes heavy use of the model structure itself (Birge & Louveaux 2011). Optimization under uncertainty techniques that make heavy use of mathematical programming are reviewed in Sahinidis (2004).

Derivative-Free Optimization Both Simulation Optimization and Derivative-Free Optimization (DFO) are referred to in the literature as black-box optimization methods. Output variability is the key factor that distinguishes SO from DFO, where the output from the simulation is deterministic. However, there are many approaches to

DFO that have analogs in SO as well (e.g., response surfaces, direct search methods, metaheuristics), cf. Section 2.3.

Another distinction is that most algorithms in DFO are specifically designed keeping in mind that function evaluations or simulations are expensive. This is not necessarily the case with SO algorithms.

With regard to rates of convergence, SO algorithms are generally inefficient and convergence rates are typically very slow. In general, one would expect SO to have a slower convergence rate than DFO algorithms simply because of the additional complication of uncertainty in function evaluations. As explained in Conn et al. (2009), some DFO algorithms, under certain assumptions, expect rates that are closer to linear than quadratic, and therefore early termination may be suitable. As described in some detail by Fu (1994), the best possible convergence rates for SO algorithms are generally $\mathcal{O}(1/\sqrt{k})$, where k is the number of samples. This is true from the central limit theorem that tells us the rate at which the best possible estimator converges to the true expected function value at a point. This implies that though one would ideally incorporate rigorous termination criteria in algorithm implementations, most practical applications have a fixed simulation or function evaluation budget that is reached first.

Machine Learning Several sub-communities in the machine learning community address problems closely related to simulation optimization. Traditional machine learning settings assume the availability of a fixed data set. Active learning methods (Cohn et al. 1996; Settles 2010) extend machine learning algorithms to the case where the algorithms are allowed to query an oracle for additional data to infer better statistical models. Active learning is closely related in that this choice of sampling occurs at every iteration in a simulation optimization setting as well. The focus of active learning is usually to learn better predictive models rather than to perform optimization.

Reinforcement learning (Stephens & Baritompa 1998) is broadly concerned with what set of actions to take in an environment to maximize some notion of cumulative reward. Reinforcement learning methods have strong connections to information theory, optimal control, and statistics. The similarity with simulation optimization is that the common problem of exploration of the search space vs. exploitation of

known structure of the cost function arises. However, in the reinforcement learning setting, each action usually also incurs a cost, and the task is to maximize the accumulated rewards from all actions—as opposed to finding a good point in the parameter space eventually.

Policy gradient methods (Peters et al. 2003) are a sub-field of reinforcement learning, where the set of all possible sequences of actions form the policy space, and a gradient in this policy space is estimated and a gradient ascent-type method is then used to move to a local optimum. Bandit optimization (Gittins 1989) is another sub-field of reinforcement learning that involves methods for the solution to the multi-armed bandit problem. The canonical example involves a certain number of slot machines, and a certain total budget to play them. Here, each choice of sample corresponds to which slot machine to play. Each play on a slot machine results in random winnings. This setting is analogous to discrete simulation optimization (DOvS) over finite sets, although with a different objective (Powell & Ryzhov 2012). Again, in DOvS over finite sets, we are only concerned with finding the best alternative eventually, whereas the cumulative winnings is the concern in the multi-armed bandit problem.

Relationship to other communities Most, if not all, simulation optimization procedures have elements that are derived from or highly related to several other fields. Direct search procedures and response surface methodologies (RSM) have strong relationships with the field of experimental design. RSM, sample path optimization procedures, and gradient-based methods heavily incorporate ideas from mathematical programming. RSM also involves the use of nonparametric and Bayesian regression techniques, whereas estimation of distribution algorithms involves probabilistic inference, and therefore these techniques are related to statistics and machine learning. Simulation Optimization has been described as being part of a larger field called computational stochastic optimization. More information is available at Powell (2013).

2.2 APPLICATIONS

SO techniques are most commonly applied to either (1) discrete-event simulations, or (2) systems of stochastic nonlinear and/or differential equations.

As mentioned in Fu (1994), discrete event simulations can be used to model many real-world systems such as queues, operations, and networks. Here, the simulation of a system usually involves switching or jumping from one state to another at discrete points in time as events occur. The occurrence of events is modeled using probability distributions to model the randomness involved.

Stochastic differential equations may be used to model phenomena ranging from financial risk (Merton 1974) to the control of nonlinear systems (Song & Grizzle 1995) to the electrophoretic separation of DNA molecules (Cho & Dorfman 2010).

With both discrete-event simulations and stochastic differential equation systems, there may be several parameters that one controls that affect some performance measure of the system under consideration, which are essentially degrees of freedom that may be optimized through SO techniques. Several applications of SO from diverse areas have been addressed in the literature and we list some of them in Table 2.2.

2.3 ALGORITHMS

Algorithms for SO are diverse, and their applicability may be highly dependent on the particular application. For instance, algorithms may (1) attempt to find local or global solutions; (2) address discrete or continuous variables; (3) incorporate random elements or not; (4) be tailored for cases where function evaluations are expensive; (5) emphasize exploration or exploitation to different extents; (6) assume that the uncertainty in simulation output is homoscedastic or that it comes from a certain probability distribution; or (7) rely on underlying continuity or differentiability of the expectation (or some function of a chosen moment) of the simulation output. The sheer diversity of these algorithms also makes it somewhat difficult to assert which one is better than another in general, and also makes it hard to compare between algorithms or their implementations.

As mentioned in Section 2.1.3, many algorithms that are available for continuous simulation optimization have analogs in derivative-based optimization and in derivative-free optimization, where function evaluations are deterministic. In any case, the key lies in the statistics of how noise is handled, and how it is integrated into the optimization scheme. We will provide pointers to references that are appli-

Table 2.2: Partial list of published works that apply simulation optimization

Domain of application	Application and citations
Operations	Buffer location (Lutz et al. 1998), nurse scheduling (Tein & Ramli 2010), inventory management (Köchel & Nieländer 2005; Schwartz et al. 2006), health care (de Angelis et al. 2003), queuing networks (Fu & Hill 1997; Bhatnagar 2005; Mishra et al. 2007)
Manufacturing	PCB production (Dengiz & Akbay 2000), engine manufacturing (Syberfeldt & Lidberg 2012), production planning (Kenne & Gharbi 2001; Kleijnen 1993), manufacturing-cell design (Irizarry et al. 2001), kanban sizing (Hall et al. 1996)
Medicine and biology	Protein engineering (Romero et al. 2013), cardiovascular surgery (Xie et al. 2012), breast cancer epidemiology (Ferris et al. 2005), bioprocess control (Vande Wouwer et al. 2001; Renotte & Vande Wouwer 2003), ECG analysis (Gerencsér et al. 2002), medical image analysis (Merhof et al. 2007)
Engineering	Welded beam design (Yang & Deb 2010), solid waste management (Yeomans 2007), pollution source identification (Ayvaz 2010), chemical supply chains (Jung et al. 2004), antenna design (Prakash et al. 2008), aerodynamic design (Xing & Damodaran 2002; 2005b;a; Kothandaraman & Rotea 2005), distillation column optimization (Ramanathan et al. 2001), well placement (Bangerth et al. 2005), servo system control (Radac et al. 2011), power systems (Ernst et al. 2007), radar analysis (Khan et al. 2006)
Computer science, networks, electronics	Server assignment (Kulturel-Konak & Konak 2010), wireless sensor networks (Dhivya et al. 2011), circuit design (Li 2009), network reliability (Kroese et al. 2007)
Transportation and logistics	Traffic control and simulation (Yun & Park 2010; Balakrishna et al. 2007; Osorio & Bierlaire 2010), metro/-transit travel times (Hill & Fu 1995; Ö. Yalçinkaya 2009), air traffic control (Kleinman et al. 1997; Hutchison & Hill 2001)

cable to simulation optimization in particular. A comprehensive review of methods for derivative-free optimization is available in Rios & Sahinidis (2013).

Each major subsection below is accompanied by pseudocode to give researchers and practitioners unfamiliar with the field an idea of the general approach taken by each of these algorithms. Many of the sections include pointers to convergence proofs for individual algorithms. Optimality in simulation optimization is harder to establish than in mathematical programming or derivative-free optimization due to the presence of output variability. Notions of optimality for simulation optimization are explored in Fu (1994); for the discrete case, Xu et al. (2010), for instance, establishes conditions for local convergence, where a point being 'better' than its $2m + 1$ neighboring solutions is said to be locally optimal. There has also been some work in establishing Karush-Kuhn-Tucker (KKT) optimality conditions for multiresponse simulation optimization (Bettonvil et al. 2009). Globally convergent algorithms will locate the global optimal solution eventually, but assuring this would require all feasible solutions to be evaluated through infinite observations; in practice, a convergence property that translates to a practical stopping criterion may make more sense (Hong & Nelson 2009).

Based on their scope, the broad classes of algorithms are classified in Table 2.3. Algorithms are classified based on whether they are applicable to problems with discrete/continuous variables, and whether they focus on global or local optimization. However, there may be specific algorithms that have been tweaked to make them applicable to a different class as well, which may not be captured by this table.

Table 2.3: Classification of simulation optimization algorithms

Algorithm class	Discrete	Continuous	Local	Global
Ranking and Selection	×			×
Metaheuristics	×	×		×
Response Surface Methodology		×	×	×
Gradient-based methods		×	×	
Direct search	×	×	×	
Model-based methods	×	×	×	×
Lipschitzian optimization		×		×

2.3.1 *Discrete optimization via simulation*

Discrete optimization via simulation is involved with finding optimal settings for input variables that can only take discrete values. This may be in the form of *integer-ordered* variables or *categorical* variables (Pasupathy & Henderson 2011). Integer-ordered variables are allowed to take on integer or discrete values within a finite interval, where the order of these values translates to some physical interpretation. For example, this could be the number of trucks available for vehicle routing, or the set of standard pipe diameters that are available for the construction of a manufacturing plant. Categorical variables refer to more general kinds of discrete decisions, ranging from conventional on-off (0-1 or binary) variables to more abstract decisions such as the sequence of actions to take given a finite set of actions. It should be noted that though integer-ordered variables, for instance, may be logically represented using binary variables, it may be beneficial to retain them as integer-ordered to exploit correlations in objective function values between adjacent integer values.

A rich literature in DOvS has developed over the last 50 years, and the specific methods developed are tailored to the specific problem setting. Broadly, methods are tailored for finite or for very large/potentially infinite parameter spaces.

2.3.1.1 *Finite parameter spaces*

In the finite case, where the number of alternatives is small and fixed, the primary goal is to decide how to allocate the simulation runs among the alternatives. In this setting, there is no emphasis on 'search', as the candidate solution pool is small and known; each iteration is used to infer the best, in some statistical sense, simulation run(s) to be performed subsequently.

The optimization that is desired may differ depending on the situation, and could involve:

1. The selection of the best candidate solution from a finite set of alternatives;

2. The comparison of simulation performance measures of each alternative to a known standard or control; or

3. The pairwise comparison between all solution candidates.

Item (1) is referred to as the *ranking and selection* problem. Items (2) and (3) are addressed under literature on *multiple comparison procedures*, with the former referred to as *multiple comparisons with a control*.

Ranking and Selection In traditional ranking and selection, the task is to minimize the number of simulation replications while ensuring a certain probability of correct selection of alternatives. Most procedures try to guarantee that the design ultimately selected is better than all competing alternatives by δ with a probability at least $1 - \alpha$. δ is called the indifference zone, and is the value deemed to be sufficient to distinguish between expected performance among solution candidates.

Conventional procedures make use of the Bonferroni inequality which relates probabilities of the occurrence of multiple events with probabilities of each event. Other approaches involve the incorporation of covariance induced by, for example, the use of common random numbers to expedite the algorithmic performance over the more conservative Bonferroni approach. Kim & Nelson (2006; 2007) provide a detailed review and provide algorithms and procedures for this setting. Extensions of fully sequential ranking and selection procedures to the constrained case have been explored as well, e.g., Andradóttir & Kim (2010).

An alternative formulation of the ranking and selection of the problem would be to try to do the best within a specified computational budget, called the *optimal computing budget allocation* formulation (Chen 1995). Chen et al. (2009) present more recent work, while the stochastically constrained case is considered in Lee et al. (2012).

Recent work (Hunter & Pasupathy 2013) in the area of DOvS over finite sets provides a quick overview of the field of ranking and selection, and considers general probability distributions and the presence of stochastic constraints simultaneously.

A basic ranking and selection procedure (Kim & Nelson 2007) is outlined in Algorithm 2.1, where it is assumed that independent data comes from normal distributions with unknown, different variances.

Multiple comparison procedures Here, a number of simulation replications are performed on all the potential designs, and conclusions are made by constructing confidence intervals on the performance metric. The main ideas and techniques for multiple comparisons in the context of pairwise comparisons, or against a known

Algorithm 2.1 Basic ranking and selection procedure for SO

Require: Confidence level $1-\alpha$, indifference zone parameter δ

1: Take n_0 samples from each of the $1, \ldots, K$ potential designs
2: Compute sample means, $\bar{t}_{k,n_0}$ and sample variances, S_k, for each of the designs
3: Determine how many new samples, $N_k := \max\left\{n_0, \left\lceil \frac{\psi^2 S_k^2}{\delta^2} \right\rceil\right\}$, to take from each system, where the Rinott constant ψ is obtained from Bechhofer et al. (1995)
4: Select the system with the best new sample mean, $\bar{t}_{k,N_k+n_0}$.

standard are presented in Hochberg & Tamhane (1987), Fu (1994), and Hsu (1996). Recent work in multiple comparisons with a control include Kim (2005) and Nelson & Goldsman (2001), which provide fully sequential and two-stage frequentist procedures respectively; and Xie & Frazier (2013), which addresses the problem using a Bayesian approach.

Comprehensive treatment of ranking and selection and multiple comparison procedures may be found in Goldsman & Nelson (1998) and Bechhofer et al. (1995). A detailed survey that traces the development of techniques in simulation optimization over finite sets is available in Tekin & Sabuncuoglu (2004).

2.3.1.2 *Large/Infinite parameter spaces*

To address DOvS problems with a large number of potential alternatives, algorithms that have a search component are required. Many of the algorithms that are applicable to the continuous optimization via simulation case are, with suitable modifications, applicable to the case with large/infinite parameter spaces. These include (1) ordinal optimization (2) random search methods and (3) direct search methods.

Ordinal optimization methods (Ho 1999) are suitable when the number of alternatives is too large to find the globally optimal design in the discrete-event simulation context. Instead, the task is to find a satisfactory solution with some guarantees on quality (called alignment probability) (Lau & Ho 1997). Here, the focus is on sampling a chosen subset of the solutions and evaluating them to determine the best among them. The key lies in choosing this subset such that it contains a subset of satisfactory solutions. The quality or satisfaction level of this selected subset can

be quantified (Chen 1996). A comparison of subset selection rules is presented in Jia et al. (2006) and the multi-objective case is treated in Teng et al. (2007).

Random search methods, include techniques such as simulated annealing (e.g., Alrefaei & Andradóttir (1999)), genetic algorithms, stochastic ruler methods (e.g., Yan & Mukai (1992)), stochastic comparison (e.g., Gong et al. (49)), nested partitions (e.g., Shi & Ólafsson (2000)), ant colony optimization (e.g., Dorigo & Stützle (2004); Dorigo & Blum (2005)), and tabu search (e.g., Glover & Hanafi (2002)). Some of these—simulated annealing, genetic algorithms, and tabu search—are described in Section 2.3.6). Ant colony optimization is described under model-based methods (cf. Section 2.3.7.2). Proofs of global convergence, i.e., convergence to the global solution, or local convergence are available for most of these algorithms (Hong & Nelson 2009) (note that these definitions differ from mathematical programming where *global convergence* properties ensure convergence to a *local optimum* regardless of the starting point).

Nested partition methods (Shi & Ólafsson 2007) attempt to adaptively sample from the feasible region. The feasible region is then partitioned, and sampling is concentrated in regions adjudged to be the most promising by the algorithm from a pre-determined collection of nested sets. Hong and Nelson propose the COMPASS algorithm (Hong & Nelson 2006) which uses a unique neighborhood structure, defined as the most promising region that is fully adaptive rather than pre-determined; a most promising 'index' is defined that classifies each candidate solution based on a nearest neighbor metric. More recently, the Adaptive Hyberbox Algorithm (Xu et al. 2013) claims to have superior performance on high-dimensional problems (problems with more than ten or fifteen variables); and the R-SPLINE algorithm (Wang et al. 2012), which alternates between a continuous search on a continuous piecewise-linear interpolation and a discrete neighborhood search, compares favorably as well.

A review of random search methods is presented in Andradóttir (2006); Ólafsson (2006). Recent progress, outlines of basic algorithms, and pointers to specific references for some of these methods are presented in Bianchi et al. (2009), Hong & Nelson (2009), and Nelson (2010).

Direct search methods such as pattern search and Nelder-Mead simplex methods are elaborated on in Section 2.3.5.

2.3.2 *Response surface methodology*

Response surface methodology (RSM) is typically useful in the context of continuous optimization problems and focuses on learning input-output relationships to approximate the underlying simulation by a surface (also known as a metamodel or surrogate model) for which we define a functional form. This functional form can then be made use of by leveraging powerful derivative-based optimization techniques. The literature in RSM is vast and equivalent approaches have variously been referred to as multi-disciplinary design optimization, metamodel-based optimization, and sequential parameter optimization. RSM was originally developed in the context of experimental design for physical processes (Box & Wilson 1951), but has since been applied to computer experiments. Metamodel-based optimization is a currently popular technique for addressing simulation optimization problems (Barton & Meckesheimer 2006; Kleijnen 2008).

Algorithm 2.2 Basic RSM procedure

Require: Initial region of approximation $\mathcal{X}$, choice of regression surface r

1: **while** not converged or under simulation budget **do**
2: Perform a design of experiments in relevant region, using k data points
3: $t_i \leftarrow \text{simulate}(x_i), \quad i = \{1, \ldots, k\}$ {Evaluate noisy function $f(x_i, \omega)$}
4: $\lambda^* \leftarrow \arg\min_\lambda \sum (t_i - r(x_i, \lambda))^2$ {Fit regression surface r through points using squared loss function}
5: $x^* \leftarrow \{\arg\min_x r(x, \lambda^*) : x \in \mathcal{X}\}$ {Optimize surface}
6: Update set of available data points and region of approximation
7: **end while**

Different response surface algorithms differ in the choice between regression and interpolation; the nature of the functional form used for approximation (polynomials, splines, Kriging, radial basis functions, neural networks); the choice of how many and where new samples must be taken; and how they update the response surface.

RSM approaches can either (1) build surrogate models that are effective in local regions, and sequentially use these models to guide the search, or; (2) build surrogate models for the entire parameter space from space-filling designs, and then use them to choose samples in areas of interest, i.e., where the likelihood of finding better solutions is good according to a specified metric. A generic framework for RSM is presented in Algorithm 2.2.

Classical sequential RSM Originally, RSM consisted of a Phase I, where first order models were built using samples from a design of experiments. A steepest descent rule was used to move in a certain direction, and this would continue iteratively until the estimated gradient would be close to zero. Then, a Phase II procedure that built a more detailed quadratic model would be used for verifying the optimality of the experimental design. A thorough introduction to response surface methodology is available in Myers et al. (2009). Recent work in the field includes automating RSM (Neddermeijer et al. 2000; Nicolai & Dekker 2009) and the capability to handle stochastic constraints (Angün et al. 2009).

Bayesian global optimization These methods seek to build a global response surface, commonly using techniques such as Kriging/Gaussian process regression (Sacks et al. 1989; Rasmussen & Williams 2006). Subsequent samples chosen based on some sort of improvement metric may balance exploitation and exploration. The seminal paper by Jones et al. (1998) which introduced the EGO algorithm for simulations with deterministic output, uses Kriging to interpolate between function values, and chooses future samples based on an expected improvement metric (Mockus et al. 1978). Examples of analogs to this for simulation optimization are provided in Huang et al. (2006); Kleijnen et al. (2012). The use of Kriging for simulation metamodeling is explored in van Beers & Kleijnen (2004); Kleijnen & van Beers (2005); Kleijnen (2009). Other criteria that have been used to choose samples are most probable improvement (Mockus 1989), knowledge gradient for continuous parameters (Scott et al. 2011), and maximum information gain (Srinivas et al. 2012).

Trust region methods Trust region methods (Conn et al. 2000) can be used to implement sequential RSM. Trust regions provide a means of controlling the region of approximation, providing update criteria for surrogate models, and are useful in analyzing convergence properties. Once a metamodel or response surface, g, is built around a trust region center x_i, trust region algorithms involve the solution of the trust-region subproblem $(\min_s g(x_i + s) : s \in \mathcal{B}(x_i, \Delta))$, where $\mathcal{B}$ is a ball defined by the center-radius pair (x_i, Δ). There are well-defined criteria to update the trust region center and radius (Conn et al. 2000) that will define the subsequent region of approximation.

The use of trust regions in simulation optimization is relatively recent, and has been investigated to some extent (Deng & Ferris 2006; Chang et al. 2013). Trust-region algorithms have been used, for example, to optimize simulations of urban traffic networks (Osorio & Bierlaire 2010).

2.3.3 *Gradient-based methods*

Stochastic approximation methods or gradient-based approaches are those that attempt to descend using estimated gradient information. Stochastic approximation techniques are one of the oldest methods for simulation optimization. Robbins & Monro (1951) and Kiefer & Wolfowitz (1952) were the first to develop stochastic approximation schemes in the early 1950s. These procedures initially were meant to be used under very restrictive conditions, but much progress has been made since then.

These methods can be thought of being analogous to steepest descent methods in derivative-based optimization. One may obtain direct gradients or may estimate gradients using some finite difference scheme. Direct gradients may be calculated by a number of methods: (1) Perturbation Analysis (specifically, Infinitesimal Perturbation Analysis) (PA or IPA), (2) Likelihood Ratio/Score Function (LR/SF), and (3) Frequency Domain Analysis (FDA). Detailed books on these methods are available in the literature (Ho & Cao 1991; Glasserman 1991; Rubinstein & Shapiro 1993; Pflug 1996; Fu & Hu 1997) and more high-level descriptions are available in papers (Tekin & Sabuncuoglu 2004; Fu 2002). Most of these direct methods, however, are either applicable to specific kinds of problems, need some information about underlying distributions, or are difficult to apply. Fu (2002) outlines which methods are applicable in which situations, and Tekin & Sabuncuoglu (2004) discuss a number of applications that have used these methods.

Stochastic approximation schemes attempt to estimate a gradient by means of finite differences. Typically, a forward difference estimate would involve sampling at least $n + 1$ distinct points, but superior performance has been observed by simultaneous perturbation estimates that require samples at just two points (Spall 2003a), a method referred to as Simultaneous Perturbation Stochastic Approximation (SPSA). The advantage gained in SPSA is that the samples required are now

independent of the problem size, and, interestingly, this has been shown to have the same asymptotic convergence rate as the naive method that requires $n+1$ points (Spall 1992). A typical gradient-based scheme is outlined in Algorithm 2.3.

Algorithm 2.3 Basic gradient-based procedure

Require: Specify initial point, x_0. Define initial parameters such as step size (α), distances between points for performing finite difference, etc.

1: $i \leftarrow 0$
2: **while** not converged or under simulation budget **do**
3: Perform required simulations, $t_i^{j_i} \leftarrow$ simulate(x_i), with j_i replications to estimate gradient, $\widehat{J}$, using either IPA, LR/SF, FDA or finite differences
4: $x_{i+1} \leftarrow x_i - \alpha\widehat{J}$
5: $i \leftarrow i+1$
6: **end while**

Recent extensions of the SPSA method include introducing a global search component to the algorithm by injecting Monte Carlo noise during the update step (Maryak & Chin 2008), and using it to solve combined discrete/continuous optimization problems (Wang & Spall 2011). Recent work also addresses improving Jacobian as well as Hessian estimates in the context of the SPSA algorithm (Spall 2009). A recent review of stochastic approximation methods is available in Spall (2012). Much of the progress in stochastic approximation has been cataloged in the proceedings of the Winter Simulation Conference over the years (`http://informs-sim.org/`). A recent review of stochastic approximation methods is available in Spall (2012), and an excellent tutorial and review of results in stochastic approximation is presented in Pasupathy & Kim (2011).

2.3.4 *Sample path optimization*

Sample path optimization involves working with an estimate of the underlying unknown function, as opposed to the function itself. The estimate is usually a consistent estimator such as the sample mean of independent function evaluations at a point, or replications. For instance, one may work with $F_n = \frac{1}{n}\sum_{i=1}^{n} f(x, y, \omega_i)$, instead of the underlying function $E[f(x, y, \omega)]$ itself. It should be noted that the functional form of F_n is still unknown, it is just that F_n can be observed or evaluated at a point in the search space visited by an algorithm iteration. The alternative name of sample average approximation reflects this use of an estimator.

As the algorithm now has to work with an estimator, a deterministic realization of the underlying stochastic function, sophisticated techniques from traditional mathematical programming can now be leveraged. Sample path methods can be viewed as the use of deterministic optimization techniques within a well-defined stochastic setting. Yet another name for them is stochastic counterpart. Some of the first papers using sample path methods are Healy & Schruben (1991) and Shapiro (1991). Several papers (Rubinstein & Shapiro 1993; Chen & Schmeiser 1994; Gürkan et al. 1994; Shapiro 1996) discuss convergence results and algorithms in this context.

2.3.5 *Direct search methods*

Direct search can be defined as the sequential examination of trial solutions generated by a certain strategy (Hooke & Jeeves 1961). As opposed to stochastic approximation, direct search methods rely on direct comparison of function values without attempting to approximate derivatives. Direct search methods typically rely on some sort of ranking of quality of points, rather than on function values.

Most direct search algorithms developed for simulation optimization are extensions of ideas for derivative-free optimization. A comprehensive review of classical and modern methods is provided in Kolda et al. (2003). A formal theory of direct search methods for stochastic optimization is developed in Trosset (2000). Direct search methods can be tailored for both discrete and continuous optimization settings. Pattern search and Nelder-Mead simplex procedures are the most popular direct search methods. There is some classical as well as relatively recent work done on investigating both pattern search methods (Trosset 2000; Anderson & Ferris 2001; Lucidi & Sciandrone 2002) and Nelder-Mead simplex algorithms (Nelder & Mead 1965; Barton & Ivey, Jr. 1996; Humphrey & Wilson 2000; Chang 2012) and their convergence in the context of simulation optimization.

These methods remain attractive as they are relatively easy to describe and implement, and are not affected if a gradient does not exist everywhere, as they do not rely on gradient information. Since conventional procedures can be affected by noise, effective sampling schemes to control the noise are required. A basic Nelder-Mead procedure is outlined in Algorithm 2.4.

Algorithm 2.4 Basic Nelder-Mead simplex procedure for SO

Require: A set of $n-1$ points in the parameter space to form the initial simplex

1: **while** not satisfied prespecified convergence criterion or under simulation budget **do**

2: Generate a new candidate solution, x_i, through simplex centroid reflections, contractions or other means

3: $t_i^{j_i} \leftarrow \text{simulate}(x_i), \quad i = \{i-n+1, \ldots, i\}, j_i = \{1, \ldots, N_i\}$ {Evaluate noisy function $f(x, \omega)$ N_i times, where N_i is determined by some sampling scheme}

4: Calculate $\frac{\sum_{j_i} t_i^{j_i}}{N_i}$, or some similar metric to determine which point (i.e., with the highest metric value) should be eliminated

5: **end while**

2.3.6 *Random search methods*

2.3.6.1 *Genetic algorithms*

Genetic algorithms use concepts of mutation and selection from theory of evolution (Reeves 1997; Whitley 1994). In general, The genetic algorithm works by creating a population of strings and each of these strings are called chromosomes. Each of these chromosome strings is basically a vector of point in the search space. New chromosomes are created by using selection, mutation and crossover functions. The selection process is guided by evaluating the fitness (or objective function) of each chromosome and selecting the chromosomes according to their fitness values (using methods such as mapping onto Roulette Wheel). Additional chromosomes are then generated using crossover and mutation functions. The cross over and mutation functions ensures that a diversity of solutions is maintained. Genetic algorithms are popular as they are easy to implement and are used in several commercial simulation optimization software packages (Table 2.4). The GECCO (Genetic and Evolutionary Computation Conference) catalogs progress in genetic algorithms and implementations.

2.3.6.2 *Simulated annealing*

Simulated Annealing uses a probabilistic method that is derived from the annealing process in which the material is slowly cooled so that its structure is frozen and it reaches a minimum energy state (Kirkpatrick et al. 1983; Bertsimas & Tsitsiklis

1993). Starting with a current point i in a state j, a neighborhood point i' of the point i is generated. The algorithm moves from point i to i' using a probabilistic criteria that is dependent on the 'temperature' in state j. This temperature is analogous to that in physical annealing, and serves here as a control parameter. If the solution at i' is better than the existing solution, then this new point is accepted. If the new solution is worse than existing solution, then the probability of accepting the point is defined as $\exp(-(f(i') - f(i))/T(j))$, where $f(.)$ is the value of objective function at a given point, and $T(j)$ is temperature at the state j. After a certain number of neighborhood points are evaluated, the temperature is decreased and new state is $j + 1$ is created. Due to the exponential form, the probability of acceptance of a neighborhood point is higher at high temperature, and is lower as temperature is reduced. In this way, the algorithm searches for a large number of neighborhood points in the beginning, but a lower number of points as temperature is reduced.

Implementation of simulated annealing procedures require choosing parameters such as the initial and final temperatures, the rate of cooling, and number of function evaluations at each temperature. A variety of cooling 'schedules' have been suggested in Collins et al. (1988) and Hajek (1988). Though simulated annealing was originally meant for optimizing deterministic functions, the framework has been extended to the case of stochastic simulations (Alkhamis et al. 1999). The ease of implementing a simulated annealing procedure is high and it remains a popular technique used by several commercial simulation optimization packages.

2.3.6.3 *Tabu search*

Tabu search (Glover 1990) uses special memory structures (short-term and long-term) during the search process that allow the method to go beyond local optimality to explore promising regions of the search space. The basic form of tabu search consists of a modified neighborhood search procedure that employs adaptive memory to keep track of relevant solution history, together with strategies for exploiting this memory (Gendreau & Potvin 2010). More advanced forms of tabu search and its applications are described in Glover & Laguna (1997).

2.3.6.4 *Scatter search*

Scatter search and its generalized form, path relinking, were originally introduced by Glover & Laguna (2000). Scatter search differs from other evolutionary approaches

(such as Genetic Algorithms (GA)) by using strategic designs and search path construction from a population of solutions as compared to randomization (by crossover and mutation in GA). Similar to Tabu search, Scatter Search also utilize adaptive memory in storing best solutions (Glover & Laguna 2000; Martí et al. 2006). Algorithm 2.5 provides the scatter search algorithm.

Algorithm 2.5 Basic scatter search procedure for SO

Require: An initial set of trial points $x \in P$, chosen to be diversified according to a pre-specified metric

1: $t_j \leftarrow$ simulate(x_j), where $j = 1, \ldots, |P|$
2: $k \leftarrow 0$
3: Use a comparison procedure (such as ranking and selection) to gather the best b solutions (based on objective value or diversity) from the current set of solutions P, called the reference set, R_k
4: $R_{-1} = \varnothing$
5: **while** under simulation budget and $R_k \neq R_{k-1}$ **do**
6: $k \leftarrow k + 1$
7: Choose $S_i \subset R$, where $i = 1, \ldots, r$ {Use a subset generation procedure to select r subsets of set R, to be used as a basis for generating new solution points}
8: **for** $i = 1$ to r **do**
9: Combine the points in S_i, to form new solution points, x_j, where $j \in \mathcal{J} = |P| + 1, \ldots, |P| + J$, using weighted linear combinations, for example
10: $t_j \leftarrow$ simulate(x_j), $j \in \mathcal{J}$ {sample the objective function at new trial solutions}
11: Update sets R_k, P
12: **end for**
13: **end while**

2.3.7 *Model-based methods*

Model-based simulation optimization methods attempt to build a probability distribution over the space of solutions and use it to guide the search process.

2.3.7.1 *Estimation of distribution algorithms*

Estimation of distribution algorithms (EDAs) (Larrañaga & Lozano 2002) are model-based methods that belong to the evolutionary computation field. However, generation of new candidate solutions is done by sampling from the inferred probability distribution over the space of solutions, rather than, say, a genetic operator such as

crossover or mutation. A comprehensive review of estimation of distribution algorithms is presented in Fu et al. (1996). EDAs usually consider interactions between the problem variables and exploit them through different probability models.

Cross-entropy methods and Model Reference Adaptive Search (MRAS) are discussed next and can be seen as specific instances of EDAs.

Cross-Entropy Methods Cross-entropy methods first sample randomly from a chosen probability distribution over the space of decision variables. For each sample, which is a vector defining a point in decision space, a corresponding function evaluation is obtained. Based on the function values observed, a pre-defined percentile of the best samples are picked. A new distribution is built around this 'elite set' of points via maximum likelihood estimation or some other fitting method, and the process is repeated. One possible method that implements cross-entropy is formally described in Algorithm 2.6.

Algorithm 2.6 Pseudocode for a simple cross-entropy implementation

Require: θ, an initial set of parameters for a pre-chosen distribution $p(x;\theta)$ over the set of decision variables; k, a number of simulations to be performed; e, the number of elite samples representing the top δ percentile of the k samples

```
while not converged or under simulation budget do
    for i = 1 → k do
        sample x_i from p(x; θ)
        t_i ← simulate(x_i)
    end for
    E ← ∅
    for i = 1 → e do
        E_j ← arg max_{i∉E} t_i
    end for
    p(x; θ) ← fit(x_E)
end while
```

The method is guaranteed (probabilistically) to converge to a local optimum, but it also incorporates an exploration step as random samples are obtained at each step. However, the intuition behind the selection of subsequent samples can be shown to be analogous to minimizing the Kullback-Leibler divergence (KL-divergence) between the optimal importance sampling distribution and the distribution used in the current iterate (Rubinstein & Kroese 2004).

There exist variants of the cross-entropy method to address both continuous (Kroese et al. 2006) and discrete optimization (Rubinstein 1999) problems. A possible modification is to use mixtures of distributions from current and previous iterations, with the current distribution weighted higher. This can be done by linearly interpolating the mean covariance in the case of Gaussian distributions. This also helps in avoiding singular covariance matrices. Cross-entropy can also deal with noisy function evaluations, with irrelevant decision variables, and constraints (Kroese et al. 2006). If decision variables are correlated, the covariance of the distribution will reflect this.

The immediately apparent merits of cross-entropy methods are that they are easy to implement, require few algorithmic parameters, are based on fundamental principles such as KL-divergence and maximum likelihood, and give consistently accurate results (Kroese et al. 2006). A potential drawback is that cross-entropy may require a significant number of new samples at every iteration. It is not clear as to how this would affect performance if samples were expensive to obtain. The cross-entropy method has analogs in simulated annealing, genetic algorithms, and ant colony optimization, but differs from each of these in important ways (de Boer et al. 2005).

More detailed information on the use of cross-entropy methods for optimization can be found in de Boer et al. (2005), a tutorial on cross-entropy and in Rubinstein & Kroese (2004), a monograph. The cross-entropy webpage provides up-to-date information on progress in the field (http://iew3.technion.ac.il/CE/).

Model reference adaptive search (MRAS) The MRAS method (Hu et al. 2005; 2007) is closely related to the cross-entropy method. It also works by minimizing the Kullback-Leibler divergence to update the parameters of the inferred probability distribution. However, the parameter update step involves the use of a sequence of implicit probability distributions. In other words, while the cross-entropy method uses the optimal importance sampling distribution for parameter updates, MRAS minimizes the KL-divergence with respect to the distribution in the current iteration, called the reference model.

Covariance Matrix Adaptation–Evolution Strategy (CMA-ES) In the CMA-ES algorithm (Hansen 2006), new samples are sampled from a multivariate normal

distribution, and inter-variable dependencies are encoded in the covariance matrix. The CMA-ES method provides a way to update the covariance matrix. Updating the covariance matrix is analogous to learning an approximate inverse Hessian, as is used in Quasi-Newton methods in mathematical programming. The update of the mean and covariance is done by maximizing the likelihood of previously successful candidate solutions and search steps, respectively. This is in contrast to other EDAs and the cross-entropy method, where the covariance is updated by maximizing the likelihood of the successful points. Other sophistications such as step-size control, and weighting of candidate solutions are part of modern implementations (Hansen 2011).

2.3.7.2 *Ant colony optimization*

Ant colony optimization methods (Dorigo & Stützle 2004; Dorigo & Blum 2005) are heuristic methods that have been used for combinatorial optimization problems. Conceptually, they mimic the behavior of ants to find shortest paths between their colony and food sources. Ants deposit pheromones as they walk; and are more likely to choose paths with higher concentration of pheromones. This phenomenon is incorporated in a pheromone update rule, which increases the pheromone content in components of high-quality solutions, and causes evaporation of pheromones in less favorable regions. Probability distributions are used to make the transition between each iteration. These methods differ from EDAs in that they use an iterative construction of solutions.

This and other algorithms that incorporate self-organization in biological systems are said to use the concept of 'swarm intelligence'.

2.3.8 *Lipschitzian optimization*

Lipschitzian optimization is a class of space-partitioning algorithms for performing global optimization, where the Lipschitz constant is pre-specified. This enables the construction of global search algorithms with convergence guarantees. The caveat of having prior knowledge of the Lipschitz constant is overcome by the DIRECT (DIviding RECTangles) algorithm (Jones et al. 1993) for deterministic continuous

optimization problems. An adaptation of this for noisy problems is provided in Deng & Ferris (2007).

2.4 SOFTWARE

2.4.1 *Simulation optimization in commercial simulation software*

Many discrete-event simulation packages incorporate some methodology for performing optimization. A comprehensive listing of simulation software, the corresponding vendors, and the optimization packages and techniques they use can be found in Table 2.4. More details on the specific optimization routines can be found in Law & Kelton (2000). OR/MS-Today, the online magazine of INFORMS, conducts a biennial survey of simulation software packages, the latest of which is available at OR/MS today 2013 survey (2013). The survey lists 43 simulation software packages, and 31 of these have some sort of optimization routine; fewer still have black-box optimizers that interact with the simulation.

2.4.2 *Academic implementations of simulation optimization*

Table 4.1 contains a small subset of academic implementations of SO algorithms, and classifies them by type. Some of these are available for download from the web, some have code with suggested parameters in corresponding papers themselves, and others are available upon request from the authors.

2.5 COMPARISON OF ALGORITHMS

As far as comparisons between algorithms are concerned, the literature does not yet provide a comprehensive survey of the performance of different implementations and approaches on large test beds. In this regard, simulation optimization lags behind other optimization fields such as linear, integer, and nonlinear programming, global optimization and even derivative-free optimization, where the first comprehensive comparison appeared in 2013 (Rios & Sahinidis 2013). A study of prior comparisons in simulation optimization is provided by (Tekin & Sabuncuoglu 2004),

Table 2.4: Simulation optimization packages in commercial simulation software

Optimization package	**Vendor**	**Simulation software supported**	**Optimization methodology**
AutoStat	Applied Materials, Inc.	AutoMod	Evolutionary strategy
Evolutionary Optimizer	Imagine That, Inc.	ExtendSim	Evolutionary strategy
OptQuest	OptTek Systems, Inc.	FlexSim, @RISK, Simul8, Simio, Arena, SIMPROCESS, Crystal Ball, AnyLogic, Enterprise Dynamics, ModelRisk	Scatter search, tabu search, neural networks, integer programming
SimRunner	ProModel Corp.	ProModel, MedModel, ServiceModel	Genetic algorithms and evolutionary strategies
RISKOptimizer	Palisade Corp.	@RISK	Genetic algorithm
WITNESS Optimizer	Lanner Group, Inc.	WITNESS	Simulated annealing, tabu search, hill climbing
GoldSim Optimizer	GoldSim Technology Group	GoldSim	Box's complex method
Plant Simulation Optimizer	Siemens AG	Siemens PLM software	Genetic algorithm
ChaStrobeGA	N/A	Stroboscope	Genetic algorithm
Global Optimization toolbox	The MathWorks	SimEvents (Matlab)	Genetic algorithms, simulated annealing, pattern search

but these comparisons are fairly dated, are inconclusive about which algorithms perform better in different situations, and compare only a small subset of available algorithms. One difficulty lies in the inherent difficulty of comparing solutions between algorithms over true black-box simulations, as one does not usually know the true optimal point and can only compare between noisy estimates observed by the solvers. Less impeding difficulties, but difficulties nonetheless, include the need to interface algorithms to a common wrapper, the objective comparison with solvers that incorporate random elements as their results may not be reproducible, and lack of standard test simulations for purposes of benchmarking.

The benchmarking of algorithms in mathematical programming is usually done by performance profiles (Dolan & Moré 2002), where the graphs show the fraction of problems solved after a certain time. For derivative-free algorithms, data profiles are commonly used (Moré & Wild 2009), where the fraction of problems solved after a certain number of iterations (function evaluations) or 'simplex gradients' is shown. The definition of when a problem is 'solved' may vary—when the true global optimum is known, the solutions found within a certain tolerance of this optimal value may be called solutions, but when this optimum is not known, the solvers that find the best solution (within a tolerance) for a problem, with respect to the other solvers being compared, may be said to have solved the problem. The latter metric may also be used when function evaluations are expensive, and no solver is able to reach within this tolerance given the limited simulation budget.

In both of these cases, the output of the simulations are deterministic, and so it is clear as to which algorithms have performed better than others on a particular problem. In simulation optimization, however, usually one does not know the true solution for the black box system, nor does one see deterministic output. All that one possesses are mean values and sample variances obtained from sample paths at different points. There does not exist a standard method to compare simulation optimization algorithms on large test beds. Many papers perform several macroreplications and report the macroreplicate average of the best sample means (along with the associated sample variance) at the end of the simulation budget. The issue with this is that the performance of the algorithms with different simulation budgets is not seen, as in the case of performance or data profiles. Other papers report the average number of evaluations taken to find a sample mean that is within the global tolerance for each problem. Here, results are listed for each problem and one

does not get an idea of overall performance. In addition, the difference in sample variance estimates is not highlighted. As simulation optimization develops, there is also a need for methods of comparison of algorithms on test beds with statistically significant number of problems.

With regard to standardized simulation testbeds, to our knowledge, the only testbed that provides practical simulations for testing simulation optimization algorithms is available at www.simopt.org (Pasupathy & Henderson 2011). At the point of writing this paper, just 20 continuous optimization problems were available from this repository. Most testing and comparisons happen with classical test problems in nonlinear optimization (many of which have been compiled in Rios & Sahinidis (2013) and available at http://archimedes.cheme.cmu.edu/?q=dfocomp), to which stochastic noise has been added. There is a need for more such repositories, not only for testing of algorithms over statistically significant sizes of problem sets, but for comparison between different classes of algorithms. The need for comparison is evident, given the sheer number of available approaches to solving simulation optimization problems, and the lack of clarity and lack of consensus on which types of algorithms are suitable in which contexts.

As observed by several papers (Fu et al. 2000; Tekin & Sabuncuoglu 2004; Hong & Nelson 2009), there continues to exist a significant gap between research and practice in terms of algorithmic approaches. Optimizers bundled with simulation software, as observed in Section 2.4, tend to make use of algorithms which seem to work well but do not come with provable statistical properties or guarantees of local or global convergence. Academic papers, on the other hand, emphasize methods that are more sophisticated and prove convergence properties. One reason that may contribute to this is that very few simulation optimization algorithms arising from the research community are easily accessible. We wholeheartedly encourage researchers to post their executable files, if not their source code. This could not only encourage practitioners to use these techniques in practice, but allow for comparisons between methods and the development of standardized interfaces between simulations and simulation optimization software.

2.6 CONCLUSIONS

The field of simulation optimization has progressed significantly in the last decade, with several new algorithms, implementations, and applications. Contributions to the field arise from researchers and practitioners in the industrial engineering/operations research, mathematical programming, statistics and machine learning, as well as the computer science communities. The use of simulation to model complex, dynamic, and stochastic systems has only increased with computing power and availability of a wide variety of simulation languages. This increased use is reflected in the identification and application of simulation and simulation optimization methods to diverse fields in science, engineering, and business. There also exist strong analogies between, and ideas that may be borrowed from recent progress in related fields. All of these factors, along with the ever increasing number of publications and rich literature in this area, clearly indicate the interest in the field of simulation optimization, and we have tried to capture this in this paper.

With increased growth and interest in the field, there are also arise opportunities. Potential directions for the field of simulation optimization are almost immediately apparent. Apart from the ability to handle simulation outputs from any well-defined probability distribution, the effective use of variance reduction techniques when possible, and the improvement in theory and algorithms, there is a requirement to address (1) large-scale problems with combined discrete/continuous variables; (2) the ability to effectively handle stochastic and deterministic constraints of various kinds; (2) the effective utilization of parallel computing at the linear algebra level, sample replication level, iteration level, as well as at the algorithmic level; (3) the effective handling of multiple simulation outputs; (4) the incorporation of performance measures other than expected values, such as risk; (5) the continued consolidation of various techniques and their potential synergy in hybrid algorithms; (6) the use of automatic differentiation techniques in the estimation of simulation derivatives when possible; (7) the continued emphasis on providing guarantees of convergence to optima for local and global optimization routines in general settings; (8) the availability and ease of comparison of the performance of available approaches on different applications; and (9) the continued reflection of sophisticated methodology arising from the literature in commercial simulation packages.

Table 2.5: Academic simulation optimization implementations

Algorithm	Type	Citation
Continuous		
SPSA	Stochastic Approximation	Spall (2003a)
SPSA 2nd Order	Stochastic Approximation	Spall (2003a)
SKO	Global response surface	Huang et al. (2006)
CE method	Cross-entropy	Kroese et al. (2006)
APS	Nested partitioning	Kabirian & Ólafsson (2007)
SNOBFIT	Multi-start local response surface	Huyer & Neumaier (2008)
CMA-ES	Evolutionary strategy	Hansen (2011)
KGCP	Global response surface	Scott et al. (2011)
STRONG	Local response surface, trust region	Chang et al. (2013)
GR	Golden Region search	Kabirian & Ólafsson (2011)
SNM	Direct search (Nelder-Mead)	Chang (2012)
DiceOptim	Global response surface	Roustant et al. (2012)
Discrete		
KG	Global response surface	Frazier et al. (2009)
COMPASS	Neighborhood search (integer-ordered problems)	Xu et al. (2010)
R-SPLINE	Neighborhood search (integer-ordered problems)	Wang et al. (2012)
Discrete and continuous		
MRAS	Estimation of distribution	Hu et al. (2005; 2007)
NOMADm	Mesh adaptive direct search	Abramson (2007)

3

THEORETICAL ANALYSIS OF TRUST REGION-BASED SIMULATION OPTIMIZATION

3.1 INTRODUCTION

3.1.1 *Problem definition*

The general problem addressed by this chapter is the unconstrained minimization of the expectation of a stochastic function, formulated as

$$\min_{x \in \mathbb{R}^d} \quad E_\omega[F(x, \omega(\xi))] \tag{P1}$$

This is also known as the Continuous Optimization via Simulation (COvS) problem, where the objective is to minimize the defined performance metric, $f(x) := E_\omega[F(x, \omega)]$. Here $F : \mathbb{R}^d \times \mathbb{R}^p$ involves a set of controls x as well as the realization of random variables ω.

This problem may also be described as zeroth-order stochastic smooth nonconvex optimization for expensive stochastic functions. This implies that (1) derivatives of the objective function f are not known; (2) the F in the objective function is corrupted by stochastic noise; (3) no assumptions on convexity are made, but f is assumed to be continuously differentiable; and (4) the objective function is costly—in terms of time, money, or resources—to evaluate. The expense of evaluation necessitates judicious sampling, and implies that we may not be able to afford to determine good derivative estimates through sampling.

3.1.2 *Prior work and contributions of our work*

We will focus on algorithms based on trust regions, as our framework also relies on trust region concepts. The basic idea is to use a regression and/or interpolation surface as a surrogate model and control its region and quality of approximation using trust region iterates. The development that follows this section is most closely related to the work by papers in the DFO area such as those by Powell (2002); Wild et al. (2008), and stochastic extensions to this by Deng & Ferris (2006; 2009), Chang et al. (2007; 2013), and Larson (2012); Larson & Billups (2014).

Deng & Ferris (2006) use a quadratic interpolation model over averaged function values through replications at points. This work is inspired by the UOBYQA algorithm (Powell 2002). The algorithm makes the decision of where to dedicate sampling effort based on which coefficients in the model affect the variability the most, and consequently which sample site is responsible for this.

The work of Deng & Ferris (2009) is also based on the UOBYQA algorithm, and integrates a Bayesian variable-number sample-path scheme. This builds on traditional sample-path methods which deal with fixed sample-paths, by reducing simulation effort in earlier iterations while still guaranteeing convergence. In the sample-path method, also known as sample path optimization, a fixed set of sample paths is used for sampling in the decision space (Healy & Schruben 1991; Shapiro 1991). As a result, for problems where the distribution of sample paths is allowed to vary across the variable space, this method may not be applicable. The analysis also assumes that simulation outputs follow a normal distribution. The sampling technique used to ensure sufficient decrease with high probability is based on a frequentist criterion by Monte-Carlo sampling from the estimated posterior distribution of the gradient. We do not make use of sample paths or the assumption of normality, and provide a different sampling scheme.

The above methods both build quadratic models within the trust regions. Unique quadratic models require a number of function evaluations that are quadratic in the dimension of the problem, namely $\frac{(d+1)(d+2)}{2}$.

The works by Chang et al. (2007; 2013) combine ideas from traditional response surface methodology and from trust region methods. The idea is to sample with replication at points that are recommended by an experimental design and to use this to fit a linear or quadratic model, and add a hypothesis test step to the trust region

update process. The above methods require that linear and quadratic polynomials are built around specific experimental design points. Convergence guarantees are provided for the case when reliable gradient estimates are available. The algorithm is designed for use when function evaluations are cheap.

Further, the work of Chang et al. (2007) and Chang et al. (2013) rely on experimental design points for interpolation within the trust region, and this approach does not make effective use of previously sampled points that may lie within the trust region but are not part of the experimental design. This adds to the computational burden of the method, especially when samples are expensive to obtain.

Larson & Billups (2014) provide a method that converges almost surely to a stationary point for Problem P1, and prove this without having to perform sample replications or being limited by fixed experimental designs. Though the possibility of convergence without resorting to sample replications is appealing, this also means that a post-optimization clean-up phase needs to be performed in order to determine the incumbent or best found solution, especially in cases where the computational budget is exhausted. The analysis provided is limited to the case when the surrogate models used are first-order polynomials. In addition, as acknowledged by the paper, convergence has been shown for the particular case of uniform additive Gaussian noise, and the use of previously sampled points is limited.

The work we present makes use of Gaussian process (GP) regression and interpolation models within an iterative trust-region framework. The use of a GP framework allows us to reduce the number of sample points that we require to build our models uniquely. We require a number of sample points that are linear in the dimension of the problem (as in Powell (2009); Wild et al. (2008)).

The method we propose makes use of replications to (1) ensure descent with high probability through hypothesis tests; to (2) improve the regression models by pinning down the intrinsic variance parameter (Section 4.3.1); and to (3) to build regression models satisfy the fully linear property (Section 3.2.1) with high probability. The choice of using replication allows the identification of an incumbent solution when the algorithm is terminated early. In this way, the use of replications allows us to include within the algorithmic procedure the validation of a solution—a common practice in discrete-event simulation. In addition, parallelization can facilitate replication through multiple simultaneous simulations.

In summary, the algorithmic framework we provide tackles the limitations in existing literature by providing a global convergence guarantee to stationary points in more general noise settings, without the availability of gradient estimates, while making effective use of previously sampled points. The approach is described below.

We begin by providing a background to the key algorithmic elements in Section 3.2. We discuss in detail the algorithmic structure in Section 3.3. Theory regarding the convergence of the method is pursued in Section 3.4. Chapter 4 extends this framework to a practical implementation we call SO-LVIT (or Simulation Optimization—Learning Via Trust regions).

3.2 MATHEMATICAL PRELIMINARIES

In this section we introduce three concepts, namely Gaussian process regression, fully linear models, and hypothesis tests. These concepts underpin the algorithm and all of the theory that we develop in subsequent sections.

First, we specify the class of functions that we consider in this framework.

Assumption 3.1. The underlying function f is continuously differentiable and has a Lipschitz continuous gradient everywhere in its domain:

$$\frac{|\nabla f(x) - \nabla f(y)|}{\|x - y\|} \leq \gamma_f.$$

Assumption 3.2. The underlying function f is bounded from below and has a compact domain.

The above two assumptions are standard assumptions made in the derivative-free optimization literature and are essential for our proof of convergence to a stationary point, which we develop in the following sections.

3.2.1 *Fully linear models*

In order to show that derivative-free trust region-based algorithms converge, we need to bound the deviation of the function or gradient values of the model we use

from the true surface. Conn et al. (2009) define this deviation through so-called fully linear models, which we restate.

Definition 3.1. Let a function $f : \mathbb{R}^d \to \mathbb{R}$ that satisfies Assumption 3.1 be given. A class of models $\mathbb{M} = \{m : \mathbb{R}^d \to \mathbb{R}\}$ is called fully linear if the following hold

1. There exist positive constants κ_{eg} and κ_{ef} such that for any point x in the domain of f and $\Delta \in (0, \Delta_{\max}]$, there exists a model function $m \in \mathcal{M}$, such that the error between the gradient of the function and the gradient of the model satisfies
   - $\|\nabla f(x+s) - \nabla m(x+s)\| \leq \kappa_{eg}\Delta$ for all $s \in \mathcal{B}(0, \Delta)$, and
   - $|f(x+s) - m(x+s)| \leq \kappa_{ef}\Delta^2$ for all $s \in \mathcal{B}(0, \Delta)$.
2. For this class $\mathcal{M}$ there exists an algorithm, called a 'model-improvement' algorithm, that in a finite, uniformly bounded (with respect to x and Δ) number of steps can
   - either provide a certificate that a given model $m \in \mathcal{M}$ is fully linear on $\mathcal{B}(x, \Delta)$, or
   - find a model $\tilde{m} \in \mathcal{M}$ that is fully linear on $\mathcal{B}(x, \Delta)$.

As mentioned in Section 3.2.2, we will be working with Gaussian Process regression and interpolation models. For the class of interpolating Gaussian Process models (Kriging models), Wild & Shoemaker (2011) have established that they are fully linear for some categories of covariance functions.

In the analysis through this chapter, we will refer to the building of a fully linear model as the assimilation of interpolation points in the variable space with appropriate geometry, such that corresponding interpolation models would be fully linear if there were no error between model values and function values at the interpolation points.

3.2.2 *Gaussian process regression*

For the choice of a surrogate model, we look to those other than the quadratic models that have traditionally been used in trust-region methods.

Gaussian Process (GP) regression models provide a way of approximating a function from a set of multivariate data. There are many interpretations of GP

regression models. The basic idea is to treat the regressed function as a very long vector, and then perform statistical inference on the distribution over this vector conditioned on the observations at the points corresponding to the available data. The inference is done in a Bayesian manner, by assuming a prior distribution over functions—this is necessary in practice, otherwise the models would allow arbitrary variation in the function—and then performing kernelized linear regression (Bishop 2006; Rasmussen & Williams 2006).

GP regression starts by assuming that observations are made from the underlying function with additive Gaussian noise with a constant, but unknown variance, ω. This induces a joint Gaussian distribution over the observations, $\tilde{F}$, conditioned on the true values, $\tilde{f}$, assuming that the observations are independent. The prior on the distribution of the $\tilde{f}$ vector is given as Gaussian with zero mean, and a chosen covariance function that embeds some notion of smoothness on the model. The covariance function Φ involves a distance measure that increases correlation between points that are closer to each other in the input space.

From the two distributions of the prior, $p(\tilde{f})$ and the conditional distribution $p(\tilde{F}|\tilde{f})$, it is straightforward to derive the marginal distribution $p(\tilde{F})$, which has a covariance function given by C, where $C(x_i, x_j) = \phi(x_i, x_j) + \omega\mathcal{I}_{ij}$, where $\mathcal{I}_{ij}$ is 1 if $i = j$ and 0 otherwise. What we are interested is the mean when this marginal distribution is conditioned on a new observation, F_{N+1}. The mean gives us a prediction of the model value at this new point, and is given by

$$m(x_{N+1}) = \phi^T C_N^{-1} \tilde{F}, \tag{3.1}$$

where N is the number of points included in the regression/interpolation. This can be interpreted either as a linear combination of observations F, or a linear combination of the basis functions ϕ, which may be written as

$$m(x_{N+1}) = \sum_{i=1}^{N} \nu_i \phi(x_{N+1}, x_i), \tag{3.2}$$

where ν_i is the ith component of $C_N^{-1}\tilde{F}$ Here, ϕ may take a number of forms, one of which could be $\phi(x_{N+1}, x_i) = \exp(-||x_{N+1} - x_i||^2/\theta^2)$, known as the isotropic Gaussian covariance function. If ϕ involve this sort of distance metric, the model can be seen as an expansion in radial basis functions. The primary task in GP

regression is to choose the model ϕ and to estimate the coefficients θ and ω. Here, θ and ω are called hyperparameters and they may be estimated by maximizing the log-likelihood which is formulated as

$$\ln p(\tilde{F}|\theta,\omega) = -\frac{1}{2}\ln|C_N| - \frac{1}{2}\tilde{F}^T C_N^{-1}\tilde{F} - \frac{N}{2}\ln(2\pi). \tag{3.3}$$

Note that ω is set to zero when GPs are used for interpolation.

These GP models are attractive as they are capable of modeling multi-modality (as the predictive model is composed of linear combinations of log-concave basis functions) and need relatively few function evaluations (on the order of the dimension of the data) to build uniquely.

The following lemma (proved in Wild & Shoemaker (2011)) will be useful to us in the following sections.

Lemma 3.1. *Assuming that the maximum sampled function value is finite and that the hyperparameters are appropriately bounded, an interpolating model of the form m defined in Equation 3.2 is twice continuously differentiable in a region $\mathcal{B} = \{x \in \mathbb{R}^d : ||x - x_b|| \leq \Delta < \infty\}$, where x_b are the interpolation points that satisfy the geometric condition for full linearity (Lemma 3.3), with*

$$\max_{x\in\mathcal{B}} \left\|\nabla^2 m_k(x)\right\| \leq \kappa_H < \infty.$$

In particular, Lemma 4.2 and Theorem 4.5 from Wild & Shoemaker (2011) show that Lemma 3.1 holds if we (1) use one of the recommended basis functions (such as the Gaussian RBF mentioned above); and (2) maintain the fully linear condition, by ensuring that a metric related to the geometry of the interpolation points is bounded (Lemma 3.3). Lemma 3.1 also implies that the gradient of the interpolating model, ∇m, is Lipschitz continuous on $\mathcal{B}_{\max}$, which is the ball containing the interpolation points and having a radius not greater than some constant factor of $\Delta_{\max}$. We denote the corresponding Lipschitz constant as γ_m. We expect that Lemma 3.1 will hold for the regression models we use as well, although we do not explicitly prove this.

Assumption 3.3. The sample points chosen in an iteration in Algorithm 3.1 can be exactly interpolated by the GP regression model.

This assumption is hardly restrictive, as GP regression models are very flexible (Rasmussen & Williams 2006).

3.2.2.1 *Trust region subproblem*

Trust region subproblems involve the minimization of the surrogate model being used—whether it is a Taylor-expansion based model as in nonlinear programming, or an interpolation or regression model in DFO—within the trust region radius. The related trust region subproblem in our context is

$$\begin{aligned} \min_{s_k \in \mathbb{R}^d} \quad & \sum_{i=1}^{|Y|} \nu_i \phi(s_k, y_i) \\ \text{s.t.} \quad & ||s_k|| \leq \Delta_k \end{aligned} \tag{P2}$$

where the trust region is in the ball defined by $\mathcal{B}(x_k, \Delta_k)$. For notational convenience, we use the trust region center x_k as a reference point. Here, s_k denotes the step from the the trust region center x_k, and Y is the matrix of interpolation/regression points $\{y_1, \ldots, y_{|Y|}\}$, which also correspond to displacements from x_k.

The trust region subproblem (TRSP), described in P2, involves the minimization of the surrogate model within the trust region. Traditional nonlinear programming algorithms make use of a quadratic model, where exact gradients and Hessians are available. Specialized algorithms to quickly compute a solution to the TRSP are described in Nocedal & Wright (1999) and Conn et al. (2000). Similar methods may be used for derivative-free optimization problems that use a quadratic model built by interpolation or regression. For general models, Conn et al. (2000) develop a method to find a step that provides a guaranteed decrease in the model, as stated below.

Lemma 3.2. *For any $\kappa_d \in (0, 1)$, there exists an $s \in \mathcal{B}_k - x_k$ that satisfies*

$$m_k(x_k) - m_k(x_k + s) \geq \beta_k := \frac{\kappa_d}{2} \left\|\nabla m_k(x_k)\right\| \min\left\{\frac{\left\|\nabla m_k(x_k)\right\|}{\kappa_H}, \Delta_k\right\},$$

for some $\kappa_d > 0$ and this can be done using a backtracking line search along the direction of steepest descent.

3.2.3 *Hypothesis tests*

We use hypothesis tests to guarantee descent with high probability at each iteration. The idea to use hypothesis tests in trust region optimization was put forth in Chang et al. (2011). The test is set up to make statements about the difference in means at two of the points within the trust region. The null hypothesis is that the sufficient decrease condition is not achieved, and the alternative is that it is, and is formulated as a one-sided two-sample test.

The null and alternative hypotheses are

$$H_0 : f(x_k) - f(x_k^*) \leq \eta_0^2 \beta_k,$$

$$H_1 : f(x_k) - f(x_k^*) > \eta_0^2 \beta_k.$$

Depending on the knowledge of the underlying distribution, we can choose an appropriate test. For the case where the noise is Gaussian and the variance is unequal, it is called the Behrens-Fisher problem. For the case when the distributions are unknown but the variances are equal, one may use a nonparametric permutation test, for example. In these and in more general cases, there exist several tests that one may use based on knowledge of the underlying distribution (symmetry, normality, equal variances, and so on) (Lehmann & Romano 2005). At each iteration k in the algorithm, we test the null hypothesis against a α_k-significance level.

Assumption 3.4. The underlying distribution at any point x, has a finite expected value and a bounded nonzero variance σ, i.e., $0 < \sigma^2(x) < \infty$ for all $x \in \mathbb{R}^d$.

Assumption 3.5. The sample mean estimator $\widehat{f}(x) := \frac{1}{n} \sum_{i=1}^{n} F(x, \omega_i)$ of $f(x)$ for any $x \in \mathbb{R}^d$ satisfies $\sup_{x \in \mathbb{R}^d} |\widehat{f}(x) - f(x)| \to 0$ w.p.1 as $n \to \infty$.

These assumptions provide regularity conditions on the underlying distribution at any point in the domain of f, and Assumption 3.5 requires that the estimator follows the uniform law of large numbers.

3.3 ALGORITHM

The algorithm we use makes use of trust-region methods in combination with nonparametric statistics. The basic idea is to use a nonparametric interpolation/re-

gression model (Gaussian Process regression) within a trust region, and embed a hypothesis test in the trust region update step. Algorithm 3.1 provides details.

When the trust region is relatively large, we employ a GP regression model as we use only a few sample replications at each point. Another reason we use GP regression is that it is possible, in the case of homoscedasticity or when the variance does not vary too much across the decision space, to estimate the variance through samples at different locations (Chapter 4) and embed this information in the regression model. When the trust region is smaller, we switch to an interpolating model where we manage the number of replications carefully. This is discussed in more detail at the end of this section.

Once a model is built, we evaluate the gradient of the model at the incumbent solution. If this is smaller than a pre-defined value ϵ_c, then we investigate whether this estimate is accurate, and we enter what is called the criticality step in the literature. The purpose of this step is to maintain the trust region radius comparable to this estimate of the gradient, while building fully linear models to retain accuracy. This is done so that when the gradient estimate is close to zero, so is the trust region radius—thus leading to an accurate model and therefore an accurate estimate of the true gradient as well. Algorithm 3.3 provides the relevant implementation details. We show in Section 3.4 that Algorithm 3.3 will eventually terminate if the incumbent solution is not a stationary point. The aspect that differs from the traditional criticality step is the requirement on the number of samples at the interpolation points whenever the trust region decreases. Replication, in addition to maintaining a fully linear model, is used to ensure the accuracy of the model.

After the criticality step, an improvement step is sought by finding a solution to the trust region subproblem (Problem P2). An approximate solution that ensures a certain sufficient decrease can be guaranteed (Lemma 3.2).

For the trust region updates, we use two criteria—a traditional ratio test and a hypothesis test, to ensure decrease with high probability. The ratio test computes the ratio of estimated decrease in function between the incumbent solution and the step recommended from the trust region subproblem, to the decrease in model value between the same two points. The hypothesis test sets up the null hypothesis to be that the incumbent is better than the step being investigated. The trust region update moves the trust region center to this point if enough evidence is collected to reject the null hypothesis.

The trust region management also involves updating the size of the trust region. We increase and decrease the trust region cautiously. In traditional nonlinear programming, the trust region radius becomes bounded away from zero as the algorithm converges to a stationary point. We do not want this to happen, as the radius provides a dual purpose—to restrict the step of the optimization, and to provide a region where the model is accurate. We do not decrease it too hastily because the smaller the trust region radius gets, the harder it is to distinguish between points due to noise. Therefore, we ensure through replications that an unfavorable ρ_k value results from the step size being too large, and not due to poor model accuracy.

Having said this, we would like to minimize simulation replications as much as possible. We do not exceed the χ_0, χ_1 simulation replication budget when the trust region radius is 'large', thus attributing the trust region step failures to poor models due to the large radius, and decrease the trust region accordingly. When trust regions are smaller, we not only want to ensure that the models are more accurate (through replications), but also want to prevent further shrinking of the trust region, as mentioned in the previous paragraph. To control what we consider 'large' or 'small', we use the threshold parameter Δ_c, below which we rigorously improve the model quality both through model geometry improvements, as well as through an increase in simulation replications according to the formula derived in Lemma 3.5. The rules involving the trust region management are detailed in Algorithm 3.2. Note that our algorithmic framework requires that the result of the ratio test has to be strictly positive ($\eta_0 > 0$) to consider a new solution to be the incumbent solution.

Similar to prevailing DFO algorithms, we make use of a model-improvement step if the ratio test does not have the ideal outcome ($\rho_k \geq \eta_1$). This involves certifying that the model is fully linear in the current iteration, and, if not, building one that is. In this way, we are not required to build fully linear models at every iteration.

The iteration count is incremented and we return to the model building step. Section 3.4 is devoted to proving that the above scheme guarantees convergence to a stationary point. Relevant parameters for the algorithm are listed in Table 3.1.

Using analogous terminology of Conn et al. (2009), we call a step **successful** if both the ratio test and the hypothesis test are passed. Any step that results in a decrease in trust region radius is called **unsuccessful**. A step that requires a

call to the model-improving algorithm is called **model-improving**, and all other situations are deemed **acceptable** steps.

Table 3.1: List of initialization parameters to simulation optimization algorithm

Trust region parameters	
Initial trust region size and maximum size	$\Delta_0, \Delta_{\max}$
Control of trust region performance	$0 < \eta_0 < \eta_1 < 1$
Update of trust region size	$0 < \lambda_0 < 1 < \lambda_1$
Threshold parameters	
Threshold trust region radius	$0 < \Delta_c < \Delta_{\max}$
Threshold model gradient to enter criticality step	$\epsilon_c > 0$
Trust region radius reduction factor in criticality step	$\tau_c \in (0, 1)$
Criticality step threshold parameters	$\mu > \zeta > 0$
Model-building threshold parameters	$\psi_0 \geq 1, \psi_1 \in (0, \psi_0^{-1}]$
Sampling and statistical parameters	
Maximum no. of points in interpolation/regression set	$\pi_{\max}$
Number of samples at TR center for $\Delta_k > \Delta_c$	χ_0
Number of samples at non-TR center points for $\Delta_k > \Delta_c$	χ_1
Initial α-level for hypothesis test	α_0

3.4 CONVERGENCE ANALYSIS

We address the quality of general interpolation models in the presence of stochastic noise. First, we state a geometric condition on fully linear models, one that may be satisfied using Algorithm 4.2 through ψ_1-thresholding of the chosen interpolation points. We state Lemma 4.2 from Wild et al. (2008),

Lemma 3.3. *If $d+1$ affinely independent points are found as per the ψ_1-thresholding in Algorithm 4.2 and included in Y_d,*

$$||Y_d^{-1}|| \leq \frac{d^{(d-1)/2}}{\Delta_k \psi_1} = \frac{\Lambda_{Y_d}}{\Delta_k},$$

where $0 < \psi_1 \leq \psi_0^{-1} \leq 1$.

Algorithm 3.1 Trust region-based simulation optimization framework

Require Initialization parameters listed in Table 3.1, and an initial point $x_0 \in \mathbb{R}^d$. Set iteration counter $k \leftarrow 0$. Set $i \leftarrow 0$, a counter that keeps track of the number of consecutive steps when $\Delta_k \leq \Delta_c$.

Step 1: Locate $d+1$ affinely independent points that lie within a distance Δ_0 from x_0 and perform χ_0 replications at x_0, the trust region center and χ_1 replications at the other points. Using these points, build a GP regression model if $\Delta_0 \geq \Delta_c$ and a GP interpolation model otherwise, using the scheme described in Section 3.2.2 and denote the model m_0.

Step 2: (criticality step)
if $\|\nabla m_k(x_k)\| < \epsilon_c$ **then**
if $m_k(x_k)$ is not fully linear in $\mathcal{B}_k(x_k, \Delta_k)$ or $\Delta_k > \mu \|\nabla m_k(x_k)\|$ **then**
Assign $\Delta_{\text{inc}} \leftarrow \Delta_k$.
Call criticality step (Algorithm 3.3).
$m_k \leftarrow m_k^{(j)}$, and $\Delta_k \leftarrow \min\{\max\{\tau_c^j \Delta_{\text{inc}}, \zeta\|\nabla m_k^{(j)}(x_k)\|\}, \Delta_{\text{inc}}\}$.
end if
end if

Step 3: (step calculation) solve trust-region subproblem (cf. Problem (P2), Lemma 3.2) to obtain a step s_k. Evaluate the function f at $x_k + s_k$ with χ_0 replications if $\Delta_k \geq \Delta_c$ and n_i replications otherwise.

Step 4: (trust region update) Compute a defined measure of fidelity

$$\rho_k := \frac{\widehat{f}(x_k) - \widehat{f}(x_k + s_k)}{m(x_k) - m(x_k + s_k)},$$

and update trust-region parameters using Algorithm 3.2. If the trust region center is moved in this step, update α_k according to Equation 4.1.

Step 5: (model building) Construct model m_{k+1} using Algorithm 4.1.

Step 6: Iterate $k \leftarrow k+1$, and return to **Step 2**.

Algorithm 3.2 Trust region update scheme

if $\rho_k \geq \eta_0$, **then**
  perform hypothesis test according to Equation 3.46 (for example)
  if hypothesis test is passed **then** $x_{k+1} \leftarrow x_k + s^*$
    if $\rho_k \geq \eta_1$ **then**
      if $\Delta_k < \Delta_c$ **then** $\Delta_{k+1} \leftarrow \Delta_{\text{inc}}, i \leftarrow 0$
      else $\Delta_{k+1} \leftarrow \lambda_1 \Delta_k$
      end if
    else
      if $\Delta_k < \Delta_c$ **then** $\Delta_{k+1} \leftarrow \Delta_{\text{inc}}, i \leftarrow 0$
      else $\Delta_{k+1} \leftarrow \Delta_k$
      end if
    end if
  else
    $x_{k+1} \leftarrow x_k$
    if $\Delta_k < \Delta_c$ **then**
      $\Delta_{k+1} \leftarrow \Delta_k; i \leftarrow i + 1$
      Call Algorithm 4.2 in model-improvement mode with the current set of interpolation points Y_k as argument. This will improve the current model m if it is not fully linear in $\mathcal{B}(x_k, \Delta_k)$
    else
      if The current model m is not fully linear within $\mathcal{B}(x_k, \Delta_k)$ **then**
        $\Delta_{k+1} \leftarrow \Delta_k$. Call Algorithm 4.2 in model-improvement mode with current set of regression data points Y_k as argument.
      else
        $\Delta_{k+1} \leftarrow \lambda_0 \Delta_k$. If $\Delta_{k+1} < \Delta_c$, assign $\Delta_{\text{inc}} \leftarrow \Delta_k$.
      end if
    end if
  end if
else if $\rho_k < \eta_0$, **then**
  $x_{k+1} \leftarrow x_k$
  Call Algorithm 4.2 in model-improvement mode to determine if model m if fully linear, and improve it if it is not.
  if The current model m is fully linear within $\mathcal{B}(x_k, \Delta_k)$ **then**
    $\Delta_{k+1} \leftarrow \lambda_0 \Delta_k$.
    if $\Delta_{k+1} < \Delta_c$ **then**
      if i $= 0$ **then**
        $\Delta_{\text{inc}} \leftarrow \Delta_k$
      end if
      $i \leftarrow i + 1$
    end if
  else $\Delta_{k+1} \leftarrow \Delta_k$.
  end if
end if

Algorithm 3.3 Criticality step algorithm

Initialize counter $j \leftarrow 0$. Assign $m_k^{(0)}(x_k) \leftarrow m_k(x_k)$ and $l_0 \leftarrow \chi_0$ if $i = 0$ and $l_0 \leftarrow n_i$ if $i > 0$.
repeat
    Increment $j \leftarrow j + 1$
    if model $m^{(j-1)}$ is not fully linear in $\mathcal{B}(x_k, \tau_c^{j-1}\Delta_{\text{inc}})$ **then**
        Call model-improvement algorithm to identify points for building a fully linear model
        Perform replications according to Lemma 3.5, i.e., until the number of replications at the required points is $l_j \leftarrow \lceil l_{j-1}\tau_c^{-4} \rceil + 1$
        Build the model $m^{(j)}$ using Algorithm 4.1 with $\psi_0 = 1$.
    end if
    Compute model gradient $\nabla m_k^{(j)}(x_k)$
until $\tau_c^{j-1}\Delta_{\text{inc}} \leq \mu \left\| \nabla m_k^{(j)}(x_k) \right\|$

Even though Algorithm 4.2 does not guarantee that $d+1$ points satisfying this property are returned, the overall scheme in Algorithm 3.1, through the evaluation of model-improving points, can provide this guarantee in a finite number of steps. Note that the bound in Lemma 3.3 is important for Lemma 3.4.

Next, we characterize the discrepancy between the model and the true function, as well as the gradients of the model and the true underlying function. The development follows that presented in Wild (2009), but extends it to the case of stochastic functions.

Lemma 3.4. *We can bound the error in the model as*

$$\begin{aligned} |m(x_k + s) - f(x_k + s)| \leq & \frac{\Delta_k^2}{2}(\gamma_f + \gamma_m)(5\Lambda_{Y_d}\sqrt{d} + 1) \\ & + \sqrt{d}\Lambda_{Y_d}|\widehat{f}(x_k + y_p) - f(x_k + y_p)| \\ & + (\sqrt{d}\Lambda_{Y_d} + 1)|\widehat{f}(x_k) - f(x_k)|, \end{aligned}$$

and the error in the gradient as

$$\begin{aligned} \|\nabla m(x_k + s) - \nabla f(x_k + s)\| \leq & \frac{5}{2}\Lambda_{Y_d}\sqrt{d}\Delta_k(\gamma_m + \gamma_f) \\ & + \frac{\sqrt{d}\Lambda_{Y_d}}{\Delta_k}(|\widehat{f}(x_k + y_p) - f(x_k + y_p)| \\ & + |\widehat{f}(x_k) - f(x_k)|), \end{aligned}$$

where y_p is any point contained in a subset of the interpolation points Y_d, $s \in \mathcal{B}(x_k, \Delta_k)$, and Λ_{Y_d} is a finite constant that satisfies $\left\|Y_d^{-1}\right\| \leq \Lambda_{Y_d}/\Delta_k$.

Proof. Let the error between the model and the true underlying function be

$$e^m(s) := m(x_k + s) - f(x_k + s), \tag{3.4}$$

where x_k is a base point (which corresponds to the incumbent solution and trust region center in our algorithm), and s is a displacement from this base point (which corresponds to a step from the trust region center in our algorithm). Let the error in the gradient be

$$e^g(s) := \nabla m(x_k + s) - \nabla f(x_k + s). \tag{3.5}$$

We perform a Taylor expansion of the model around the interpolation points,

$$\begin{aligned} m(x_k + y_p) &= m(x_k + s + (y_p - s)) \\ &= m(x_k + s) + \int_0^1 \nabla m(x_k + s + t(y_p - s))^T (y_p - s)\, \mathrm{d}t, \end{aligned} \tag{3.6}$$

where $Y = \{y_1, \ldots, y_\pi\}$ is the set of displacements from the trust region center that correspond to interpolation points. Similarly, we do this for the function f,

$$f(x_k + y_p) = f(x_k + s) + \int_0^1 \nabla f(x_k + s + t(y_p - s))^T (y_p - s)\, \mathrm{d}t. \tag{3.7}$$

The task is to provide bounds on the quantities e_m as well as e_g. We start by taking the inner product of the error in the gradient at a point s with the displacement of the interpolation points from s,

$$\begin{aligned} e^g(s)^T (y_p - s) &= [\nabla m(x_k + s) - \nabla f(x_k + s)]^T (y_p - s) \\ &= \nabla m(x_k + s)^T (y_p - s) - \nabla f(x_k + s)^T (y_p - s) \\ &\quad + e^m(s) - e^m(s), \end{aligned} \tag{3.8}$$

for all $y_p \in Y$, where we have added and subtracted the term $e^m(s)$.

Substituting for $e^m(s)$ using Equations 3.4, 3.6 and 3.7, we get for each $y_p \in Y$,

$$\begin{aligned} e^g(s)^T(y_p - s) =& \nabla m(x_k + s)^T(y_p - s) - \nabla f(x_k + s)^T(y_p - s) - e^m(s) \\ & - \int_0^1 \nabla m(x_k + s + t(y_p - s))^T(y_p - s)\,\mathrm{d}t + m(x_k + y_p) \\ & + \int_0^1 \nabla f(x_k + s + t(y_p - s))^T(y_p - s)\,\mathrm{d}t - f(x_k + y_p). \end{aligned} \tag{3.9}$$

We consolidate the terms to get

$$\begin{aligned} e^g(s)^T(y_p - s) =& \int_0^1 [\nabla f(x_k + s + t(y_p - s)) - \nabla f(x_k + s)]^T(y_p - s)\,\mathrm{d}t \\ & - \int_0^1 [\nabla m(x_k + s + t(y_p - s)) - \nabla m(x_k + s)]^T(y_p - s)\,\mathrm{d}t \\ & + m(x_k + y_p) - f(x_k + y_p) - e^m(s), \end{aligned} \tag{3.10}$$

for all $y_p \in Y$. Now, for the trust region center, where $y_1 = 0$, this becomes

$$\begin{aligned} e^g(s)^T(-s) =& \int_0^1 [\nabla f(x_k + s - ts) - \nabla f(x_k + s)]^T(-s)\,\mathrm{d}t \\ & - \int_0^1 [\nabla m(x_k + s - ts) - \nabla m(x_k + s)]^T(-s)\,\mathrm{d}t \\ & + m(x_k) - f(x_k) - e^m(s). \end{aligned} \tag{3.11}$$

We can now subtract Equation 3.11 from Equation 3.10 to get an expression that is purely dependent on the error in the gradient and the set of interpolation points on the left-hand side:

$$\begin{aligned} e^g(s)^T y_p =& \int_0^1 [\nabla f(x_k + s + t(y_p - s)) - \nabla f(x_k + s)]^T(y_p - s)\,\mathrm{d}t \\ & - \int_0^1 [\nabla m(x_k + s + t(y_p - s)) - \nabla m(x_k + s)]^T(y_p - s)\,\mathrm{d}t \\ & - \int_0^1 [\nabla f(x_k + s - ts) - \nabla f(x_k + s)]^T(y_p - s)\,\mathrm{d}t \\ & + \int_0^1 [\nabla m(x_k + s - ts) - \nabla m(x_k + s)]^T(-s)\,\mathrm{d}t \\ & + m(x_k + y_p) - f(x_k + y_p) - m(x_k) + f(x_k). \end{aligned} \tag{3.12}$$

Now, we attempt to bound the left-hand side of Equation 3.12 by individually bounding each term on the right-hand side. We start with the first term, where

$$\begin{aligned}&\left\|\int_0^1 [\nabla f(x_k + s + t(y_p - s)) - \nabla f(x_k + s)]^T (y_p - s)\,\mathrm{d}t\right\| \\ &\quad \leq \int_0^1 \|\nabla f(x_k + s + t(y_p - s)) - \nabla f(x_k + s)\|\,\|(y_p - s)\|\,\mathrm{d}t,\end{aligned} \tag{3.13}$$

due to the Cauchy-Schwarz inequality. Further, due to Assumption 3.1, this is

$$\begin{aligned}&\leq \int_0^1 \gamma_f \|t(y_p - s)\|\,\|y_p - s\|\,\mathrm{d}t \\ &= \int_0^1 \gamma_f t \|y_p - s\|^2\,\mathrm{d}t \\ &\leq \int_0^1 \gamma_f t \|2\Delta_k\|^2\,\mathrm{d}t,\end{aligned} \tag{3.14}$$

where the last inequality is due to the fact that both s and y_p are within the trust region of radius Δ_k. Then, we can state

$$\begin{aligned}&\left\|\int_0^1 [\nabla f(x_k + s + t(y_p - s)) - \nabla f(x_k + s)]^T (y_p - s)\,\mathrm{d}t\right\| \\ &\quad \leq 4\gamma_f \Delta_k^2 \int_0^1 t\mathrm{d}t = 4\gamma_f \Delta_k^2 \left[\frac{t^2}{2}\right]_0^1 = 2\gamma_f \Delta_k^2.\end{aligned} \tag{3.15}$$

We can perform an identical procedure for the second, third, and fourth terms in Equation 3.12, using the relevant Lipschitz constant γ_m or γ_f to get

$$\begin{aligned}[\nabla m(x_k + s) - \nabla f(x_k + s)]^T (y_p) \leq \frac{5}{2}\Delta^2(\gamma_f + \gamma_m) + |m(x_k) - f(x_k)| \\ + |m(x_k + y_p) - f(x_k + y_p)|.\end{aligned} \tag{3.16}$$

Now, since our model m interpolates $\widehat{f}$,

$$m(x_k + y_p) = \widehat{f}(x_k + y_p) \qquad \text{for all } y_p \in Y. \tag{3.17}$$

Therefore,

$$\begin{aligned}[\nabla m(x_k + s) - \nabla f(x_k + s)]^T (y_p) \leq \frac{5}{2}\Delta^2(\gamma_f + \gamma_m) + |\widehat{f}(x_k) - f(x_k)| \\ + |\widehat{f}(x_k + y_p) - f(x_k + y_p)|.\end{aligned} \tag{3.18}$$

Now, while we may interpolate at more than $d+1$ points, we restrict our attention to any d of these points barring the trust region center, and call this interpolation submatrix as Y_d. Now as Y_d is square and nonsingular by construction (Lemma 3.3), we may write,

$$\|e^g(s)\| = \left\|Y_d^{-T} Y_d^T e^g(s)\right\| \leq \left\|Y_d^{-T}\right\| \left\|Y_d^T e^g(s)\right\|. \tag{3.19}$$

Further,

$$\|e^g(s)\| \leq \frac{\Lambda_{Y_d}}{\Delta_k} \left\|Y_d^T e^g(s)\right\|. \tag{3.20}$$

using the bound on $\left\|Y_d^{-T}\right\|$ from the statement of this lemma. The enforcement of this bound is described in Lemma 3.3 and Algorithm 4.2. Now, from the equivalence of norms, $\|\cdot\|_2 \leq \sqrt{d}\,\|\cdot\|_\infty$,

$$\begin{aligned}\left\|Y_d^T e^g(s)\right\| \leq & \sqrt{d}\left\|Y_d^T e^g(s)\right\|_\infty \\ \leq & \sqrt{d}\left(\frac{5}{2}\Delta^2(\gamma_f+\gamma_m) + |\widehat{f}(x_k+y_p) - f(x_k+y_p)|\right. \\ & \left. + |\widehat{f}(x_k) - f(x_k)|\right),\end{aligned} \tag{3.21}$$

for some $y_p \in Y_d$. Therefore, using Equations 3.20 and 3.21,

$$\begin{aligned}\|e^g(s)\| \leq & \frac{5}{2}\Lambda_{Y_d}\sqrt{d}\Delta_k(\gamma_m+\gamma_f) + \frac{\sqrt{d}\Lambda_{Y_d}}{\Delta_k}(|\widehat{f}(x_k+y_p) - f(x_k+y_p)| \\ & + |\widehat{f}(x_k) - f(x_k)|).\end{aligned} \tag{3.22}$$

Now, rearranging Equation 3.11,

$$\begin{aligned}e^m(s) = & \int_0^1 [\nabla f(x_k+s-ts) - \nabla f(x_k+s)]^T(-s)\,\mathrm{d}t \\ & - \int_0^1 [\nabla m(x_k+s-ts) - \nabla m(x_k+s)]^T(-s)\,\mathrm{d}t \\ & + m(x_k) - f(x_k) + e^g(s)^T(s).\end{aligned} \tag{3.23}$$

Using the bounds on the first two terms with the triangle inequality, and subsequently substituting for $\|e^g(s)\|$ from Equation 3.22,

$$\begin{aligned}|e^m(s)| \leq & \frac{\gamma_f \Delta^2}{2} + \frac{\gamma_m \Delta^2}{2} + |m(x_k) - f(x_k)| + \|e^g(s)\| \Delta \\ \leq & \frac{\Delta^2}{2}(\gamma_f + \gamma_m) + |\widehat{f}(x_k) - f(x_k)| + \frac{5}{2}\Delta^2 \Lambda_{Y_d} \sqrt{d}(\gamma_f + \gamma_m) \\ & + \sqrt{d}\Lambda_{Y_d} |\widehat{f}(x_k + y_p) - f(x_k + y_p)| + \sqrt{d}\Lambda_{Y_d} |\widehat{f}(x_k) - f(x_k)|.\end{aligned} \tag{3.24}$$

In summary,

$$\begin{aligned}|m(x_k + s) - f(x_k + s)| \leq & \frac{\Delta^2}{2}(\gamma_f + \gamma_m)(5\Lambda_{Y_d}\sqrt{d} + 1) \\ & + \sqrt{d}\Lambda_{Y_d} \left|\widehat{f}(x_k + y_p) - f(x_k + y_p)\right| \\ & + (\sqrt{d}\Lambda_{Y_d} + 1) \left|\widehat{f}(x_k) - f(x_k)\right|.\end{aligned} \tag{3.25}$$

■

Algorithm 3.1 defines a threshold Δ_c, below which, every time the trust region is forced to decrease through Algorithm 3.2, the number of samples increases. This is required in order to ensure that the function value estimates converge to the true value faster than the trust region shrinks. We provide a modified version of the lemma from Chang et al. (2013), where we do not provide a bound for the gradient estimate, as we do not make assumptions of its availability.

Lemma 3.5. *If Assumption 3.4 holds, for any $x_i \in \mathbb{R}^d$, where i is the ith consecutive iteration where the trust region radius is decreased (or maintained) below Δ_c,*

$$P\left(\lim_{i \to \infty} |\widehat{f}(x_i + y) - f(x_i + y)| > \Delta_i^2\right) = 0,$$

if the sampling scheme used is such that $n_{i+1} \geq \lceil n_i \lambda_0^{-4} \rceil + 1$.

Proof. If $x_i + y \in Y_i$, the interpolation set at iteration i, then by Chebyshev's inequality,

$$P\left(|\widehat{f}(x_i + y) - f(x_i + y)| > \Delta_i^2\right) \leq \frac{\sigma^2(x_i + y)}{n_i \Delta_i^4} \leq \frac{\sup_{x \in \mathbb{R}^d} \sigma^2(x)}{n_i \Delta_i^4}, \tag{3.26}$$

where n_i is the minimum number of replications at each of the interpolation points Y_i. If we can show that

$$\sum_{i=1}^{\infty} \frac{\sup_{x \in \mathbb{R}^d} \sigma^2(x)}{n_i \Delta_i^4} < \infty, \tag{3.27}$$

Then we will have completed the proof due to the first Borel-Cantelli lemma. We can achieve the desired target by enforcing, say, the series ratio test to hold (Rudin 1976), namely that

$$\frac{\sup_{x \in \mathbb{R}^d} \sigma^2(x)}{n_{i+1} \Delta_{i+1}^4} \cdot \left[\frac{\sup_{x \in \mathbb{R}^d} \sigma^2(x)}{n_i \Delta_i^2} \right]^{-1} < 1, \tag{3.28}$$

or that

$$n_{i+1} > \frac{n_i \Delta_i^4}{\Delta_{i+1}^4}. \tag{3.29}$$

Since the failed step update rule for the trust region radius is $\Delta_{i+1} = \lambda_0 \Delta_i$, this implies $n_{i+1} > n_i \lambda_0^{-4}$. We can ensure this by assigning $n_{i+1} \geq \lceil n_i \lambda_0^{-4} \rceil + 1$. This completes the proof. ∎

We note that this sample update requirement is marginally different from the one proposed in Chang et al. (2013).

In the previous lemma, we have chosen to bound the error between the estimates and the true function by Δ^2. The reason for this will become apparent in the following lemma, where we bound the difference between the predicted value at a point and the observed value by a similar quantity.

Lemma 3.6. *For any $x_i \in \mathbb{R}^d$, where i is as defined in Lemma 3.5,*

$$P\left(\lim_{i \to \infty} |m_i(x^*) - \widehat{f}(x^*)| > c \cdot \Delta_i^2 \right) = 0,$$

where $c = \frac{\gamma_f + \gamma_m}{2}(5\Lambda_{Y_d}\sqrt{d} + 1) + 2\Lambda_{Y_d}\sqrt{d} + 2$.

Proof. Now, Lemma 3.5 can also be applied to points other than the interpolation points, i.e., at a point x^*. So, $P\left(\lim_{i \to \infty} |\widehat{f}(x^*) - f(x^*)| > \Delta_i^2\right) = 0$.

From the triangle inequality,

$$|m_i(x^*) - \widehat{f}(x^*)| \leq |m_i(x^*) - f(x^*)| + |\widehat{f}(x^*) - f(x^*)|. \tag{3.30}$$

Inserting the result from Lemma 3.4,

$$\begin{aligned} |m_i(x^*) - \widehat{f}(x^*)| \leq & \frac{\Delta^2}{2}(\gamma_f + \gamma_m)(5\Lambda_{Y_d}\sqrt{d} + 1) \\ & + \sqrt{d}\Lambda_{Y_d}|\widehat{f}(x_k + y_p) - f(x_k + y_p)| \\ & + (\sqrt{d}\Lambda_{Y_d} + 1)|\widehat{f}(x_k) - f(x_k)| + |\widehat{f}(x^*) - f(x^*)|. \end{aligned} \tag{3.31}$$

We assign $c = \frac{\gamma_f + \gamma_m}{2}(5\Lambda_{Y_d}\sqrt{d} + 1) + 2\Lambda_{Y_d}\sqrt{d} + 2$ to complete the proof. ∎

Lemma 3.7. *For any $x_i \in \mathbb{R}^d$, where i is as in Lemma 3.5, if $\|\nabla f(x_i)\| > 0$, Algorithm 3.1 can always find a new satisfactory solution.*

Proof. We assume that $\|\nabla f(x_i)\| = \epsilon > 0$. We first consider the ratio comparison test, where

$$\rho_i - 1 = \frac{\widehat{f}(x_i) - \widehat{f}(x^*)}{m_i(x_i) - m_i(x^*)} - 1 = \frac{m_i(x^*) - \widehat{f}(x^*)}{m_i(x_i) - m_i(x^*)}, \tag{3.32}$$

as $\widehat{f}(x_i) = m_i(x_i)$ due to the fact that x_i is an interpolation point. Taking the absolute value of this quantity,

$$|\rho_i - 1| = \frac{|m_i(x^*) - \widehat{f}(x^*)|}{|m_i(x_i) - m_i(x^*)|}. \tag{3.33}$$

Let us assume, for contradiction, that this test fails for infinite consecutive steps, which means $P(\lim_{i\to\infty} \Delta_i > \delta) = 0$, for any $\delta > 0$. Also,

$$\begin{aligned} \|\nabla f(x_i)\| &= \|\nabla f(x_i) - \nabla m_i(x_i) + \nabla m_i(x_i)\| \\ \epsilon &\leq \|\nabla f(x_i) - \nabla m_i(x_i)\| + \|\nabla m_i(x_i)\|. \end{aligned} \tag{3.34}$$

Recall, from Lemma 3.4,

$$\begin{aligned} \|e^g(s)\| \leq & \frac{5}{2}\Lambda_{Y_d}\sqrt{d}\Delta_k(\gamma_m + \gamma_f) + \frac{\sqrt{d}\Lambda_{Y_d}}{\Delta_k}(|\widehat{f}(x_k + y_p) - f(x_k + y_p)| \\ & + |\widehat{f}(x_k) - f(x_k)|). \end{aligned} \tag{3.35}$$

This, along with Lemma 3.5, implies that

$$P(\lim_{i\to\infty} \|\nabla m_i(x_i)\| < \epsilon/2) = 0. \tag{3.36}$$

As $P(\lim_{i\to\infty} \Delta_i > \delta) = 0$ for any $\delta > 0$. Therefore, for sufficiently large i, $\Delta_i \leq \frac{\|\nabla m_i(x_i)\|}{\kappa_H}$, and therefore,

$$\min\left\{\frac{\|\nabla m_i(x_i)\|}{\kappa_H}, \Delta_i\right\} = \Delta_i. \tag{3.37}$$

Now, from Lemma 3.6,

$$P\left(\lim_{i\to\infty} |m_i(x^*) - \widehat{f}(x^*)| > c \cdot \Delta_i^2\right) = 0, \tag{3.38}$$

or when divided through by a constant,

$$P\left(\lim_{i\to\infty} \frac{|m_i(x^*) - \widehat{f}(x^*)|}{\frac{\kappa_d}{2}\|\nabla m_i(x_i)\|\Delta_i} > \frac{c \cdot \Delta_i^2}{\frac{\kappa_d}{2}\|\nabla m_i(x_i)\|\Delta_i}\right) = 0. \tag{3.39}$$

We can then replace the constants $\|\nabla m_i(x_i)\|$ and Δ_i on the right-hand side by $\epsilon/2$ and δ respectively, giving

$$P\left(\lim_{i\to\infty} \frac{|m_i(x^*) - \widehat{f}(x^*)|}{\frac{\kappa_d}{2}\|\nabla m_i(x_i)\|\Delta_i} > \frac{4c\delta}{\kappa_d\epsilon}\right) = 0. \tag{3.40}$$

We use the sufficient decrease condition to replace the denominator on the left-hand side of the inequality to yield

$$P\left(\lim_{i\to\infty} \frac{|m_i(x^*) - \widehat{f}(x^*)|}{|m_i(x_i) - m_i(x^*)|} > \frac{4c\delta}{\kappa_d\epsilon}\right) = 0, \tag{3.41}$$

or

$$P\left(\lim_{i\to\infty} |\rho_i - 1| > \frac{4c\delta}{\kappa_d\epsilon}\right) = 0. \tag{3.42}$$

Choosing $\delta = \frac{(1-\eta_0)\kappa_d\epsilon}{4c}$,

$$P\left(\lim_{i\to\infty} |\rho_i - 1| > 1 - \eta_0\right) = 0, \tag{3.43}$$

or

$$P\left(\lim_{i\to\infty} \rho_i < \eta_0\right) = 0, \tag{3.44}$$

which is a contradiction. Therefore, the ratio test will be passed with probability 1.

Next, we consider the hypothesis test. The algorithm we propose performs the hypothesis test if the ratio test is passed, so

$$\widehat{f}(x_i) - \widehat{f}(x^*) \geq \eta_0(m_i(x_i) - m_i(x^*)) \geq \eta_0\beta_i. \tag{3.45}$$

The proposed test statistic in the Behrens-Fisher case (normally distributed random variables with unequal variances Lehmann & Romano (2005)) is

$$t_i = \frac{\widehat{f}(x_i) - \widehat{f}(x^*) - \eta_0^2\beta_i}{S_i} \geq \frac{\eta_0(1-\eta_0)\beta_i}{S_i}, \tag{3.46}$$

where S_i is the sample standard deviation, and is computed by $S_i^2 = S^2(x_i, n_i)/n_i + S^2(x^*, n_i^*)/n_i^*$.

Now, assume for contradiction that the hypothesis test is failed infinitely consecutively often, or $t_i \leq t_{1-\alpha_k,df}$ for all $i = 1, 2, \ldots$. Then, according to the algorithm, both $n_i \to \infty$ and $n_i^* \to \infty$, thus $S_i^2 \to 0$ w.p.1. Since $\|\nabla f(x_i)\| = \epsilon > 0$ and the ratio test is passed for sufficiently large i, Δ_i is bounded away from zero, and thus

$$\beta_i = \frac{\kappa_d}{2}\|\nabla m_i(x_i)\| \min\left\{\frac{\|\nabla m_i(x_i)\|}{\kappa_H}, \Delta_i\right\} \tag{3.47}$$

is also bounded away from zero w.p.1 for sufficiently large i (using Equation 3.36). Therefore, from Equation 3.46, $\lim_{i\to\infty} t_i \to \infty$ and therefore $t_i > t_{1-\alpha_k,df}$, and therefore the hypothesis test passes, which contradicts our initial assumption. Note that for other hypothesis tests, the corresponding test statistic will fall in the rejection region with high probability (if the null hypothesis is false) as sample sizes increase, which is the case with the above scheme. The above Behrens-Fisher test only serves as an example. ∎

We now show that the criticality step will terminate if the incumbent solution is not a first-order point.

Lemma 3.8. *If $\nabla f(x_k) \neq 0$, Algorithm 3.3 will terminate w.p.1.*

Proof. Suppose $\nabla f(x_k) = \epsilon > 0$.

Initially, either the model $m^{(0)} := m_k$ is not fully linear in $\mathcal{B}(x_k, \Delta_k)$ or the radius is above $\mu \|\nabla m_k(x_k)\|$. The model is then improved using one iteration of Algorithm 3.3 and the new model which we call $m^{(1)}$ is built and it's gradient computed at x_k. If, now, $\Delta_{\text{inc}} \leq \mu \left\|\nabla m^{(1)}(x_k)\right\|$, then we stop the procedure. If this is not the case, we scale the radius by τ_c, increment the algorithm loop counter j, and repeat the procedure.

The only way that Algorithm 3.3 does not terminate is in the case

$$\mu \left\|\nabla m_k^{(j)}(x_k)\right\| < \tau_c^{j-1} \Delta_{\text{inc}}, \tag{3.48}$$

for all $j \geq 1$. This implies that $\lim_{j\to\infty} \left\|\nabla m_k^{(j)}(x_k)\right\| = 0$. Since we ensure that the error between gradients of the model and the true function converges (Lemmas 3.4 and 3.6) to zero as $\Delta_k \to 0$, and because $\|\nabla f(x_k)\| = \epsilon \leq \left\|\nabla f(x_k) - \nabla m_k^{(j)}(x_k)\right\| + \left\|\nabla m_k^{(j)}(x_k)\right\|$,

$$P\left(\lim_{j\to\infty} \left\|\nabla m_k^{(j)}(x_k)\right\| < \frac{\epsilon}{2}\right) = 0. \tag{3.49}$$

This is a contradiction, and therefore Algorithm 3.3 will terminate w.p.1 if the incumbent solution is not a stationary point. ∎

Lemma 3.9. *If the number of successful iterations is finite, then*

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

Proof. After the last successful iteration, the trust region radius is never increased (Algorithms 3.1 and 3.2). If $\Delta_k \geq \Delta_c$, then trust region radius is reduced at least every D steps, where D is a finite number of steps required to make a model fully linear. This will continue until $\Delta_k < \Delta_c$.

If $\Delta_k < \Delta_c$, Lemma 3.7 ensures that we will always be able to find a successful step if $\|\nabla f(x_k)\| > 0$. Since we have already passed the last successful iteration, this implies that the incumbent solution is a stationary point, and therefore that $\|\nabla f(x_k)\| = 0$.

∎

The following development will focus on the case when the number of successful iterations is infinite. We proceed to prove a series of statements in order to show convergence of the algorithm, results that typically appear in the context of trust region-based DFO algorithms (Conn et al. 2009).

Lemma 3.10. *If we enforce* $\sum_k \alpha_k < \infty$,

$$\lim_{k\to\infty} \Delta_k = 0.$$

Proof. Let the set of successful iterations be denoted by $\mathcal{S}$. For $k \in S$,

$$\widehat{f}(x_k) - \widehat{f}(x_{k+1}) \geq \eta_0(m(x_k) - m(x_k + s_k)). \tag{3.50}$$

For each such iteration,

$$P\Big(f(x_k) - f(x_{k+1}) < \eta_0(m(x_k) - m(x_k + s_k))\Big) \leq \alpha_k. \tag{3.51}$$

Since $\sum_k \alpha_k < \infty$ holds for $k \in \mathcal{S}$,

$$P\left(\lim_{k\to\infty} f(x_k) - f(x_{k+1}) < \eta_0(m(x_k) - m(x_k + s_k))\right) = 0. \tag{3.52}$$

due to the first Borel-Cantelli lemma which says that if for an event A_k, $\sum_k P(A_k) < \infty$, then the probability of A_k happening infinitely often is 0. This can be rewritten as

$$P\left(\lim_{k\to\infty} f(x_k) - f(x_{k+1}) \geq \eta_0(m(x_k) - m(x_k + s_k))\right) = 1. \tag{3.53}$$

Now, using Lemma 3.2,

$$P\left(\lim_{k\to\infty} f(x_k) - f(x_{k+1}) \geq \right. \tag{3.54}$$

$$\left. \eta_0 \left[\frac{\kappa_d}{2} \|\nabla m_k(x_k)\| \min\left\{\frac{\|\nabla m_k(x_k)\|}{\kappa_H}, \Delta_k\right\}\right]\right) = 1. \tag{3.55}$$

Since Algorithm 3.1 ensures that $\|\nabla m_k(x_k)\| \geq \min\{\epsilon_c, \mu^{-1}\Delta_k\}$,

$$P\left(\lim_{k\to\infty} f(x_k) - f(x_{k+1}) \geq \right. \tag{3.56}$$

$$\left. \eta_0 \left[\frac{\kappa_d}{2}\min\{\epsilon_c, \mu^{-1}\Delta_k\}\min\left\{\frac{\min\{\epsilon_c, \mu^{-1}\Delta_k\}}{\kappa_H}, \Delta_k\right\}\right]\right) = 1. \tag{3.57}$$

If the right-hand side does not converge to zero, we can achieve infinite decrease. Since we know that f is bounded from below, we cannot achieve this decrease, implying that $\lim_{k\in\mathcal{S}} \Delta_k = 0$. For the iterations $k \notin \mathcal{S}$, we know from Algorithms 3.1 and 3.2 that $\Delta_k \leq \lambda_1 \Delta_{s_k}$, where s_k is the index of the last successful iteration before k. Since $\Delta_{s_k} \to 0$, then $\Delta_k \to 0$ for $k \notin \mathcal{S}$. ∎

Lemma 3.11.

$$\liminf_{k\to\infty} \|\nabla m_k(x_k)\| = 0.$$

Proof. From Algorithm 3.1, $\Delta_k \geq \min\{\zeta \|\nabla m_k(x_k)\|, \Delta_{\text{inc}}\}$. If $\|\nabla m_k(x_k)\| > \kappa_1 > 0$ for all k, then $\Delta_k > 0$ (from Lemma 3.7). But this is a contradiction, as we proved in Lemma 3.10 that $\lim_{k\to\infty} \Delta_k = 0$. ∎

Lemma 3.12. *For any subsequence $\{k_j\}$ such that*

$$\lim_{j\to\infty} \left\|\nabla m_k(x_{k_j})\right\| = 0, \tag{3.58}$$

it also holds that

$$\lim_{j\to\infty} \left\|\nabla f(x_{k_j})\right\| = 0.$$

Proof. From Equation 3.58, $\left\|\nabla m_{k_j}(x_{k_j})\right\| \leq \epsilon_c$ for j sufficiently large. The criticality step in Algorithm 3.1 ensures that the model is fully linear on $\mathcal{B}(x_{k_j}, \Delta_{k_j})$, where $\Delta_{k_j} \leq \mu \left\|\nabla m_{k_j}(x_{k_j})\right\|$ for j sufficiently large, and $\nabla f(x_{k_j}) \neq 0$. From Equation 3.22,

$$\begin{aligned}\left\|\nabla f(x_{k_j}) - \nabla m_{k_j}(x_{k_j})\right\| \leq & \frac{\sqrt{d}\Lambda_{Y_d}}{\Delta_k}\left(|\widehat{f}(x_k + y_p) - f(x_k + y_p)|\right. \\ & \left. + |\widehat{f}(x_k) - f(x_k)|\right) + \frac{5}{2}\Lambda_{Y_d}\sqrt{d}\Delta_k(\gamma_m + \gamma_f).\end{aligned} \tag{3.59}$$

From Lemmas 3.5 and 3.10, which apply to the right-hand side of Equation 3.59, we get

$$\lim_{j\to\infty} \left\| \nabla f(x_{k_j}) - \nabla m_{k_j}(x_{k_j}) \right\| = 0. \tag{3.60}$$

We also have the triangle inequality which states

$$\left\| \nabla f(x_{k_j}) \right\| \leq \left\| \nabla f(x_{k_j}) - \nabla m_{k_j}(x_{k_j}) \right\| + \left\| \nabla m_{k_j}(x_{k_j}) \right\|. \tag{3.61}$$

From Lemma 3.11 and Equation 3.60, we get the desired result. ∎

Lemmas 3.11 and 3.12 immediately give

Theorem 3.1.

$$\liminf_{k\to\infty} \nabla f(x_k) = 0.$$

3.5 CONCLUSIONS

In this work, we have extended the theory regarding fully linear models to the case when there is stochastic noise associated with the black-box evaluations. We handle this noise through careful trust region management and sampling schemes. Though we maintain full linearity regularly in terms of model point geometry, the models become closer and closer to the deterministic definition of full linearity in terms of the discrepancy between true function value and estimated value as the algorithm progresses and the sample replications increase. In this way we simultaneously build a true fully linear model while converging to a stationary point. This helps with significantly reducing simulation effort in the initial iterations of the algorithm.

In summary, the algorithm we propose does not require gradient estimates, makes few assumptions on the underlying distributions, and deals with expensive function evaluations by judicious sampling (both in terms of model-building as well as replications), all while providing a guarantee of global convergence to a stationary point.

4

SO-LVIT: SIMULATION OPTIMIZATION—LEARNING VIA TRUST REGIONS

4.1 INTRODUCTION

In this chapter, we propose extensions and enhancements to individual algorithmic elements presented in Chapter 3, and justify each of these choices. In particular, we discuss the choice of the surrogate model that we use, the manner of sampling and construction of this model, the estimation of underlying variance, stopping criteria, and the effect of globally optimizing the trust region subproblem. Through this, we outline the development of the theory into a practical implementation, which we call SO-LVIT, or **S**imulation **O**ptimization—**L**earning **Vi**a **T**rust regions.

We follow this discussion with comparative testing of the algorithm on a large test bed against other available implementations. We first compile a 500-problem test set to assess the performance of the implementation within a pre-specified computational budget. With these promising results, we experiment with different applications from chemical engineering to demonstrate the applicability of such techniques in practical settings.

The first example we investigate relates to a classical problem in the supply chain and operations literature, which has to do with inventory optimization. The system of interest is modeled using a discrete-event simulation that incorporates uncertain customer demands, lead times, and other system dynamics.

The second example involves the optimization of the design of a DNA separation device. The movement of a DNA strand under electrophoretic forces has been modeled using Brownian dynamic simulations, where the progress of the DNA strand may be impeded using obstructions in the device. The sizing of these obstructions

affects the movement of different lengths of DNA differently, and therefore, the problem is posed as finding the optimal obstruction size in order to maximize the separation between two pre-specified lengths of DNA.

We then summarize the results of the chapter, and discuss future directions for implementations and comparisons, potential new applications, and extensions to current applications in simulation optimization for engineers.

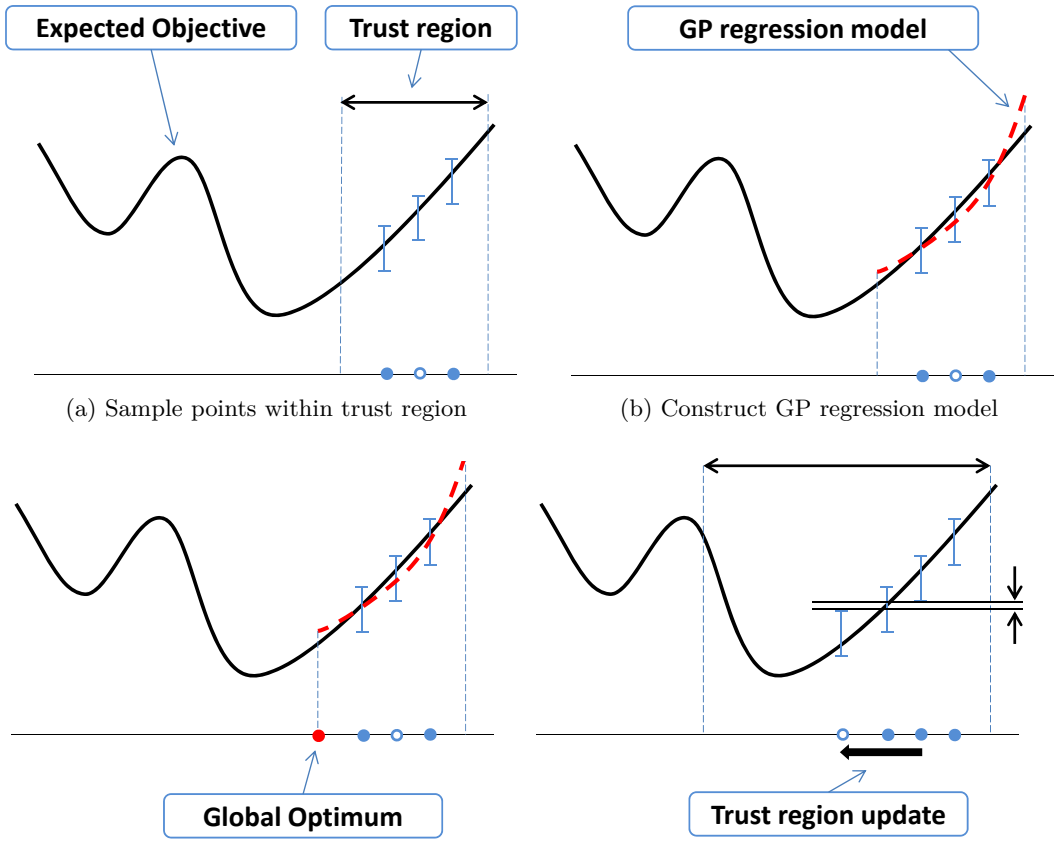
4.2 IMPLEMENTATION DETAILS

Figure 4.1 shows a 1-D schematic of an iteration of the SO-LVIT algorithm. Figure 4.1a illustrates the expected value of the true underlying function in solid black; the portion of the variable space that is within the trust region radius; a confidence interval on each of the sample points, one of which corresponds to the trust region center. Section 4.2.1 provides implementation details on the selection of these points for model-building.

Figure 4.1b illustrates the GP regression model that fits these samples using a dashed red line, and the schematic only shows this regression function within the trust region, to show that we are only concerned about its predictive performance within this region. The implementation of the GP regression scheme is described in Section 4.2.2.

Figure 4.1c illustrates the globally optimal solution to the trust region subproblem in red, which is the point that minimizes the GP regression model within the trust region radius. In general, it is very difficult to find the global optimum of the regression function, but it may be worth it if the underlying simulation is expensive to evaluate. Sections 4.3.2 and 4.4.2 provide more details on the benefits of using global search for the TRSP.

Figure 4.1d illustrates that the black-box simulation has been sampled at this global optimum, and is a candidate for being the next trust region center. The figure illustrates that the upper confidence level for the candidate point is still below the sample mean of the incumbent, indicating a high probability that the candidate point may indeed be a better point. This comparison is formally performed between the sample means, and if the null hypothesis (i.e., incumbent is better than candidate) is rejected at the α_k-significance level, and the trust region center and radius may



(a) Sample points within trust region

(b) Construct GP regression model

(c) Globally optimize trust-region subproblem

(d) Update trust region center and radius

Figure 4.1: The four key steps in an iteration of the SO-LVIT algorithm are shown above. The black surface is the underlying expectation function of the simulation.

be updated as shown in the figure. The management of the α_k-significance level is provided in Section 4.2.3 and Algorithm 3.1.

To quote Fu et al. (2005), some of the key issues in an implementation of an SO algorithm are:

“

1. neighborhood definition;
2. mechanism for exploring/sampling (search), especially how previously generated (sampled) solutions are incorporated;
3. determining which candidate solution(s) to declare the best (or “good”); statistical statements?

4. the computational burden of each function estimate (obtained through simulation replications) relative to search (the optimization algorithm). ”

We have addressed each of these issues in SO-LVIT. The neighborhood definition is carefully controlled by a trust-region framework. The mechanism for exploring/sampling is done through the trust region subproblem, while previously sampled points may be used to build our surrogate model at each iteration. Hypothesis testing along with careful control of the α-significance level are used to provide 'good' statistical statements. Finally, we spend effort on simulation replications purely when necessary in order to ensure descent with high probability.

4.2.1 *Selection of points and maintenance of geometry for interpolation and regression models*

Algorithms 4.1 and 4.2 outline the procedure we use to choose points for our interpolation and regression models. To represent the underlying function accurately we need to consider the number of sample replications as well as the geometry of selected points (Lemma 3.4). Algorithm 4.1 focuses on the sample replication aspect, while Algorithm 4.2 on the geometric aspect.

Algorithm 4.2 provides a sequential procedure to select points for interpolation/regression. The construction of a fully linear model involves satisfying the geometric condition imposed by Lemma 3.3. Given a set of points that have already been selected for model-building (the trust region center is always included), we find a basis to the nullspace of the space spanned by this set. Then, we cycle through a sequence of given candidate (previously sampled) points, and project them onto this nullspace. If this projected point is far enough away from the space spanned by the set of included points (via the ψ-thresholding), we include the candidate point in our model. We continue until we get $d+1$ points, which are now sufficiently affinely independent. If we are unable to find $d+1$ such points, the columns that form the basis of the nullspace at the end of the procedure, scaled by the trust region radius, provide potential points that may be chosen for sampling and to be included in the model. This is a fairly standard procedure for DFO algorithms, and it is easy to see that this procedure requires a finite number of steps.

Algorithm 4.1 provides the sequence of candidate points to Algorithm 4.2. By sorting candidate points within an enlarged trust region by the number of sample

replications, candidate points with more replications are considered first. Though this is not guaranteed to give you a set of sufficiently affinely independent points with the highest number of cumulative replications, it provides an effective way to include points which have been sampled more,

It is natural to want points with higher replications, as the sample mean at each would be closer to the true value with high probability, and it requires fewer further replications at each of these points to satisfy the sampling criteria as in Step 4 of the algorithm (from Lemma 3.5).

The notation used in the two algorithms is from Table 3.1.

4.2.2 *GP regression*

We choose to use Gaussian Process (GP) regression for multiple reasons. When trust regions are larger, we only require χ_1 (a small number of) replications. As sample mean estimates may not be very accurate, we choose to use regression in this case. However, when the trust regions are smaller (below Δ_c), we require a larger number of replications, and perform interpolation using GPs.

GP regression allows the flexibility of incorporating many points in the regression (as opposed to a fixed number of points, or points on a fixed grid in space as used in the response surface literature). This allows the use of previously sampled points very effectively. GPs are also attractive as they automatically provide a trade-off between model accuracy and model complexity, without having to specify an explicit parameter to control this (Rasmussen & Williams 2006). GP regression is also capable of handling discrete inputs—potentially a major generalization compared to other response surface methods.

In addition, the multiple simulation replications we perform at each sample site allow us to build, as described in Section 4.3.1, an estimate for the variance in the stochasticity of the underlying response surface, and this estimate can be easily incorporated within the GP framework. A concern that sometimes arises with GP regression relates to its expense, but we assume that the time it takes to perform is insignificant in the case of expensive stochastic simulations.

Aside from the above features, as mentioned in Chapter 3, GP regression techniques that incorporate radial basis function kernels have been shown to fall into the fully linear framework, and we make use of this.

Algorithm 4.1 Construction of interpolation/regression model

Step 1: Collect previously sampled points within the enlarged trust region with radius $\psi_0\Delta_k$, where $\psi_0 \geq 1$. Form the set $\Pi = \{y_1 = 0, y_2, \ldots, y_{|\Pi|}\}$, where y_p is a displacement from the trust region center x_k.

Step 2: Sort points by number of sample replications, from largest to smallest. If more than $d+1$ of these points have been sampled (1) more than n_1 times if $\Delta_k < \Delta_c$, or (2) greater than or equal to χ_1 times otherwise, then move points within main trust region Δ_k to the front of the list.

Step 3: Call Algorithm 4.2 in model-building mode with the set of candidate points Π as an argument, and retrieve a set of interpolation points Y_k.

Step 4:
if $\Delta_k \leq \Delta_c$ **then**
  if $i = 0$ **then**
    Sample such that the number of replications at each of the chosen points in Y_k is $n_0 = \lceil \chi_0 \lambda_0^{-4} \rceil + 1$, as recommended in Section 3.4, Lemma 3.5.
  else
    Sample such that the number of replications at each of the chosen points in Y_k is $n_i = \lceil n_{i-1} \lambda_0^{-4} \rceil + 1$, as recommended in Section 3.4, Lemma 3.5 (only if n_i has not already been defined).
  end if
  Add up to $\pi_{\max} - (d+1)$ from the remaining points in Π to Y_k, if they have been sampled at at least $\psi_2 n_i$ times, where $0 < \psi_2 < 1$.

  Build model m_k that interpolates all points in Y_k using the GP interpolation scheme described in Section 3.2.2.
else
  Add up to $\pi_{\max} - (d+1)$ from the remaining points in Π to Y_k, if they have been sampled at at least χ_1 times.

  Build a regression model m_k over the points in Y_k using the GP regression scheme described in Section 3.2.2.
end if

Algorithm 4.2 Model-building and Model-improving algorithm

Step 1:
if Algorithm was called in model-building mode **then**
Initialize the set of points chosen for interpolation/regression, $Y_k = \{y_1 = 0\}$.
Initialize $Z = I_d$.
else if algorithm called in model-improvement mode **then**
Initialize Z as a basis of the nullspace of the given set of points Y_k. Denote $\Pi = Y_k$.
end if

Step 2: Initialize $p = 1$.
while $y_p \in \Pi$ **do**
if $||ZZ^T(1/(\Delta_k\psi_0))y_p|| \geq \psi_1$, i.e., the projection of the candidate point onto the nullspace of Y_k is greater than ψ_1, **then**
Include y_p in set of interpolation points Y_k.
Update Z to be an orthonormal basis for the nullspace of Y_k. Update $p \leftarrow p + 1$.
end if
end while

Step 3
If called in model-building mode and $|\Pi| < d + 1$, evaluate $f(x_k + z_p)$ for all columns of Z χ_1 times, and add these points to Y_k.
if $|Y_k| < d + 1$ deem model as not fully linear. **then**
If called in model-improvement mode, use the first column of Z to be the model-improving direction, and evaluate it n_i times if $\Delta_k < \Delta_c$, or χ_0 times otherwise.
else
If $|Y_k| = d + 1$, deem model as fully linear.
end if

Though there are readily available packages for GP regression (The Gaussian Processes Website 2014), getting them to work in practice often requires a lot of experience. The choice of hyperparameters and their optimization, the method of inference, and the choice of covariance (basis) functions can be very important. A bad choice of these can lead to a completely flat regression surface with a few spikes at the data points. As a consequence, poor models affect the progress of the trust region algorithm.

Leave-one-out likelihood The most common way to optimize the hyperparameters for GP regression is through maximizing the log-marginal likelihood (Equation 3.3). Rather than using this, we use the recommendation of Sundararajan & Keerthi (2001) and find that maximizing the leave-one-out (LOO) likelihood provides better results in terms of reducing overfitting. This LOO-likelihood is computed by maximizing the sum of log-likelihoods, where each log-likelihood is constructed by leaving one of the training data (sample) points out.

Model selection and starting points We also find that choosing an appropriate covariance function and relevant hyperparameters can aid in model fit. As noted in Chapter 3, there are many possible basis functions that may be used that fall into the fully linear framework. At the outset, we cycle through a set of covariance or basis functions and optimize the LOO-likelihood for each to find the best hyperparameters. The corresponding covariance function is used along with the recommended hyperparameter values. Rasmussen & Nickisch (2011) provides a BFGS routine to optimize this likelihood function, and we use several rounds of starting points in the search for the best hyperparameters. We then retain the best found basis function as well as hyperparameters as initial starting points for subsequent iterations of the algorithm.

4.2.3 *Hypothesis test significance-level*

In order to satisfy the requirement that $\sum_{k=1}^{\infty} \alpha_k < \infty$ (from Lemma 3.10), we apply the rule that

$$\alpha_k = a \cdot (b)^k, \tag{4.1}$$

where k is the kth trust region center. We also require $0 < a \leq 0.5$, $0 < b < 1$, so that α is always less than 0.5, and that the series ratio test is always satisfied.

4.3 ALGORITHMIC ENHANCEMENTS

4.3.1 *Variance learning*

As outlined in Chapter 3, the algorithmic framework we use depends on simulation replications at every point. The algorithmic scheme made use of replications not only to regress over averaged values, but in hypothesis tests to determine when to move the trust region center.

Here we show that these replications can, in addition, be used to make GP regression even more accurate. In particular, we first show that these replications can be used to build a consistent estimator of the underlying variance in the case of uniform Gaussian noise, one that will grow ever more accurate with the progress of the algorithm. This estimator can then be used to provide good bounds on the parameter that represents the intrinsic noise in the Gaussian process regression model, ω, as defined in Section 3.2.2. For this, we need at least two samples at every data point.

Lemma 4.1. *If $F(x_i, \omega) \sim \mathcal{N}(f(x_i), \sigma^2)$, an unbiased estimator for σ^2 is*

$$\widehat{\sigma}^2 = \frac{\sum_{k \in K} \widehat{\sigma}_k^2}{|K|},$$

where $k \in K$ means that we have sampled at some points with k replications, n_k is the number of such points, and

$$\widehat{\sigma}_k^2 = \sum_{i=1}^{n_k} \sum_{j=1}^{k} \frac{(F_{ij} - \overline{F}_i)^2}{(k-1)n_k}.$$

Proof. We begin by introducing an estimator that uses all points that have been sampled k times, and the size of this set is n_k,

$$\widehat{\sigma}_k^2 = \sum_{i=1}^{n_k} \sum_{j=1}^{k} \frac{(F_{ij} - \overline{F}_i)^2}{kn_k}. \tag{4.2}$$

This estimator makes intuitive sense, as the numerator represents a squared deviation from some mean, and this deviation is averaged out over the kn_k total data points. Next, we investigate the consistency of this estimator. We assign $X_i = \sum_{j=1}^{k} \frac{(F_{ij}-\overline{F}_i)^2}{k}$. So, $\widehat{\sigma}_k^2 = \frac{1}{n_k}\sum_{i=1}^{n_k} X_i$, and we know that this estimator converges in probability to

$$\frac{1}{n_k}\sum_{i=1}^{n_k} X_i \xrightarrow{P} E(X_i) \tag{4.3}$$

from the law of large numbers. Now,

$$\begin{aligned} E(X_i) &= E\left(\sum_{j=1}^{k}\frac{(F_{ij}-\overline{F}_i)^2}{k}\right) = E\left(\sum_{j=1}^{k}\frac{F_{ij}^2+\overline{F}_i^2-2F_{ij}\overline{F}_i}{k}\right) \\ &= E\left(\sum_{j=1}^{k}\frac{F_{ij}^2}{k}-\overline{F}_i^2\right) = E\left(\frac{\sum_{j=1}^{k}F_{ij}^2}{k}\right) - E(\overline{F}_i^2) \\ &= \frac{\sum_{j=1}E(F_{ij}^2)}{k} - E(\overline{F}_i^2). \end{aligned} \tag{4.4}$$

From the identity $E(X^2) = V(X) + E^2(X)$, this becomes

$$\begin{aligned} E(X_i) &= \frac{1}{k}\sum_{j=1}^{k}(V(F_{ij})+E^2(F_{ij})) - V(\overline{F}_i) - E^2(\overline{F}_i) \\ &= \sigma^2 + E^2(F_{ij}) - V\left(\frac{\sum_{j=1}^{k}F_{ij}}{k}\right) - E^2\left(\frac{\sum_{j=1}^{k}F_{ij}}{k}\right), \end{aligned} \tag{4.5}$$

which for independent random samples is

$$\begin{aligned} E(X_i) &= \sigma^2 + E^2(F_{ij}) - \frac{1}{k^2}\sum_{i=1}^{k}V(F_{ij}) - \left(E\left(\frac{\sum_{j=1}^{k}F_{ij}}{k}\right)\right)^2 \\ &= \sigma^2 + E^2(F_{ij}) - \frac{\sigma^2}{k} - E^2(F_{ij}) \\ &= \frac{k-1}{k}\sigma^2. \end{aligned} \tag{4.6}$$

This shows that the estimator we considered is biased, but is easily fixed by multiplying through by $\frac{k}{k-1}$. Now we compute the corresponding metric for all such $k \in K$, and use the sample average

$$\widehat{\sigma}^2 = \frac{\sum_{k \in K} \widehat{\sigma}_k^2}{|K|}, \tag{4.7}$$

which is clearly an unbiased, consistent estimator of σ^2. ∎

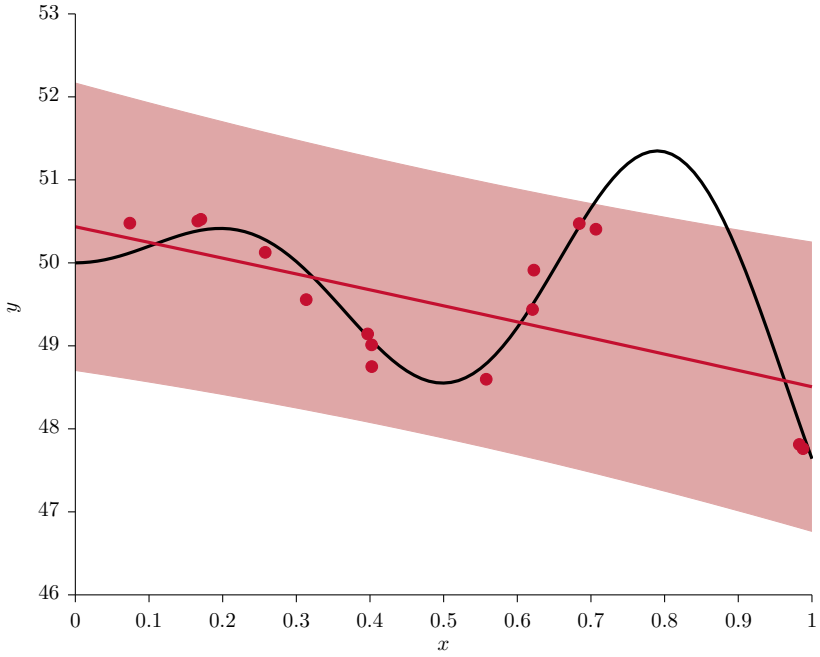
Once we have an estimator for the variance, we can use confidence bounds on the estimator as bounds on the intrinsic variance hyperparameter during the hyperparameter estimation step. This bounding of the variance can help to improve the quality of the regression significantly. We illustrate this in Figure 4.2, where the black line is the true underlying function. The red points are the data, the red line is the mean of the regression, and the pink band includes the fitted variance of the Gaussian Process as well as the uncertainty in function value at points away from data points. Figure 4.2a clearly underfits the data, whereas Figure 4.2b seems to fit the data much better.

The restriction of the variance helps help focus the multi-start optimization of the likelihood to a more realistic range of values. In some cases, this just helps in finding a better local solution to the problem, and, in other cases, it cuts off a global solution to the maximum likelihood problem that corresponds to an underfit due to some artifact of the data.

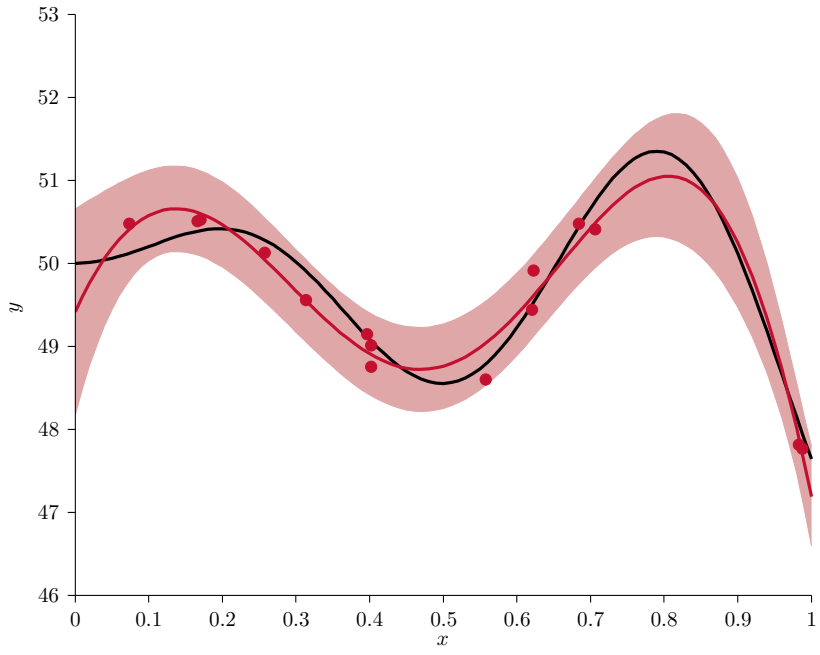
We can use these confidence intervals to bound the variance in the GP regression directly. In addition, even if the nature of the underlying stochasticity is not uniform Gaussian, we can make use of this estimate using points that are only within the current trust region, assuming that the variance does not change significantly in this region.

4.3.2 *Global optimization of TRSP*

Lemma 3.2 in Chapter 3 involves a backtracking line search scheme to achieve a sufficient decrease condition when solving the trust region subproblem (TRSP). In other words, this decrease suffices to show global convergence to stationary points. Traditional trust region methods for nonlinear programming focus on ways to update the solution of the TRSP by performing fast linear algebra. In our context, however,



(a) Unbounded intrinsic variance; underfit to data



(b) Bounded intrinsic variance; good fit to data

Figure 4.2: Two different regression surfaces for twenty samples taken from the function $f(x) = 50 + \frac{x \sin x}{4} - \frac{x^2}{100}$

function evaluations are expensive, and we assume that any time taken algorithmic computations are insignificant in comparison.

In this situation we hypothesize that solving the TRSP to optimality, or even global optimality, may be worth it, especially if the surrogate models we construct are of good quality. The form of the GP interpolation functions we use are expressed by the difference of sums of functions, where the functions are in general nonconvex (log-concave in the case of a Gaussian RBF). Therefore, in order to find a global solution to such a surface, we make use of the BARON software (Tawarmalani & Sahinidis 2005; Sahinidis 2013) which performs deterministic global optimization through a spatial branch-and-bound procedure. We omit the details of the implementation, but show that we gain some benefit as compared with a local search, as illustrated in Section 4.4.

4.3.3 *Stopping criteria*

In the fields of derivative-free optimization and simulation optimization, expensive function evaluations are often the bottleneck for algorithmic iterations. This necessitates the enforcement of a computational budget for most practical applications. This is common as convergence rates are slower than derivative-based methods, and so it reasonable to run the algorithm until the computational budget is exhausted.

Trust region methods incorporating fully linear models provide a natural stopping criterion in the derivative-free optimization case. If, during the search procedure, the algorithm enters the criticality step, and stays in the criticality step until $\Delta_k \leq \frac{\epsilon_c}{\kappa_{eg}}$, then from the triangle inequality,

$$||\nabla f(x_k)|| \leq ||\nabla m_k(x_k)|| + ||\nabla f(x_k) - \nabla m_k(x_k)|| \leq \epsilon_c + \kappa_{eg}\frac{\epsilon_c}{\kappa_{eg}} = 2\epsilon_c, \quad (4.8)$$

and this can be used as a stopping criterion.

Similarly, for simulation optimization, we look at the case when

$$\Delta_k^{(j)} := \tau_c^{j-1}\Delta_{\text{inc}} \leq \frac{2\epsilon_c}{5\Lambda_{Y_d}\sqrt{d}(\gamma_f + \gamma_m)}, \quad (4.9)$$

in accordance with Lemma 3.4. We see that at this stage in the criticality step algorithm that

$$\begin{aligned} ||\nabla f(x_k)|| &\leq ||\nabla m_k^{(j)}(x_k)|| + ||\nabla f(x_k) - \nabla m_k^{(j)}(x_k)|| \\ &\leq 2\epsilon_c + \frac{\sqrt{d}\Lambda_{Y_d}}{\Delta_k^{(j)}}(|\widehat{f}^{(j)}(x_k + y_p) - f(x_k + y_p)| + |\widehat{f}^{(j)}(x_k) - f(x_k)|), \end{aligned} \tag{4.10}$$

and since, with high probability, $|\widehat{f}(x_i + y_p) - f(x_i + y_p)| \leq (\Delta_k^{(j)})^2$, we also have

$$\begin{aligned} ||\nabla f(x_k)|| &\leq 2\epsilon_c + 2\sqrt{d}\Lambda_{Y_d}\Delta_k^{(j)} \\ &\leq 2\epsilon_c + 2\sqrt{d}\Lambda_{Y_d}\frac{2\epsilon_c}{5\Lambda_{Y_d}\sqrt{d}(\gamma_f + \gamma_m)} \\ &= 2\epsilon_c + \frac{4\epsilon_c}{5(\gamma_f + \gamma_m)}, \end{aligned} \tag{4.11}$$

with high probability. From Lemma 3.5, this probability is

$$P\left(|\widehat{f}(x_i + y) - f(x_i + y)| \leq \Delta_i^2\right) \geq 1 - \frac{\sup_{x \in \mathcal{B}(x_k, \Delta_k^{(j)})} \sigma^2(x)}{l_j(\Delta_k^{(j)})^4}. \tag{4.12}$$

We can approximate $\sup_{x \in \mathcal{B}(x_k, \Delta_k^{(j)})} \sigma^2(x)$ with the variance estimate in Lemma 4.1. The algorithm would then be terminated when the trust region radius falls below the threshold in Equation 4.9, and the probability from Equation 4.12 is more than some high value (say, 0.99).

In practice, one may also simply terminate when more than some pre-defined number of criticality step iterations have taken place (i.e., j is greater than some value).

4.4 RESULTS AND APPLICATIONS

4.4.1 *Comparisons on large test sets*

To benchmark the performance of our algorithm, we selected six different algorithms from the literature to compare against, and developed a problem test set derived from the one used in Rios & Sahinidis (2013). The six algorithms that we bench-

marked against are listed in Table 4.1. The algorithms we chose to compare include those that use response surfaces (as we relied on these as well), local search methods such as Nelder-Mead simplex procedures (as our method has a local scope as well), and those that use trust regions.

Table 4.1: List of algorithms compared against

Algorithm	Type	Citation
SKO	Global response surface	Huang et al. (2006)
SNM	Direct search (Nelder-Mead)	Chang (2012)
SNOBFIT	Multi-start local response surface	Huyer & Neumaier (2008)
SPSA	Stochastic Approximation	Spall (2003a)
SPSA 2^{nd} Order	Stochastic Approximation	Spall (2003a)
STRONG	Local response surface	Chang et al. (2011)

Both versions of the SPSA (Simultaneous Perturbation Stochastic Approximation) algorithm follow a stochastic approximation scheme. The idea behind SPSA-Basic is to gain some measure of the gradient by using just two function evaluations, as opposed to the more common $n+1$ function evaluations. This is done by simultaneously perturbing a point in all dimensions, as opposed to one dimension at a time. The estimate now becomes independent of the dimension of the problem. SPSA-2nd Order also uses some estimate of second-order information in this process.

SNOBFIT (Stable Noisy Optimization by Branch and Fit) uses a multi-start method, where it builds quadratic regression models in an iterative trust-region framework. SKO (Sequential Kriging Optimization) is a modification of the EGO algorithm Jones et al. (1998) for the stochastic case, where a global response surface is built via a Kriging model over an initial design of experiments, and subsequent samples are chosen via the maximization of a measure of expected improvement. SNM (Stochastic Nelder-Mead) is a version of the Nelder-Mead algorithm that uses an effective sample size scheme to control noise. STRONG (Stochastic Trust Region Response-Surface Method) is a trust-region based algorithm that also builds linear and quadratic models and uses replications to characterize output uncertainty.

The test set that was used involved 502 deterministic black-box real-valued functions (from Rios & Sahinidis (2013)), whose output value was perturbed by

a predetermined level of random Gaussian noise to represent stochasticity in the output. The types of problems in the test set are listed in Figure 4.3.

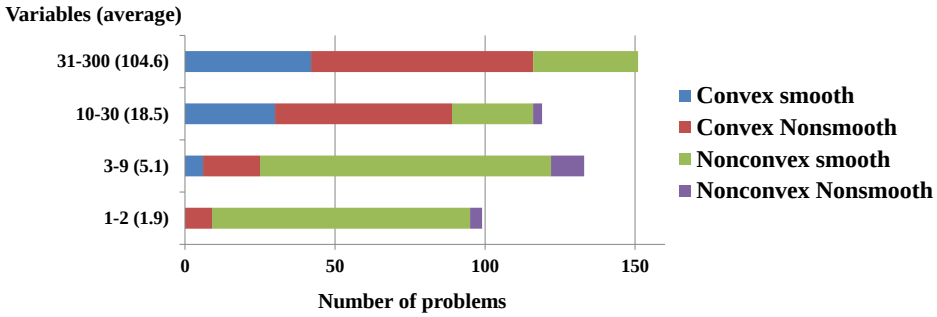


Figure 4.3: A breakdown of problem types in the 502-problem test set

To add noise to the deterministic results, ten random points were sampled from each function, and their average, A, was taken. For each sample, independent noisy observations were obtained by sampling from $\epsilon \sim \mathcal{N}(0, \sigma^2)$, with $\sigma = 0.1$, and this was multiplied by the average A, and added to the deterministic output at the sampled points. The idea behind this was to add a level of noise on the order of the range of variation in function value.

The reason that we use underlying deterministic functions is so that it gives us a basis for comparison. The solvers considered query the simulation at a number of points, and record and use the output values from the simulation. All of the solvers don't necessarily report the point that they believe is the 'best' point (or the incumbent solution). Since we know the underlying functional form, we are able to evaluate the black boxes to determine the true function value. We give the benefit of doubt to the solvers, and assume the true best point that was evaluated is also recorded as the incumbent solution by the solvers.

The algorithms mentioned in Table 4.1 and our algorithm, SO-LVIT, were run on the 502-problem test set. As we are interested in solving problems where simulation replications are expensive to obtain, a limit of 300 iterations (i.e., simulations) was fixed for this study. The purpose is to assess the performance of the algorithms under this strict budget of simulations.

The results we obtained are summarized in Figure 4.4. The vertical axis denotes the relative fraction of problems solved. By this, we mean that an algorithm is said to have 'solved' a particular problem if, within a certain tolerance, it is the best-performing algorithm on that problem. Two algorithms are said to have solved a

problem if they both find objective values that are within a pre-specified tolerance of each other. The tolerance used for this study is 10^{-3} of the best solution found.

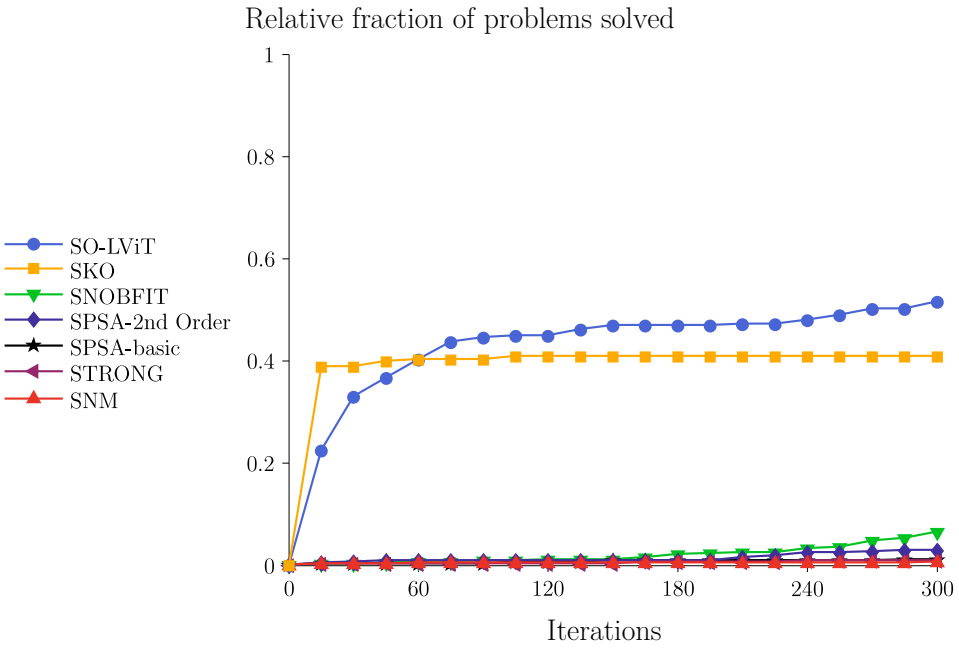


Figure 4.4: Relative fraction of problems solved vs. Number of simulations for seven solvers

In Figure 4.4, each point represents the number of simulations needed by a solver to find a solution that is better than any other point found by any other solver over the entire simulation budget for a certain fraction of problems. As we know that true underlying functions, we can actually judge whether one solution is better than the other. So, for example, the yellow square point in Figure 4.4 at $(120, 0.4)$ indicates that the solver SKO finds the best solution for 40% of the problems within a budget of 120 simulations. In general, graphs that are higher and more to the left are the ones that perform better on the test set.

From Figure 4.4, we see that SKO and SO-LVIT outperform the other algorithms. SKO is able to find good solutions early to a significant fraction of problems, but is not able to improve on the other problems with increased simulations. SO-LVIT is able to solve a larger fraction of problems at around the 60 iteration mark, and continues to find better solutions than other algorithms. This is not to say that the other solvers do not progress on a large fraction of the problems—but only indicates that they find the best solution only in a small number of problems.

We believe that the reason that SO-LVIT performs better is that it is able to build surrogate regression models with fewer function evaluations, is able to optimize these models to global optimality, ensures descent with high probability for each subsequent trust region, and handles noise in an effective manner through hypothesis tests.

The other surrogate-based algorithms—SKO, STRONG and SNOBFIT—each have their own shortcomings. SKO performs an initial design of experiments, which seems to help it find good solutions for many problems, but it seems unable to build on these and continue to refine solutions. STRONG, which is designed to refine solutions to local optimality, suffers as it attempts to build full quadratic models at every iteration below a certain threshold trust region radius. These require a number of points that is quadratic in the dimension of the problem, a number that grows rapidly with larger problem sizes. STRONG also does not effectively use previously sampled points. Though SNOBFIT refines solutions locally and has a global component to it in the form of multi-starts, it seems unable to negotiate highly noisy conditions in an effective manner during descent.

For applications, it is not clear how to compare the performance between two competing algorithms due to the noise. This is true especially for applications with expensive function evaluations. One of the reasons to do a comparative testing between algorithms on a large test set is to be able to pick the best one and then use it for applications. Thus, we use the SO-LVIT algorithm for the applications that follow.

4.4.2 *Global vs. local optimization for TRSP*

In Section 4.3.2, we proposed to solve the TRSP using global optimization, as we may be able to afford this when function evaluations are expensive. To test this hypothesis, we ran a deterministic version of the SO-LVIT algorithm on the 500-problem test set described in the previous subsection, and compared the performance when we use a backtracking line search versus the global optimizer BARON to recommend search directions at each iteration. Figure 4.5 shows the relative performance of the two when $\tau = 10^{-3}$ for a limit of 100 iterations. We see that the version using BARON outperforms the version using a backtracking line search by about 10%. Note that we enforce a 60 second time limit on BARON for the subproblems.

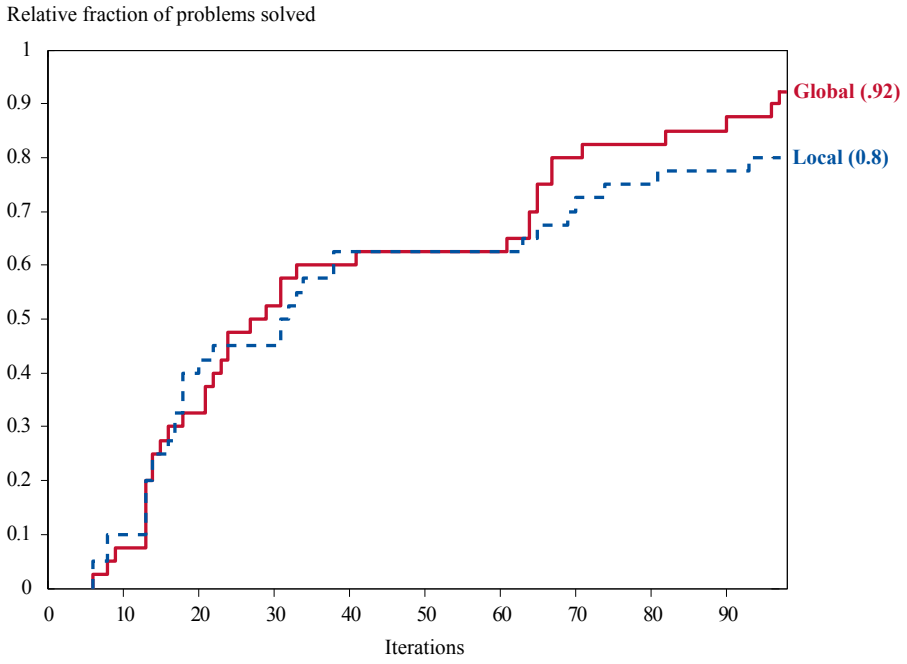


Figure 4.5: Impact of global and local optimization of the TRSP on overall performance

4.4.3 *Identification of reorder points in chemical supply chains*

Inventory planning is a fundamental problem in supply chain management (Cachon & Terwiesch 2008). In supply chain networks, it is commonly desired to keep inventory levels as low as possible for various reasons (holding costs, safety, product perishability, etc.). This minimization is traded off with some metric that incorporates demand satisfaction, such as customer satisfaction level of the event-based (alpha) or quantity-based (beta) type.

Many decisions are involved in the inventory planning process including location of inventory points in the network, sizing of inventory tanks/warehouses, and, depending on the inventory policy used, other parameters such as base stock levels and re-order points. Straightforward solutions may be derived for simple situations that may involve a single echelons with normally distributed uncertainties. The analysis of such networks, however, becomes increasingly complicated in networks with non-normally distributed and correlated uncertainties in demands, lead times and production times, multiple echelons, and other system-specific constraints. Chemical supply chains may pose additional differences from conventional supply chains,

such as inventory levels that may continuously change, or if the tanks are part of a complex process network structure in a chemical site.

As the inventory planning problem is a fairly general one, many approaches tackle this problem in the literature. A survey of the inventory literature is provided in Graves et al. (1992). However, few approaches focus on chemical supply chains, and fewer still are able to incorporate some or all of the features mentioned in the previous paragraph for performing inventory optimization. Further, it is very straightforward to incorporate rule-based inventory policies, complex probability distributions and non-trivial network structures within a discrete-event simulation framework. Due to this complexity, recommended policies from current optimization-based frameworks perform quite poorly when validated through such simulations.

Our motivation is to show that the simulation optimization techniques we have developed may help in optimizing directly over the simulations, especially when dealing with continuous parameters. We now introduce an example problem on which to test our algorithm. The example involves a multi-echelon system with one intermediate storage tank/warehouse and seven downstream warehouses. The task is to determine optimal re-order points for each of the warehouses, given the inventory policy being used. The network studied is shown in Figure 4.6.

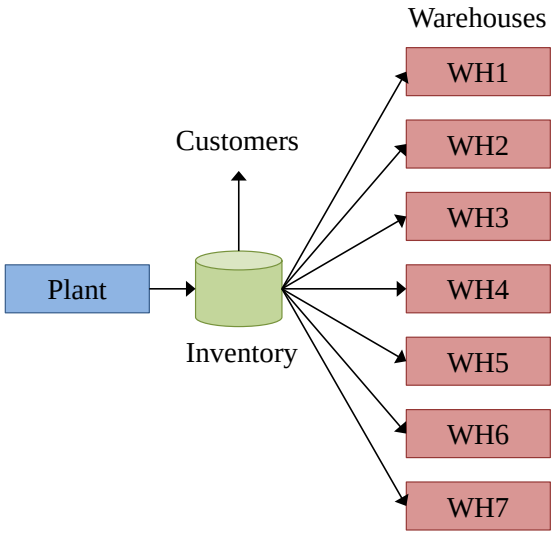


Figure 4.6: Supply chain network structure

This problem consists of 8 decision variables. However, the situation is not the same unconstrained case as investigated in Chapter 3. There are also constraints involved—for instance, we are required to satisfy a 95% quantity-based customer satisfaction level. The simulation we work with not only outputs an estimate of the profit, but also gives a measure of the customer satisfaction levels for each customer. To deal with this additional requirement, we use an exact penalty function to penalize our objective function.

Fig 4.7 shows the progress of the SO-LVIT algorithm on a 700-simulation budget limit for three different runs. We see that the solver is able to reduce the total inventory volume from the initial point by about 80% within this simulation budget.

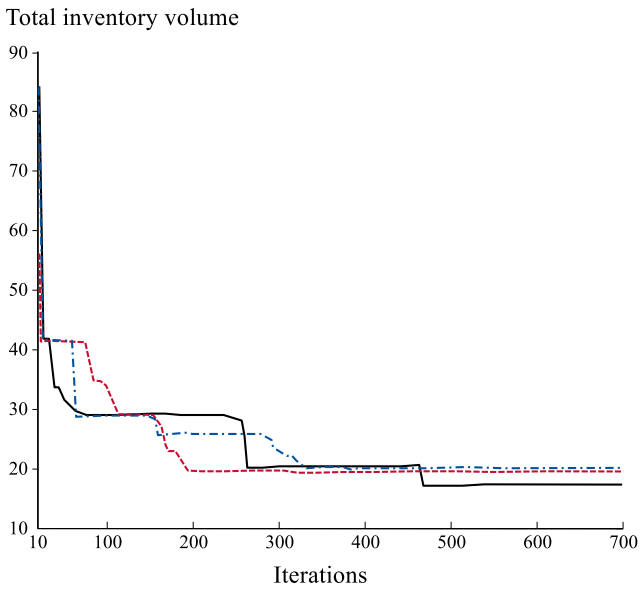


Figure 4.7: Total inventory volume vs. iteration count

4.4.4 *Optimum obstacle sizing in length-based DNA separation via post arrays*

Separation of different lengths of DNA strands is a fundamental problem in the biological and engineering sciences, and is an essential step in many types of DNA analyses, including DNA sequencing (Sanger et al. 1977; Albrecht et al. 2011) and mutation detection for cancer diagnosis (Wu et al. 1997; Albrecht et al. 2013).

One fast technique that attempts to resolve different lengths of DNA is the gel-free end-labeled free-solution electrophoresis separation (ELFSE) method (Ren et al.

1999; Meagher et al. 2008). Separation of DNA by electrophoresis is very common, though the electrophoretic mobility of DNA is length-independent. Electrophoretic mobility is the ratio of charge to friction, and since DNA strands have one charge per base pair, and since the friction on the strands also increases with length, this results in length-independent scaling. To break this length-independence, DNA strands are made to navigate through obstacle courses in order to deform them from their native randomly coiled state and introduce additional friction on each strand. Then, separation occurs by a combination of differing electrophoretic mobilities, and the different times taken to drape around, hook and slide off around the post obstacles. Thus, end-labeled DNA separation may be integrated with modern separation techniques that employ micro-fabricated obstacle courses, such as those comprising of nano-post arrays (Volkmuth & Austin 1992; Doyle et al. 2002; Ou et al. 2009). Such systems can offer significant speed up of separation, but are hard to tune. An illustration of a micro-fabricated device to achieve this can be found at `http://www.cchem.berkeley.edu/sjmgrp/people/nerayo/nerayo.htm`.

The phenomena described above may be simulated using Brownian dynamics simulations to gain insights for designing such devices (Kim & Doyle 2007; Patel & Shaqfeh 2003; Cho & Dorfman 2010; Olson et al. 2011), but they are usually very expensive to simulate (on the order of several hours). The simulation we use models polymers as sections that are lumped together and represented as a spring connected to beads. The Brownian motion of each bead follows from the stochastic collisions with bulk solvent water molecules. Realistic spring force laws and wall interactions have been modeled in the implementation that we work with (Fahrenkopf 2014). The solution of the model involves integrating the system of stochastic differential equations (SDEs) that describe the system using principles from Itō calculus. We avoid going into further details of the simulation.

The design problem is of sizing the post radius to optimally separate two specific lengths of DNA. As we have access to a stochastic simulation that models the phenomenon, we use the simulation optimization algorithm SO-LVIT developed in Chapter 3 and in this chapter to optimize this radius. The simulations model the movement of the DNA molecules around a single post. In practice, the DNA molecule could theoretically hook around hundreds of posts in its path. This repeated negotiation around obstacles further resolves the separation, as would multiple replications of the simulation.

The simulation output consists of the time difference (in centi-seconds) between the two strands crossing a certain point (i.e., a detector), and the individual times they take to elute.

Figure 4.8 illustrates the progress of the algorithm with the objective function. The optimization is transformed to a minimization problem in the figure. The problem illustrated here is the separation of two lengths of DNA, one that has about 23,400 base pairs, and another that has about 31,800 base pairs. We run the algorithm until we have run out of a pre-specified computational budget of 2,500 iterations.

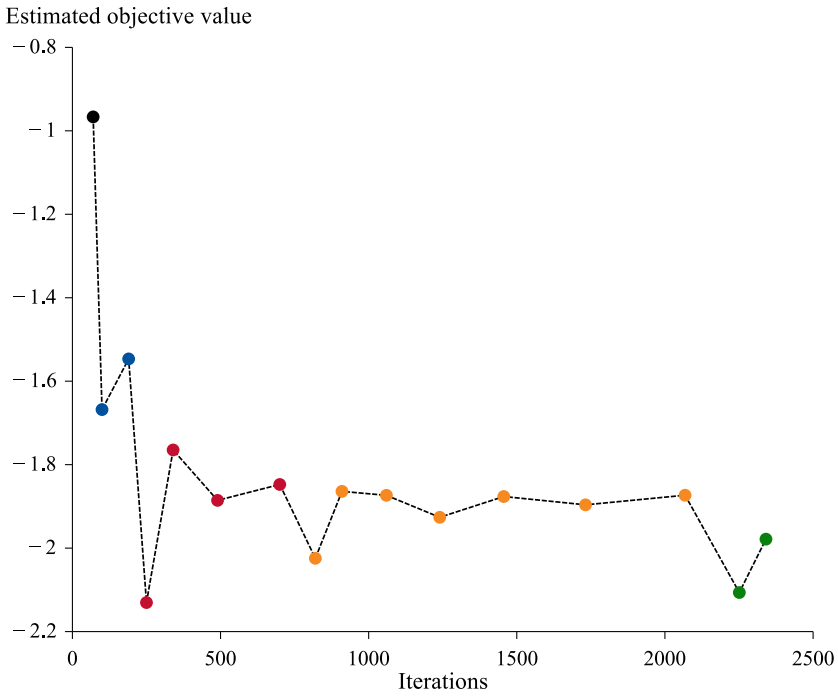


Figure 4.8: Mean difference in elution times vs. algorithm iterations

Each color in the plot corresponds to a different trust region center. Even though we are dealing with a one-dimensional problem, the noise levels are very high, and a large number of replications are required to tell one point from another. The estimates of the mean at the trust region centers change with increased iterations, and this is reflected by the different objective function values for the same trust region center. Of course, as replications increase, the variation in the estimate of the mean decreases, and the confidence in the true value increases, and this is seen

with the yellow-colored trust-region center, which is the center from iterations 800 to 2,100. We start with a post radius of about 2.5μm and after 2,500 iterations, the incumbent solution is about 1.67μm.

4.5 CONCLUSIONS

In this chapter, we provide implementation details and algorithmic enhancements to the theory developed in Chapter 3 in form of a package we call SO-LVIT. The implementation is then tested against other available implementations from the literature, with promising results. Then, examples from two different areas of engineering, namely inventory planning and DNA separations, are investigated through the simulation optimization framework that we propose in the previous chapters.

The first example comes from chemical production operations and supply chains, where the simulation is a discrete-event simulation and the objective is to maximize profit subject to customer satisfaction considerations. The evaluations here are relatively cheap, the variance in the output moderate with regard to curvature of the underlying surface. The decision variables are around 8–10 in number. The challenge here involves the presence of stochastic inequality constraints which required us to satisfy a certain customer satisfaction level. These constraints were addressed naturally using an exact penalty method within the simulation optimization framework which guarantees convergence at a pre-specified level of probability.

The second example came from the biomolecular engineering domain within chemical engineering, where the underlying phenomena were modeled using a system of stochastic differential equations. The objective is to identify an optimal design parameter that would expedite DNA separations through a particular electrophoretic method. Here, even though we were dealing with essentially one decision variable, the simulations are extremely expensive, and the variance in the output is extremely high compared with the curvature of the surface.

In addition to the above examples, this framework may also be used within a response surface methodology framework when optimizing over experiments. In the future, we would like to include examples that involve real experiments which clearly take time.

Part III

SEQUENTIAL AND SIMULTANEOUS STRATEGIES FOR OPTIMIZATION UNDER UNCERTAINTY: MODELS AND APPLICATIONS FOR SITE-WIDE MAINTENANCE

Maintenance turnarounds of plants in integrated sites represent large costs and major disruptions. This part addresses (1) long-term maintenance planning via two discrete-time MILP models to determine optimal turnaround frequency and to maximize long-term profit through synergistic coordination of maintenance turnarounds; and (2) medium-term maintenance scheduling under duration uncertainty via models that are robust to variable manpower availability requirements while maximizing expected profits through consideration of possible recourse actions in production decisions.

5

LONG-TERM TURNAROUND PLANNING FOR INTEGRATED SITES

5.1 INTRODUCTION

Plant turnarounds are periodic, necessary disruptions in material flow through chemical production sites that involve multiple tasks that help to improve plant reliability, arrest failure rates, and to boost productivity and profits. However, they not only incur enormous costs and consume resources, but also result in lost sales. As a result, turnarounds significantly affect demand and supply of materials within a chemical site network and are tightly coupled to production planning, resource planning and inventory management decisions.

The purpose of this work is to present modeling approaches to optimize maintenance turnaround frequencies as well as site-wide long-term turnaround schedules, while accounting for these practical considerations and constraints. We demonstrate the effectiveness of the models by applying them to a representative industrial-size chemical site network, and analyzing optimized schedules. Of particular interest is that the resulting schedules often lend themselves to the rational justification of the sequences and alignment of tasks, based on basic network structure and economic arguments. This gives further insight into the behavior of the network, a feature that will appeal to practitioners in the area.

Optimal turnaround scheduling of integrated chemical sites is a challenging combinatorial problem due to the following factors:

- *Operational constraints:* Due to complex flow relationships between production units, a turnaround performed on one of the units can result in blocking of upstream operations or starving of downstream operations. Typically these

relationships require involved analysis because of the availability of buffer storage capacities, as well as priority rules for allocation of raw material under limited supply.

- *Timing of turnarounds:* Each production unit in an integrated site has its own turnaround frequency and duration. In addition, depending upon the location of the site, turnarounds cannot be carried out during certain time periods in a year to ensure safety of the workforce. For example, for a site located in Northern Canada, it might not be appropriate to carry out the turnaround in winter months due to extreme weather conditions.

- *Resource constraints:* Turnarounds for a production envelope (a subset of plants that interact, are concerned with the same family of products, and are typically aligned with a specific business) within a site can require a large workforce of a few hundred personnel with different skill sets (such as welders, assembly workers, supervisors). Management and allocation of maintenance personnel in a site requires efficient planning and coordination. For instance, the sporadic requirement of a large workforce for a short interval can be difficult, as it is typical for such workers to be hired on contract. It may be difficult to ensure their hire if there are turnarounds being carried out by competitors, if the site is in a remote location, or if the logistics associated with a large workforce descending upon a site is cause for safety concerns.

- *Financial impact:* Turnarounds can have a significant impact on the revenue of the company. For a large integrated site, these turnarounds can lead to significant reduction in production rates and thus a reduction in short-term revenues. In order to reduce the impact on financial results, the turnarounds must be appropriately scattered over the planning horizon.

This work contributes to literature by (1) extending existing models for determining process unit maintenance frequencies from profit and failure rate relationships to determining turnaround frequencies of a plant unit with multiple subunits; (2) developing mixed-integer programming (MILP) formulations for long-term maintenance planning of integrated sites with continuous chemical plants; (3) incorporating financial performance, timing constraints, and manpower availability—considerations that require looking at a time and size scale not studied previously;

and (4) demonstrating the real-world applicability of potential schedules that have been optimized under various scenarios.

The remainder of Section 5.1 provides a brief background on turnaround scheduling and a literature review on maintenance planning in various industries, including the process industry. Section 5.2 provides an example network structure, the associated unit information required for optimization, and uses a small network example to motivate the potential for optimization. Section 5.3 provides a mixed-integer model for determining optimal turnaround frequencies, and provides a framework for deciding whether a plant needs one or multiple kinds of turnarounds. Section 5.4 outlines the solution approach, and provides details of the MILP model. Section 5.5 provides results, and a detailed analysis of them along with financial and sensitivity studies, and alternative formulations and results for other scenarios. Finally, Section 5.6 summarizes the work and discusses possible extensions and directions for future study.

5.1.1 *Definition, concepts, and significance of maintenance scheduling*

Maintenance can be defined as all actions appropriate for retaining an item/part/equipment in, or restoring it to a given condition (Dhillon 2002). A maintenance turnaround is the periodic shutdown of chemical plants for overhaul. A turnaround may be required to (1) prevent unplanned shutdowns due to equipment failures or wear and tear (e.g., in the case of pumps and compressors), (2) replace aging parts and instrumentation, (3) perform cleaning of pipes and equipment, (4) replace catalysts, and (5) perform welding or other structural reinforcement tasks.

Approaches to maintenance scheduling vary widely and depend on (1) whether the maintenance is preventive, corrective, or opportunistic; (2) what the sources of uncertainty in the operations are; (3) whether operations are multipurpose batch processes or continuous plants; and (4) whether maintenance planning is short-term or long-term.

The American National Standards Institute definition of preventive maintenance (2014) is *the planned maintenance of plant infrastructure and equipment with the goal of improving equipment life by preventing excess depreciation and impairment.* On the other hand, planned corrective maintenance is the maintenance carried out after a failure has occurred and intended to restore the item to a state in which it

can perform its required function. Opportunistic maintenance is the exploitation of failure events to plan other maintenance activities in conjunction, and the altering of future maintenance schedules based on this.

Long-term maintenance planning uses information such as equipment reliability, usage, and maintenance histories and manufacturers recommendations to determine the approximate moments or frequencies of preventive maintenance. The solution to this problem may result in a list of equipment items that are due to be maintained over the next short-term scheduling period.

The primary task in short-term maintenance scheduling is to develop a scheme with detailed timing of maintenance activities that allots resources (maintenance crews, workers, and equipment) to tasks (machines or units) and satisfying certain constraints (such as crew availability, network constraints, or shift constraints), while not only maintaining regular production or operation to satisfy customer demand, but to do it in an optimal fashion so as to minimize losses, down-times, customer dissatisfaction, or some other metric. On many occasions, if a production schedule is also required, it is done side-by-side (either simultaneously or in a sequential-iterative manner) with the maintenance scheduling.

According to Christer & Whitelaw (1983), annual expenditure on maintenance by a medium-sized company at the time exceeded GBP 1 million. Tan & Kramer (1997) provide extensive references for costs from production losses due to equipment down-time. They estimate that lost production costs in a chemical plant may range from \$500-\$100,000 per hour; and that refineries experience about 10 days of down-time every year, with losses of up to \$30,000 per hour. Grievink et al. (1993) estimate that about 50% of operating cost variability comes from maintenance. Large chemical companies budget annually for spending on the order of hundreds of millions of dollars on maintenance, just for parts and manpower and not including the value of lost sales. According to a more recent estimate by Industrial Info Resources (2014), a global marketing intelligence agency, major ethylene plants commonly schedule turnarounds once every few years, where the average duration is 25 days at an average cost of \$15MM or more (Global Ethylene Database 2014). Figure 1.3 in Chapter 1.1 illustrates that cumulative expenditure on maintenance turnarounds in the chemical process industry exceeded USD 1 billion in 2008, not including lost sales. Thus, the optimization of maintenance turnaround operations and the associated production planning could significantly improve profit margins.

5.1.2 *Literature Review*

5.1.2.1 *Maintenance scheduling in chemical engineering*

Formulations of maintenance scheduling models in chemical engineering depend on whether the plants are continuous or batch, whether it is for a single process or site-wide, and approaches taken (Monte Carlo with genetic algorithms or MILP models).

Dedopoulos & Shah (1995a) investigate the short-term maintenance scheduling problem for multipurpose batch plants. The task is to generate detailed timing of maintenance activities simultaneously with a production schedule. The task involves the allotment of crews to various tasks. The study performed in the paper is that of a lubricant plant, with three grades, two blenders, seven storage tanks, five packaging lines and three warehouses. The problem is formulated as an MILP with 2000 binary variables, 2000 continuous variables and about 5000 constraints. Uncertainty in intermediate stages of operation is not incorporated in the proposed formulation, and the number and allotment of maintenance tasks is predetermined.

Sanmartí et al. (1997) deal with production and maintenance scheduling for multipurpose batch plants under equipment failure uncertainty. Uncertainty analysis of failure is performed prior to production scheduling to mitigate disruptions, and to use the fixed maintenance schedule as long as possible without changing it. The number of production tasks and, hence, the number of maintenance tasks depend on the production requirements of the corresponding time period. The work also introduces certain reliability metrics, for both equipment as well as the entire schedule to reduce occurrences of equipment failure and to increase robustness in the case of equipment failure.

Tan & Kramer (1997) discuss the economics of maintenance, several approaches to maintenance (including preventive, corrective, and opportunistic) and a number of issues regarding modeling of equipment failure in their process of developing a general framework for maintenance scheduling. They use a combination of Monte Carlo simulation and a genetic algorithm to generate the maintenance schedule.

Pistikopoulos et al. (2001) formulate an MILP to perform production scheduling and exploit idle times in equipment utilization to perform preventive maintenance. Piecewise-constant failure rates of equipment are assumed, and these are incorpo-

rated in a multi-period planning model. Explicit up-time constraints are included to quantify availability of equipment.

Cheung & Hui (2001) formulate an MILP to perform long-term maintenance scheduling for a chemical complex with eight plants and associated utility plants. The planning horizon considered is two years, with monthly planning periods. A cyclic schedule is built, assuming that each unit needs to be shut down for maintenance once a year. Scenarios including availability of intermediate storage, and the import of electricity and intermediate products are considered.

Cheung et al. (2004) formulate a multi-period MILP to perform site-wide short-term maintenance scheduling. The task is to decide which plants to shut down in each planning period subject to a site model (that contains information about plant material balances and utilities). The model, although built with deterministic times for maintenance, accommodates varying electricity prices and demands over time periods.

Laggoune et al. (2009) propose a method for opportunistic maintenance for continuous plants, in a situation where different units in the process interact with each other. The task is to decide which units have to be taken down, and hence which items are to be replaced/maintained in which time period. Here, the breakdown of a particular unit or piece of equipment is seen as an opportunity to replace/maintain other parts that are also taken down by the event, and the decision to do this is based on a trade-off between the cost associated with the possible breakdown of associated units before the next maintenance cycle and the cost involved in performing maintenance at that juncture. A Monte Carlo scheme is used to simulate the breakdowns using failure distributions, and an algorithm is outlined to choose the maintenance schedule (the frequency of maintenance for each component, and which components undergo maintenance together) based on the simulated breakdowns. The case study investigated is that of a refinery centrifugal compressor.

Megow et al. (2011) also consider turnaround scheduling in the chemical industry, specifically in continuous plants. The task is to minimize the cost of maintenance with respect to the resources used, which are manpower and maintenance equipment. This minimization is subject to pre-set precedence rules for maintenance tasks and resource scheduling constraints that involve shift calendars for maintenance workers. The assignment constraint, which assigns maintenance resources to jobs in each time period, gives the detailed maintenance schedule. The interesting trade-off is

the time-cost trade-off, where more expensive external resources can be availed of, in order to perform a certain task in reduced time.

Castro et al. (2014) consider the maintenance scheduling of a gas engine power plant, where a single maintenance crew is available for maintenance. The task is to schedule the shutdown of parallel units so as to minimize idle time and shutdowns in high-tariff periods assuming seasonal variation in electricity prices. In addition, the model allows for staggering the duration of shutdowns in the case of low power demands. A continuous-time formulation is proposed, and a general disjunctive programming scheme is used to solve the problem efficiently.

From this literature survey, it is apparent that most work focuses on performing short-term maintenance scheduling with only brief mentions as to how one might tackle the long-term planning problem. Ideally, one would like to reconcile long-term and short-term schemes; incorporate uncertainty in operations; and address enterprise-wide maintenance scheduling rather than just single process or plant maintenance, while maintaining production schedules and demand satisfaction criteria. Other details in the model would depend on the specifics of the plants under consideration, the economics involved, and the priorities and targets of the company.

5.1.2.2 *Maintenance scheduling in other fields*

Academic literature on maintenance scheduling and related areas appears in the fields of airline scheduling and logistics, electrical power generation and distribution, and in the manufacturing and the chemical industries. A list of industries and a sample of associated papers from the literature is available in Table 5.1.

Certain features from these fields may be relevant in the chemical industry. For instance, the power generation literature addresses maintenance scheduling for continuous plants, where the task is to perform preventive maintenance on some plants while satisfying electrical power demand to all customers at the required level. Maintenance problems in the airline industry involve assigning maintenance crews to different tasks and may involve the cost of transporting maintenance crew—a feature that may be relevant when tackling enterprise-wide maintenance problems in the chemical industry.

Table 5.1: Industries and associated literature in maintenance scheduling

Industry	Papers
Airlines	Moudani & Mora-Camino (2000), Sriram & Haghani (2003), Feo & Bard (1989), Cohn & Barnhart (2003)
Chemical industry	Tan & Kramer (1997), Sanmartí et al. (1997), Pistikopoulos et al. (2001), Laggoune et al. (2009)
Manufacturing	Yao (2004), Sloan & Shanthikumar (2000), Gharbi & Kenné (2005), Yang et al. (2008)
Electrical Power	Yamayee (1982), Satoh & Nara (1991), Mukerji (1991), Volkanovski et al. (2008)

5.2 PROBLEM DESCRIPTION

5.2.1 *The ideal solution and scope of current work*

The most general and detailed solution to turnaround optimization involves addressing a multi-faceted problem. This chapter addresses turnaround optimization on a multi-year time scale and Chapter 6 focuses on a medium-term time scale of several months.

As is the case with any operations problem, there are several uncertainties that one may consider, including demands for products, raw material price and availability, failure rates of components, reliabilities in production, and lengths of turnarounds. These kinds of uncertainties can be incorporated to some extent by the use of historical data. In practice, these effects are coupled with evolving business strategy, new process integration, infrastructure investments, and capacity expansion—effects that are difficult to capture and quantify.

It may be the case that individual envelopes or businesses within the integrated site optimize their respective operations separately. The challenge is to demonstrate the benefits of synergistic decisions across the entire site through holistic modeling and optimization, which is one of the aims of this work.

Given the issues of scale and uncertainty, it is impossible to determine the perfect decisions for turnarounds (and other operational decisions, as they are invariably tightly coupled). However, as the previous sections elaborate, turnaround decisions can have a large financial impact, and there is still immense potential to make better, more informed decisions to improve asset utilization as well as financial performance. The choice of solution approach is important in order to decide the right scale of problem—both in terms of process detail, as well as time scale—to consider.

Uncertainties arising from component failures may be modeled and studied in order to decide the frequency of turnaround for each unit. The uncertainty associated with production reliability and demands and supply of products and raw materials within the site are short term effects, and the study of these may result in defining base stock levels for inventory tanks. In Section 5.3, we show how turnaround frequencies and policies may be determined, assuming that data corresponding to the deterioration of profit over time is given. We assume that inventory base stock levels are available to us. Uncertainty in length of turnarounds due to discovery work requires analysis on a shorter time scale as well, as this affects the manpower resource availability and ability to satisfy demand in the required time. Accounting for the effect of discovery work is addressed in Chapter 6.

Uncertainty in external demands is a significant factor for production planning problems on a six-month to one-year horizon for the bulk chemical industry. In the turnaround optimization setting, we deem it sufficient to consider aggregate demands from each plant rather than individual demands of multiple products in a plant, as plant turnarounds interrupt all production. The variability in aggregate demands is less, and we choose to avoid explicitly considering demand variability in our formulation.

The scope of the current chapter is primarily to focus on providing long-term plans for turnarounds, while modeling production and inventories in sufficient detail without delving into short-term effects. To this end, we look at a plant-level abstraction and a time horizon of several years. Emphasis is placed on incorporating practical, real-world constraints and considerations as outlined in Section 5.1 such as seasonal constraints, purchase of intermediate inventory from the market, financial performance, peak manpower reduction, and general models that allow for transitioning into new schedules and responding to disruptions. We use a sample industrial-size integrated site network to study the tractability of mixed-integer op-

timization models under these constraints, interpret the applicability of resulting schedules, and present the first work that demonstrates the value of optimizing integrated site turnarounds at this scale.

5.2.2 *Problem setup*

The chemical site network used as a running example for this study is illustrated in Figure 5.1. The red units with the black outline indicate final products, the white units process intermediate products and the green cylinders indicate inventory capacity. The solid arrows between units and storage tanks represent connecting pipes that impose flow restrictions in the network. Dotted blue arrows close to certain storage tanks indicate the ability to sell and purchase intermediate product to and from the market.

Each unit in the site network represents an entire plant, which is typically taken down in its entirety during a turnaround. As can be seen, the site is tightly integrated, and the shutdown of a plant will affect the production of plants upstream and downstream to it.

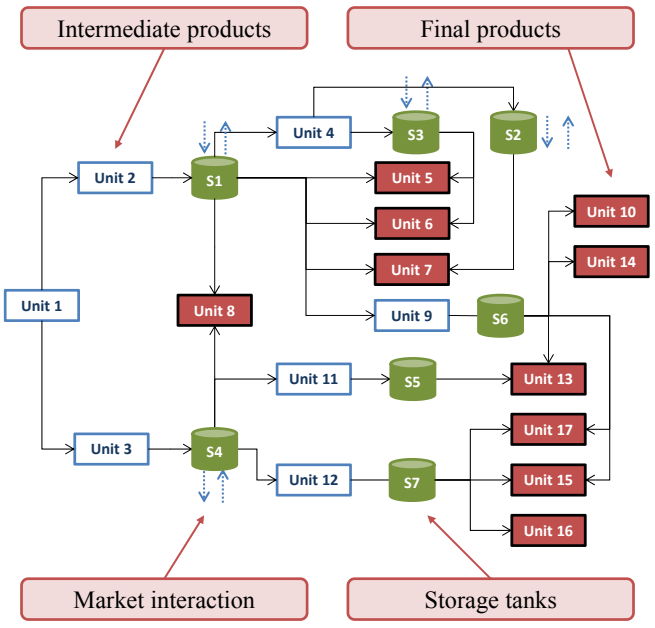


Figure 5.1: Example site network abstracted at the plant level for turnaround scheduling purposes

The inventory tanks between units represent capacity to deal with both planned and unexpected disruptions. Inventory management is especially important within chemical and petrochemical plants as compared with other manufacturing sites, as inventory capacity is expensive—intermediate products typically require cryogenic storage or special measures for storing hazardous substances—and storage tanks can cost on the order of tens of millions of dollars. Inventory level set points provide a buffer to deal with short-term fluctuations in production, and consequently in supply and demand. Larger storage tanks provide for longer outages as well, such as turnarounds.

Different plants require turnarounds that are estimated to last a certain amount of time (usually on the order of weeks), and these turnarounds are done at a certain frequency. The frequencies are estimated from historical data, reliability studies for various components, and other practical considerations using, for example, the scheme developed in Section 5.3.

Each turnaround is associated with a manpower requirement, which represents the primary cost of a turnaround. To minimize lost sales from turnarounds, it is usually beneficial to take down certain combinations of plants together.

Figure 5.2 reveals some of the decisions that have to be made. Here, f indicates the time period (1/frequency) of turnarounds, and T indicates the number of weeks to perform the turnaround. The margins represent the relative profit margins of the final products from the corresponding units. First, the choice of implementing a cyclic schedule or a rolling horizon schedule must be made. If we seek a cyclic schedule, the time periods of all the units need to be aligned and a least common multiple of the time periods is chosen to represent the length of the cycle. So, one may choose to maintain Unit A every five years and Unit C every two-and-a-half years, or improve reliability to make Unit B require maintenance once every three years. Next, if the capacity of S cannot hold more than, say, 3 weeks of inventory, then it may be a good idea to take down Units B and C for maintenance along with Unit A.

On the other hand, as the downstream units require maintenance twice as frequently as the upstream unit, it may be more economical to line up, say, Unit A's turnaround with Unit B's turnaround, while using inventory to feed Unit C in the mean time; and take Unit C down asynchronously and build up inventory when only Unit A and Unit B are running.

A third situation may involve the inventory capacity being big enough, or import/export capability to the intermediate storage, which may allow all turnarounds to be decoupled. Of course, if turnarounds occur together, then the decision of how to stagger them needs to be made as well. Which of these scenarios is optimal depends on the several, specific numbers involved—the pipe capacities, minimum flows for units, relative profit margins of products, frequencies, inventory set points and capacities, and times for maintenance. All of these factors along with the potential number of scenarios present an opportunity for an optimization-based analysis of the network. Tight coupling of this small network to the larger site network, the presence of units requiring multiple inputs or producing multiple outputs, and the existence of recycle loops further complicate the optimization of these turnarounds.

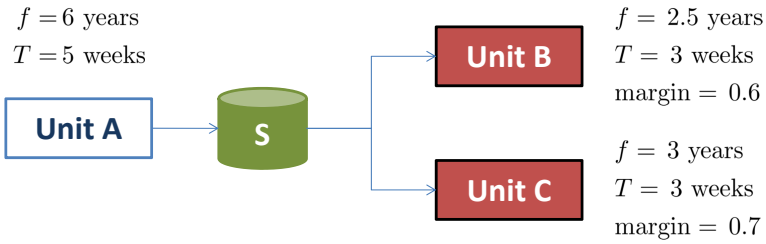


Figure 5.2: Illustration of interactions and interdependencies in a subnetwork

From a long-term perspective, it is typically more economical to minimize overall lost sales by taking down the entire site at once. This is, in practice, not a viable alternative. One limitation is that on peak manpower, and the other is economics. Competition for manpower, diminishing marginal manpower utility, and upper limits on available manpower in the nearby geographic region make it important to keep peak manpower as low as possible. In addition, shutting down an entire site for maintenance at once severely impacts short-term cashflow due to reduced production levels. These considerations indicate the need for a more involved study to determine turnaround schedules.

5.3 MIXED-INTEGER LINEAR PROGRAMMING MODEL FOR DETERMINING OPTIMAL TURNAROUND FREQUENCIES

In this section, we provide an extension to the mixed-integer model proposed in Dedopoulos & Shah (1995b) for determining a preventive maintenance policy for a

particular plant unit. The paper proposes a long-term maintenance model to determine frequencies of maintenance for a particular process unit or piece of equipment within a plant. The model assumes that we have data for expected profit over time. Such information may be determined by analyzing failure rates and reliabilities of individual equipment and their effect on production levels, and hence profit. This can be done, for instance, by a modification of the model proposed by Kondili et al. (1993) to a multi-stage stochastic program to account for the possibility of equipment failure at each time interval. In general, it is common in practice to perform extensive analyses on failure rates and their effect on productivity, and we too assume that these relationships are available for our purposes.

The problem we focus on is to decide, given the profit over time relationship from reliability analysis for individual pieces of equipment or a groups of process units, what type of turnaround policy should be implemented for a particular plant and how frequently it should be performed.

Note that we use the term expected profit to be consistent with notation in Dedopoulos & Shah (1995b). In practice, this may actually correspond to (negative) costs of operation for each unit group considered. In this case, we assume to have relationships of costs over time for each unit group. These costs may be associated with utility costs, operating costs, and minor maintenance/upkeep costs. For instance, increased fouling over time in a heat exchanger may result in increased utility costs.

The objective function is to maximize expected profit earnings,

$$\max \sum_{t \in T \setminus \{0\}} E_t, \tag{5.1}$$

where the individual earnings over discrete time intervals are accumulated to calculate the long-term expected profit.

Next, we assume that the expected profit in a time period can be approximated by an average of the profits from each of the groups of process units. For instance, each term corresponds to the expected profit of the plant when each individual group has a given current state of productivity and the rest of the units are in

Table 5.2: Set, parameter and variable notation for the turnaround frequency determination model

Entity	Set
Set of unit groupings	K
Time period	T
Parameter	**Notation**
Expected profit in a maintenance period	ρ_k^m
Big-M parameter for expected profit	M^{big}
Expected profit over time for a particular unit group	$R_{t,k}^{\text{calc}}$
Variable	**Notation**
True expected profit per period	E_t
Expected profit for a particular unit group	$R_{t,k}$
Binary representing a maintenance period	$z_{t,k}$
Binary for representing time since last maintenance	$q_{t,t,k}$

perfect condition. However, the profit in a particular period drops if maintenance is taking place. These conditions are modeled through

$$E_t \leq \frac{1}{|K|} \sum_{k \in K} R_{t,k} \qquad \forall\, t \in T \backslash \{0\}, k \in K, \tag{5.2}$$

to represent the profit in a period when no maintenance is taking place, and

$$E_t \leq (1 - z_{t,k}) M^{\text{big}} + \rho_k^m \qquad \forall\, t \in T \backslash \{0\}, k \in K, \tag{5.3}$$

which bounds the expected profit in a period based on whether maintenance takes place in the period or not.

The remainder of the formulation mirrors the development in Dedopoulos & Shah (1995b). First we compute the profit in a period with no maintenance through

$$R_{t,k} = \sum_{\tau \in T} q_{t,\tau,k} R_{\tau,k}^{\text{calc}} \qquad \forall\, t \in T \backslash \{0\}, k \in K, \tag{5.4}$$

where $q_{t,\tau,k}$ assumes the value 1 if, at time t, maintenance was performed for the kth unit group τ time periods ago. Then we ensure that either preventive mainte-

nance occurs or the binary variable representing the time since the last preventive maintenance is nonzero by

$$\sum_{\tau \in T} q_{t,\tau,k} + z_{t,k} = 1 \qquad \forall\, t \in T \backslash \{0\}, k \in K. \tag{5.5}$$

We also enforce that if the binary variable representing the time since the last maintenance takes the value 1, the corresponding binary variable in the next time period also does, unless preventive maintenance takes place through

$$q_{t+1,\tau,k} \geq q_{t,\tau-1,k} - z_{t+1,k} \qquad \forall\, t \in T, |T| > t \geq \tau > 0, k \in K, \tag{5.6}$$

and

$$q_{t-1,|T|,k} \leq q_{t,|T|,k} + z_{t,k} \qquad \forall\, t \in T \backslash \{0,1\}, k \in K. \tag{5.7}$$

We prevent maintenance from taking place in two consecutive time periods,

$$q_{t+1,0,k} = z_{t,k} \qquad \forall\, t \in T \backslash \{0\}. \tag{5.8}$$

We also assume that maintenance has just taken place at the first time period

$$z_{1,0} = 1. \tag{5.9}$$

Test cases We now investigate some examples, where the aim is to determine whether (1) the different sets of units considered should be taken down together as one turnaround; (2) the different sets of units undergo major and minor turnarounds; or (3) the turnarounds are decoupled, and each has its respective turnaround frequency.

We consider a 60-month (5-year) horizon with monthly discretizations, and look at two sets of units within a particular plant to see how their turnarounds may be coordinated. The formulation accommodates situations with more such sets as well. This results in models with about $7,500$ continuous variables, $7,500$ discrete variables, and about $4,500$ constraints.

For the expected profit vs. time relationship that we assume given, we use a linear degradation in expected profit scenario for illustration. We illustrate the findings in Figure 5.3. The figure illustrates the turnaround policy recommended by the

model for different scenarios. In the first and second examples, we see that the optmizer recommends that both unit groups undergo synchronized maintenance every 20 months, and every 30 months. In the next two example, the recommendation is to perform turnarounds on the two groups at different frequencies, but to sync the two when suitable for a major turnaround, and have minor turnarounds at the other time points. In the last example, it seems most favourable to completely decouple the two turnarounds.

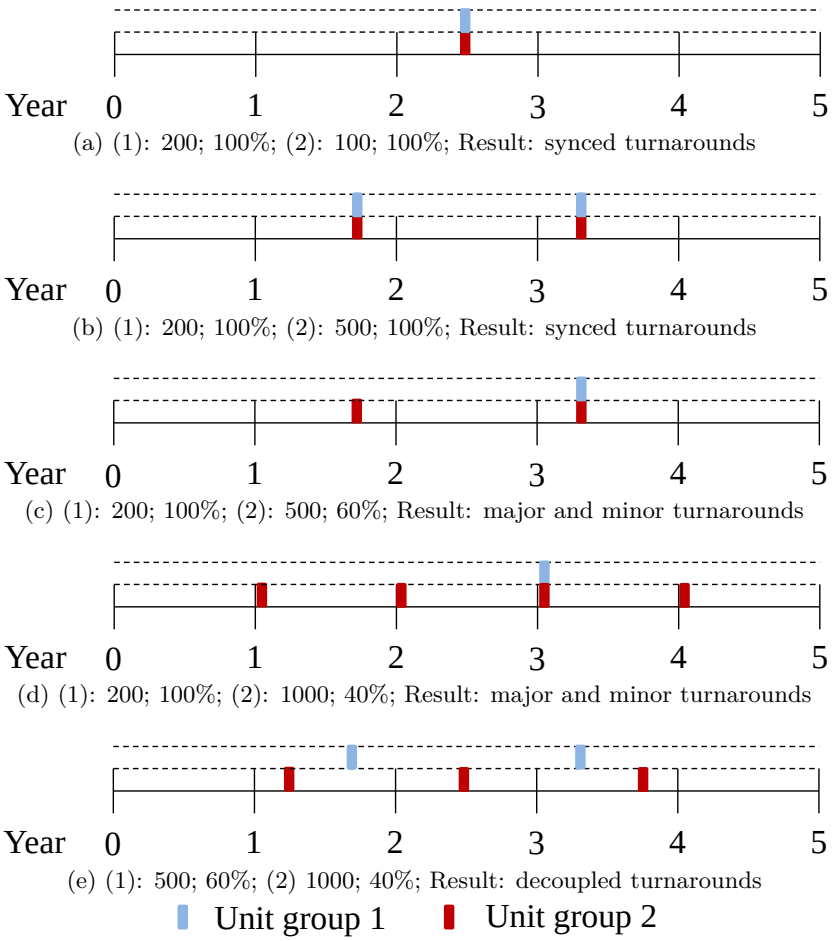


Figure 5.3: Illustration of coordination of subtasks in different scenarios (rate of profit decline (units/month); production decline during maintenance (%); recommended turnaround policy)

The model was solved by CPLEX 12.5 under GAMS 24.3, and each scenario was solved to a zero optimality gap, given a 1000-second budget. Formulations for handling the different turnarounds policies, namely (1) synced turnarounds; (2) major and minor turnarounds; and (3) decoupled turnarounds are described in later part

of Section 5.5.2. In summary, we now have a mechanism to determine turnaround frequencies, and these serve as inputs to the models described from Section 5.4 onwards. Exact turnaround frequencies may not be used, but a bound on the time between turnarounds may be used instead to allow for flexibility, as in Section 5.5.2.

5.4 MIXED-INTEGER LINEAR PROGRAMMING MODEL FOR INTEGRATED TURNAROUND PLANNING

The approach taken in this work is to develop a deterministic, discrete-time mixed-integer programming model that abstracts the process site network to a level that balances inclusion of relevant detail with tractability. Relevant constraints are added for mass balances, unit ratios, turnarounds, demands, and manpower. The abstractions and assumptions made for the mixed-integer model are detailed in the following section for each case but, in general, the validity of these will depend on the particular site.

In this section, we formulate the MILP scheduling model to recommend a potential cyclic schedule for performing turnarounds over a multi-year horizon. We work with this formulation initially, and explore alternative scenarios, extensions, and assumptions in subsequent sections. Relevant notation is listed in Tables 5.3, 5.4, and 5.5.

Table 5.3: Set notation for the long-term scheduling model

Entity	Set
Process or storage unit, or external interaction	I
Units requiring inputs to be in a certain ratio	I_{in}
Units requiring outputs to be in a certain ratio	I_{out}
Unit with flow	F
Time period	T
Unit requiring turnaround	M
Final product	P
Unit allowing import of raw material from market	B

Table 5.4: Parameter notation for the long-term scheduling model

Parameter	Notation
Turnarounds	
Number of turnarounds for ith unit	u_i
Number of time periods between turnarounds of ith unit	f_i
Manpower required for maintenance of ith unit	m_i
Number of time periods to perform turnaround of ith unit	n_i
Manpower limit	$m^{\max}$
Flows	
Upper bounds on flows	$x_{i,j}^U$
Lower bounds on flows	$x_{i,j}^L$
Inventories	
Lower inventory limit	s_i^L
Upper inventory limit	s_i^U
Initial inventory	s_i^0
Costs/prices	
Price of product from unit i	$p_{i,t}$
Cost of manpower	$c_{m_{i,t}}$
Cost of imported material i	$c_{r_{i,t}}$
Holding cost in inventory tank	$h_{i,t}$
Demands on final products	$\delta_{i,t}$

Table 5.5: Variable notation for the long-term scheduling model

Variable	Notation
Flow between units i and j at time period t	$x_{i,j,t}$
Inventory at time t of unit i	$s_{i,t}$
Binary representing start of maintenance for unit i at time t	$y_{i,t}$

Objective function The objective

$$\max \sum_{t\in T}\left(\sum_{i\in P} p_{i,t}\, x_{i,\infty,t} - \sum_{i\in M} m_i\, y_{i,t} \sum_{\tau=t}^{t+n_i-1} c_{m_{i,\tau}} - \sum_{i\in B} c_{r_{i,t}} x_{0,i,t} - \sum_{i\in I\setminus F} h_{i,t} s_{i,t}\right) \tag{5.10}$$

represents the maximization of net present value (NPV). The relative margin, $p_{i,t}$, reflects the prices that final products are sold at, as well as raw material, processing and production costs per unit of final product. Here, product, manpower, and intermediate import prices vary with time. Manpower costs are a function of the unit, as each unit may require different ratios of workers with different skill sets. The coefficient $c_{m_{i,t}}$ then represents the average cost. The subscripts $0 \in I$ and $\infty \in I$ indicate material sourced from outside the network and material exiting the network, respectively.

Network flow constraints The first constraint represents a standard mass balance constraint, which accounts for flows in and out of nodes in the network (plants or inventory tanks), and links inventory levels across time periods in the case of storage units. The constraint may be formulated as

$$\sum_{i\in I} x_{i,j,t} + s_{j,t-1} = s_{j,t} + \sum_{i\in I} x_{j,i,t} \qquad j \in I,\, t \in T, \tag{5.11}$$

where

$$s_{j,t} = 0 \qquad j \in F,\, t \in T \tag{5.12}$$

is used to distinguish between inventory units and process units, and disallows inventory storage in process units.

As chemical plants typically involve the chemical transformation of feed streams into final products, network flow constraints between plants include maintaining a certain prespecified ratio between streams entering the plants (e.g., the stoichiomet-

ric ratio), $r^{\text{in}}_{i,j}$, or, analogously, between product streams, $r^{\text{out}}_{i,j}$, if there are multiple products, as in

$$\begin{aligned} x_{k,i,t} &= r^{\text{in}}_{k,j}\, x_{j,i,t} \qquad \forall\, t \in T,\, i \in I_{\text{in}},\, j,k \in I, \\ &\quad \text{where } x^U_{j,i}, x^U_{k,i} \neq 0,\, j < k, \end{aligned} \tag{5.13}$$

$$\begin{aligned} x_{i,j,t} &= r^{\text{out}}_{j,k}\, x_{i,k,t} \qquad \forall\, t \in T,\, i \in I_{\text{out}},\, j,k \in I, \\ &\quad \text{where } x^U_{i,j}, x^U_{i,k} \neq 0,\, j < k, \end{aligned} \tag{5.14}$$

In reality, plants may use different ratios of raw materials for different product grades, and though different feed qualities may result in different output ratios, we assume these unit ratios are fixed and we use the time-averaged values.

In terms of inventory capacity constraints, there are

$$s_{j,t} \geq s^{\min}_j \qquad j \in I \backslash F,\, t \in T \tag{5.15}$$

$$s_{j,t} \leq s^{\max}_j \qquad j \in I \backslash F,\, t \in T \tag{5.16}$$

which ensure that the inventory level of a tank does not exceed its storage capacity at any time. Similarly, there is a constraint on the lower limit on the storage tank levels, and is set at a predefined base stock level. There are also standard nonnegativity constraints on all of the flows in the network,

$$x_{i,j,t} \geq 0 \qquad i,j \in I,\, t \in T. \tag{5.17}$$

In some cases, certain production plants cannot run below a minimum operation level, and we formulate this level as a fraction $\mu_{i,j}$ of the maximum operating capacity,

$$x_{i,j,t} \geq \mu_{i,j}\, x^U_{i,j} \left(1 - \sum_{\tau=t-n_i+1}^{t} y_{i,\tau} \right) \qquad i \in F,\, t \in T, \tag{5.18}$$

where the summation is required as the binary represents only the starting time period of the turnaround for each unit.

It should be noted that the flow levels and inventory levels are coarse estimates, as the time discretization we use is on the order of weeks. As the inventory tanks are not capable of holding a week's worth of production, a study on a shorter time scale is warranted if we want to focus on inventory and daily production decisions.

Turnaround constraints The next constraints enforce the turnaround constraints in the case of a fixed cyclic schedule. Here the length of the cycle is a multiple of each unit's turnaround time period (1/frequency). The constraint

$$\sum_{t \in T} y_{i,t} = u_i \qquad i \in M, \tag{5.19}$$

enforces the number of times a unit has to go down for a turnaround in each cycle. The performance of a turnarounds at time period intervals are enforced by equating the corresponding binaries, as in

$$y_{i,t} = y_{i,t+f_i} \qquad i \in M,\, t \in T. \tag{5.20}$$

To ensure that flows out of a unit are zero during maintenance, we write the big-M constraint

$$x_{j,i,t} \leq x^U_{j,i} \left(1 - \sum_{\tau=t-n_i+1}^{t} y_{i,\tau} \right) \qquad t \in T,\, i \in M,\, j \in I. \tag{5.21}$$

Manpower resource constraints As far as resource constraints are concerned, the primary concern is with manpower availability. This is a major consideration in turnaround scheduling, as (1) maintenance manpower is typically hired on contract; (2) chemical sites are always competing for manpower and, as a result, manpower requirements must be determined well in advance; (3) accommodating and transporting a large number of maintenance personnel to remote site locations may be difficult; and (4) a significant increase in the workforce at a site may cause safety concerns. The following constraint ensures that a hard limit on the peak available manpower is respected.

$$\sum_{i \in M} \sum_{\tau=t-n_i+1}^{t} y_{i,\tau} m_i \leq m^{\max} \qquad t \in T. \tag{5.22}$$

Finally, we define binary variables, which are turned "on" to simulate the starting time period of a maintenance turnaround.

$$y_{i,t} \in \{0, 1\} \qquad i \in M,\, t \in T. \tag{5.23}$$

Demand constraints To model demand in general, one should consider tank farms that may be available downstream to final product lines, in order to model inventory build up to satisfy customer demands during turnaround periods. However, with the weekly timescale that we focus on, modeling this build up of inventory with accuracy is beyond the scope of this work. To model demand, one may include constraints of the form

$$\sum_{t \in D_t} x_{i,\infty,t} \leq \delta_{i,t} \qquad i \in P,\ D_t = \{t, \ldots, t + d_t\}, \tag{5.24}$$

where D_t denotes the set of time periods included in the demand period starting at time t, and d_t denotes the length of the demand period. Some sort of penalty (proportional to the quantity by which demand is not satisfied, or the number of times it is not satisfied) may be incurred if demand is not satisfied (e.g., Jung et al. (2004)). Typically monthly demand forecasts may be known to sufficient accuracy for six months into the future, and beyond that estimates may be used for quarterly demands based on historical data. In the case of turnarounds, if it is known well ahead of time when a turnaround is taking place, it may be possible to negotiate with a customer regarding reduced supply during the corresponding period. In this study, we do not focus on how to model penalties on unmet demand, and assume that either there is some prior negotiation with a customer or that intermediate raw materials may be purchased from the market during known periods of reduced supply (turnarounds).

The above objective and constraints constitute the major part of the mixed-integer programming model. Extensions to this are discussed in Sections 5.5.1 and 5.5.2.

5.5 RESULTS AND ANALYSIS

5.5.1 *Study 1: Fixed cyclic schedule*

The previous section proposed a maintenance scheduling formulation considering a fixed cyclic schedule that may be reused (every five years, in the example considered), assuming that the parameters and relevant network remain the same. This is usually not the case—commodity prices, cost and availability of manpower, reli-

abilities of equipment, and capacity expansion or business restructuring effects on site network structure form the primary reasons for this. However, the analysis of this particular model, along with an example network and a corresponding optimal schedule in this section provide valuable insights into maintenance operations. Subsequent sections elaborate on appropriate modifications under relaxed assumptions, and more realistic conditions.

The scheduling model described in Section 5.4 used a time discretization of one week, and was aimed at finding a five-year horizon cyclic turnaround schedule. This was found to be a suitable horizon, as this was an approximate least common multiple of turnaround frequencies for all units that required maintenance. This resulted in a discrete-time model with close to 200,000 variables ($\sim 4,500$ binary variables).

The model was built in GAMS 24.1.3, and was solved using the mixed-integer solver CPLEX 12.5. Most scenarios that were run were solved in under 200 seconds and achieved convergence within a $< 3\%$ relative gap. This result is satisfactory in the current context, as the confidence in the data is estimated to be of the same order. The achieved gap and solution time justify the formulation. The formulation is a discrete-time, material flow-based model.

The model is designed to serve as a tool to compare schedules under different scenarios—changes in network configuration, prices, availability of manpower, availability of intermediate products through imports, and so on.

For the case study that we present, we do not consider demands and assume that third party contracts exist whereby all material produced is sold. Thus, the optimizer drives the model towards the maximization of profit through the maximization of production volumes.

A sample output from the model is shown in Figure 5.4. The figure consists of a Gantt chart above and a bar graph below. The Gantt chart shows a schedule of which units are to be maintained in which weeks, and plots colored bars that denote the week in which turnarounds start and their widths denote the length of turnarounds. The bars corresponding to each unit on the Gantt chart are color-coded by the product envelope that the unit belongs to. The bar chart below plots how much manpower is used as a fraction of the total manpower availability, and the stacked bars are color-coded by how much of the available manpower units of a certain product envelope are using in each week. The two graphs are grouped as they

have a common time axis, which makes it easier to visualize the interaction between unit shutdown and the corresponding manpower resource usage. The thick vertical lines divide the time axis by year, and the faint dotted lines represent quarters in each year. Below these two charts is a network diagram, with identical network structure to that in Figure 5.1, with units belonging to the same product envelope colored similarly.

Figure 5.4 represents a potential schedule that is based on realistic data. For the potential schedule to be considered for transfer to practice, it must represent realistic and justifiable results. This is valuable both for validating the optimization framework and modeling approach, as well as to convince decision-makers of the capability of the tool.

The following points justify the potential schedule from Figure 5.4:

1. Most units belonging to the same envelope are scheduled for maintenance at around the same time—this is observed for blue (units 1–3), orange (units 9–11, 13, 14), and yellow (units 12, 15–17) envelopes. This makes sense as these units are close together in the plant network, and heavily depend on each other for raw material supply and product consumption. The performance of turnarounds on any one of these plants offers the opportunity to couple turnarounds with adjacent plants in the network, and the optimizer recommends which of these units could be coupled for better economics, all other factors considered.

2. The turnarounds of most units belonging to the red envelope (units 4–7) seem to be decoupled. This may be explained by two effects—significant inventory buffers relative to turnaround times, and the potential to buy and sell intermediate product from and to the market. These two features allow turnarounds on units in the envelope to take place without affecting upstream and downstream units.

3. Units 10 and 14 require the same raw materials and are adjacent in the plant network structure, but their turnarounds are decoupled. This can be attributed to the use of upstream raw material—when one of the units is undergoing a turnaround, the other may be able to take advantage of raw material supply without affecting upstream units.

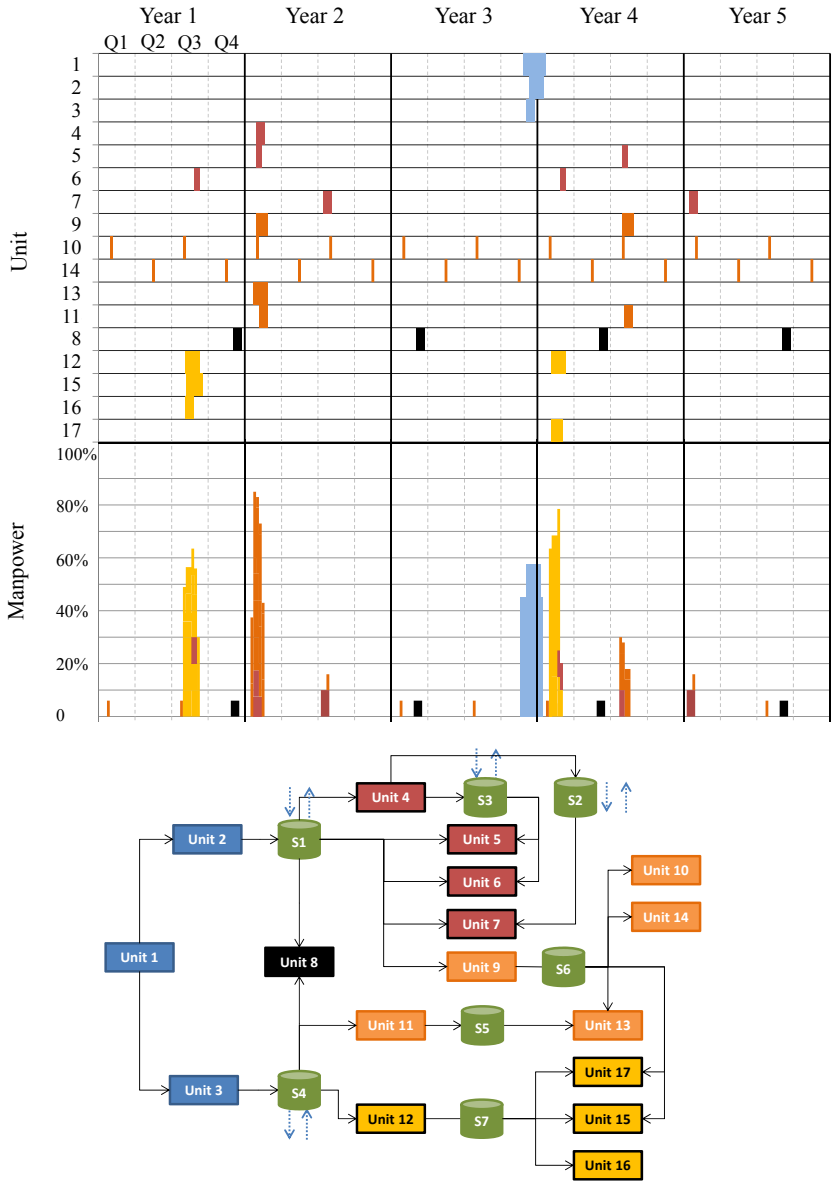


Figure 5.4: Sample schedule output from optimization model. A Gantt chart shows the turnaround schedule for each unit. The manpower utilization chart shows what fraction of the available manpower is engaged at each point in the time horizon. The site network shown below is color-coded by product envelope to emphasize the relationship between position of units in the network, and their relative turnaround schedules.

4. Unit 17 requires a large amount of manpower, and so its turnaround is decoupled from the main yellow envelope turnaround in Year 1, Quarter 3 in order to respect the maximum available peak manpower. In this case, the optimizer recommends performing maintenance on Unit 17 when Unit 12 from the yellow envelope undergoes its second turnaround in the five-year cycle. An intuitive explanation for this is to ensure that Unit 17 undergoes maintenance when some other part of the yellow envelope has to undergo maintenance as well (Unit 12's second turnaround), so as to make opportunistic use of the disruption in the site network caused by Unit 12's maintenance.

5. The particular scenario that was run included increased feedstock prices during Q1 of each year, a situation commonly seen in practice with some raw materials. We choose to decrease relative margins by 10% to simulate this scenario. The schedule responds to this by recommending a large number of the turnarounds to be performed in Q1 as seen in the case of the red, blue, yellow as well as orange envelopes. In this way, feedstock is preferentially purchased in Q2, Q3, and Q4 when prices are lower and most units are running.

6. The different envelopes have their turnarounds scheduled in different quarters, and in different years as well. From a manpower availability as well as a financial viewpoint, this is favorable. In terms of the site reliability, this is favorable as well—it may not be desirable to have most turnarounds happening in a short span of time, as individual plant reliability, and more significantly, average plant reliability, would be at its lowest across the board just prior to these turnarounds. The financial performance over the five-year schedule shown in Figure 5.4 is illustrated in Figure 5.5. In the figure, the left vertical axis corresponds to weekly profits, and the right vertical axis corresponds to quarterly profits. The top horizontal axis denotes time by quarters, while the bottom axis does it by weeks. As can be seen, turnarounds can increase the variability of profit margins over time. Specifically, the sets of turnarounds around week 30 and week 162 have an effect of the smoothness of the profit profile. The fact that these sets are in different years helps to some extent, and it is intuitive that the appropriate spreading of turnarounds over the time horizon is important in order to reduce the variability in profits over time. A systematic way to do this is elaborated on in Section 5.5.2.

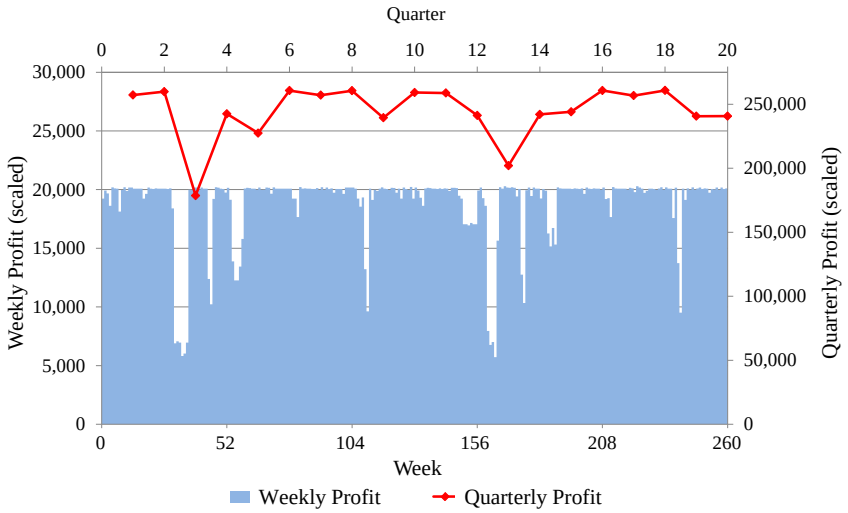


Figure 5.5: Weekly and quarterly financial performance of an optimal schedule

Lack of import capability of intermediate products The model proposed assumes that some intermediate storage tanks are able to interact with the market, and import raw materials when required. As we see with the analysis of the schedule proposed in the first part of the Results section, this allows us to decouple the turnarounds in the red unit envelope, when the price of these intermediate products is favorable. When there is inability to import or buy raw materials from the market, one can see that this may in turn affect the best choice of turnaround coordination. This is illustrated in the optimal schedule without import in Figure 5.6. The optimizer recommends that the turnarounds of Units 4, 5 and 7 now be aligned. Unit 6 is able to draw from the inventory in S3 when Unit 4 is down. This choice of alignment is intuitive, as there is no longer the possibility to import intermediate product and use this to decouple the turnarounds of the red envelope.

What we see here is the effect of supply of raw material on the optimal schedule, and this sort of analysis is important to do when one is not completely sure of being able to import or buy raw material whenever required. The use of one or the other schedule can be traded off using the difference in estimated profits from the two scenarios.

The discussion so far has been on the analysis of a specific schedule, under a given set of assumptions and constraints. The following subsections describe the generation of multiple possible schedules and analyze solutions in alternative scenarios.

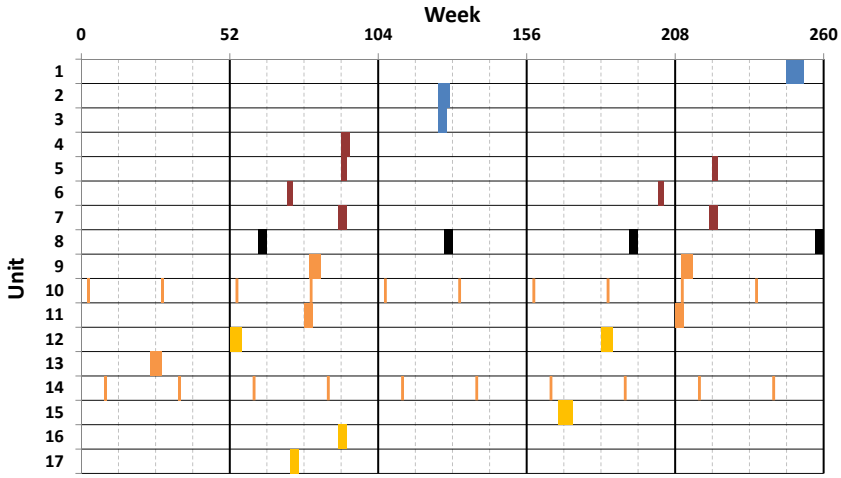


Figure 5.6: Schedule when no import of intermediate material is possible

5.5.2 *Study 2: Rolling horizon framework*

The cyclic scheduling model serves to illustrate (1) the nature of the schedules generated from which we can gain some insight into network behavior; and (2) the tractability of an industrial-size problem under such a framework. However, there are several limitations of the cyclic scheduling framework in the context of maintenance scheduling with a chemical site network over long-term horizon. In this study, we pose the scheduling problem within a rolling horizon framework rather than using a fixed cyclic schedule. We investigate each of the limitations of the cyclic scheduling framework, and demonstrate that the rolling horizon framework provides a natural way to address them.

A rolling horizon framework respects the regular network and turnaround constraints, but is solved to a much longer horizon, say 15 years, called the planning horizon (Stadtler 2000). However, only the decisions that correspond to an initial horizon called the decision horizon, say 5 years, are implemented (Figure 5.7). Then, the relevant information from the ending state (gathered from the period denoted by the thicker red line) which involves inventory levels and the time since the last maintenance of each of the units is used as the starting state for this next iteration. In this way, the rolling horizon scheme attempts to incorporate the effect of both future turnaround decisions as well as demand and price forecasts on short-term decisions.

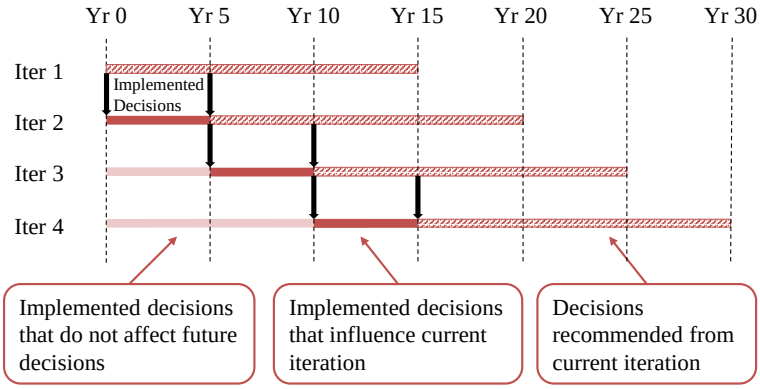


Figure 5.7: A potential rolling horizon scheme

The horizon for the cyclic schedule forms the natural cycle. We use the suggestion by Baker (1977) that the planning horizon be at least as long as the natural cycle.

The horizon for the fixed cyclic schedule is determined by the least common multiple of all the turnaround frequencies for the units, so that each unit is maintained exactly f_i time periods after it was maintained before. To determine a suitable horizon for a moving horizon problem in this context, apart from choosing a horizon that is tractable and for one which there are reasonable forecasts, one has to consider the situation where the turnaround for a unit that is maintained only once in a cycle gets pushed to the end of the horizon and include the costs incurred for the maintenance beyond the time period as well. To include these considerations, we consider a horizon of 15 years and three months (61 quarters, 793 time periods)—about thrice the length of the original horizon.

In terms of constraints, instead of Equations 5.19 and 5.20, which enforce that a unit undergoes maintenance exactly once between two time intervals that span its turnaround time period. Instead, we write

$$\sum_{\tau=t}^{t+f_i-1} y_{i,\tau} \geq 1 \qquad t \in T,\, i \in M. \tag{5.25}$$

This equation says that between any two time periods that span the turnaround time period of Unit i, the unit has to undergo at least one turnaround.

Another consideration is the last time each of the Units underwent maintenance. To account for this, a constraint of the form

$$\sum_{\tau=t}^{t+f_i-L_{i,t}-1} y_{i,\tau} \geq 1 \qquad i \in M,\, t \in T, \tag{5.26}$$

is added, where t here is the current time, and $L_{i,t}(= L_{i,0} + t)$ is the number of weeks that have elapsed since the last turnaround that took place prior to the time span considered in the model of Unit i at time t. Note that Equation 5.26 becomes inactive for $f_i \leq L_{i,t}$ and makes Equation 5.25 redundant when $f_i - L_{i,t} > 0$, which may correspond to the first few time periods in the horizon. The corresponding redundant equations may be removed.

To ensure that the binary variables that represent the time periods just after the start of a turnaround are turned off for the length of the corresponding turnaround, we enforce

$$\sum_{\tau=t}^{t+n_i-1} y_{i,\tau} \leq 1 \qquad t \in T,\, i \in M. \tag{5.27}$$

This represents a tight formulation of the required condition.

In principle, the longer the horizon one is able to solve for, the better the incorporation of this information. As solving an infinite horizon is intractable, one needs a way of obtaining near-optimal short-term scheduling decisions by approximating the infinite horizon by a 'long enough' horizon. In this situation, one typically has to be concerned with end effects, or terminal constraints. We enforce that final inventories in tanks be the same as the initial inventories. In the current context, we anticipate that a fifteen year horizon, apart from being subject to uncertainties that are unaccounted for, is long enough to plan around a reduced supply at the tail end of the moving horizon.

To compare the rolling horizon formulation with the cyclic schedule, we run the rolling horizon formulation for four iterations to obtain 20 years of estimated profit and calculate the average profit over five years using the scheme in Figure 5.7. We assume that raw material and product prices vary on an annual cycle for purposes of comparison in Table 5.6. We expect the rolling horizon schedule to be at least as good as the cyclic schedule under the above assumptions, and we see a 1.36% improvement in profit with the rolling horizon schedule. Even though we do not

report zero-gap solutions, we are able to find a rolling horizon solution that is 0.6% better than the best possible solution for the cyclic schedule. A 0.6% improvement in profit is significant, given the scale of the problem that we are looking at, and this comparison further justifies the use of the rolling horizon formulation in the current context.

Table 5.6: Comparison of profit from cyclic schedule and rolling horizon schedule

Schedule	Avg. profit units	Relative gap	Time to solve (s)
Cyclic	2,564,801	0.7%	36
Rolling horizon	2,599,788	0.4%	1219 (4 iterations)
Improvement	**= 1.36%**		

For the network studied, the rolling horizon formulation had approximately $16,000$ binary variables and $600,000$ continuous variables at each iteration. We are able to solve the problem to within a 1% relative gap without changing the granularity of modeling detail for all time periods.

Peak manpower availability and financial performance As mentioned in the previous section, the effect of turnarounds on financial performance is significant. So, as long as long-term profit estimates are not affected very much, there is a trade-off to be made between a balanced financial performance and identifying synergistic opportunities to perform maintenance across multiple plants simultaneously. In fact, one may formalize this importance by explicitly requiring that profits do not fluctuate greatly across time periods. One way to do this would be to enforce that profit in each time period is greater than some fraction (say 90%) of the average profit estimate across periods. If we consider Q_{tot} periods, the profit estimate is given by

$$z_j = \sum_{t \in Q_j} \Bigg[\sum_{i \in P} p_{i,t}\, x_{i,\infty,t} - \sum_{i \in M} m_i\, y_{i,t} \sum_{\tau=t}^{t+n_i-1} c_{m_{i,\tau}} - \sum_{i \in B} c_{r_{i,t}}\, x_{0,i,t} - \sum_{i \in I \setminus F} h_{i,t} s_{i,t} \Bigg] \qquad j \in N_Q, \tag{5.28}$$

where $N_Q = \{1, 2, \ldots, Q_{\text{tot}}\}$ is the set of periods and Q_j represents the set of time intervals in the jth period. We add the constraint

$$z_j \geq \frac{\lambda}{Q_{\text{tot}}} \sum_{i \in N_Q} z_i \qquad j \in N_Q, \tag{5.29}$$

which ensures that the profit in each period is at least some fraction λ of the average profit across periods. For the particular instance we study, we vary λ from 0.93 to 0.99 and provide the results in Figure 5.8. We see that beyond $\lambda = 0.96$, the overall profit estimate begins to decrease. This means that beyond this point, we have no way to further smooth out the profit profile without decreasing the overall profit, i.e., causing some scale-backs in production to ensure this. This emphasis on balancing financial performance may be thought of as a secondary objective, and the corresponding constraint as an ϵ-constraint that is useful for trading off the two objectives.

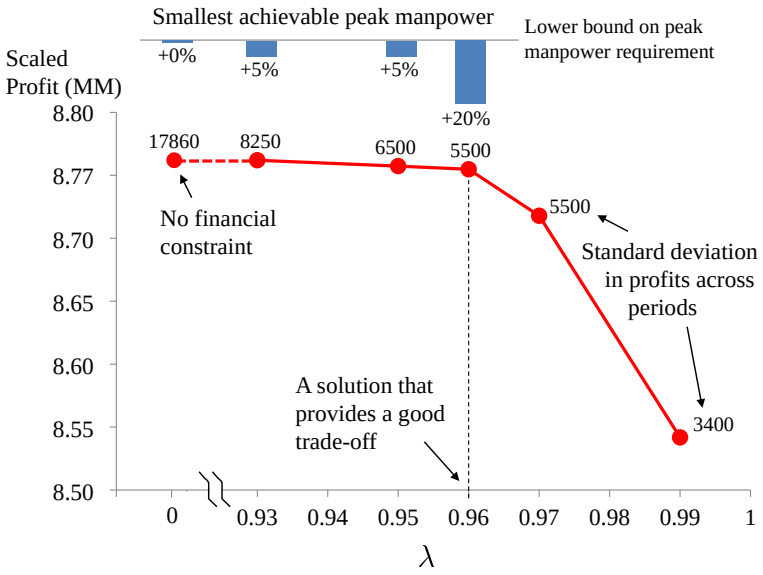


Figure 5.8: A Pareto curve that trades off overall profit vs. coefficient in Eq 5.29, λ. The numbers above each data point represent the standard deviations in profit over the horizon. The bar graph above each data point denotes the least peak manpower achievable for each of the scenarios. This is represented as a percentage increase from the lower bound on peak manpower.

An additional objective is to try and reduce peak manpower. As mentioned in Section 5.4, the ability to reduce peak manpower, i.e., the highest manpower requirement at any time period in the horizon, without affecting long-term profit

margins not only allows for greater bargaining power with firms that hire out manpower, but gives the opportunity to bid for the more productive and skilled fraction of the available manpower. We consider balancing financial performance as higher priority than reducing peak manpower. So, for each λ that signifies a different level of requirement on financial performance, we try and reduce the peak manpower. We see that the solution for which $\lambda = 0.96$, the least peak manpower acheived is 20% higher than the lower bound on peak manpower. The lowest possible manpower requirement corresponds to the unit that requires the largest workforce at all times during its turnaround.

Calculation of big-M parameters The big-M constraints in Equation 5.21 derive the big-M value in a natural way, i.e., from the upper bounds on flowrates in the given network structure. However, on closer examination, one may see that these may not necessarily be the tightest valid upper bound that one could supply. This is observed in two ways: (1) The upper bound on the flow out of a unit may be greater than the upper bound on the flow into it; and (2) The stoichiometric ratio of inputs/outputs of a unit may not be accurately reflected by the corresponding upper bounds. This is illustrated in Figure 5.9. For instance, a maximum of 24.36 units enter Unit 9, but a maximum of 25.16 is allowed to exit it, which is clearly a loose bound. Another example is the input streams to Unit 5; the ratio constraint enforces $x_{S1,5,t} = 3.53\, x_{S3,5,t}$. The physical maximum flow rate throught the pipe connections is $x^U_{S1,5} = 17.18$, whereas the ratio constraint will allow only a maximum of $3.53\, x^U_{S3,5} = 3.53 \times 3.43 = 12.10$ units. The inclusion of these updated upper bounds in the big-M constraints is useful in improving the LP relaxation, and the effect of this is illustrated in Table 5.7. The solver we experiment with, CPLEX, has the capability of identifying and improving such bounds, but we observe that their explicit inclusion helps to speed up the solution by up to 45%.

The observation that reducing the 'big-M' in the flow constraints motivates us to look for a systematic way to discover all such connections in the network. One straightforward method to do this is to solve a linear program for each arc—where we maximize the flow through the arc subject to unit ratio constraints, nonnega-

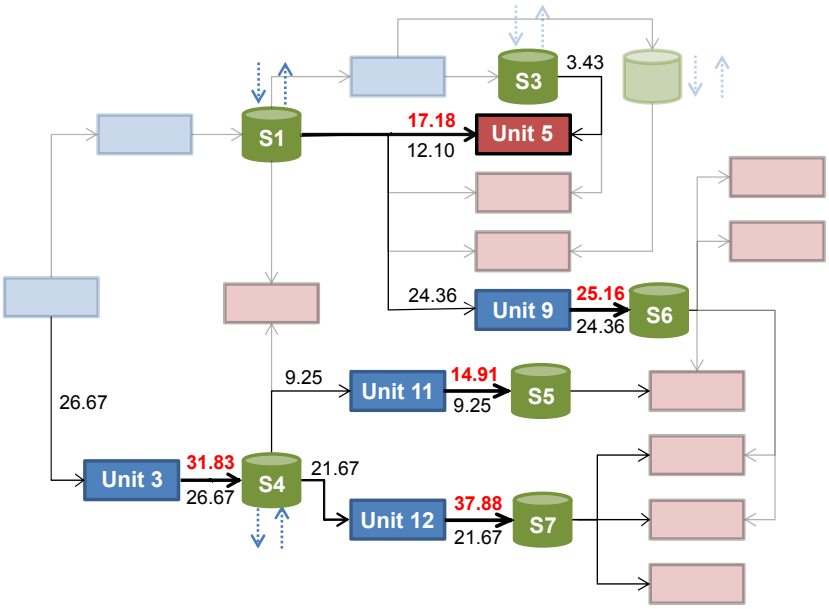


Figure 5.9: Example network highlighting which flows were modified to improve big-M constraints

Table 5.7: Cumulative effect of updated big-M constraints on solution time

Original network	354	
Modifications	Solve time(s)	% improvement
x^U_{3,S_4}, x^U_{12,S_7}	315	11.0
$x^U_{S_1,5}$, x^U_{9,S_6}	235	33.6
$x^U_{11,S5}$	194	45.2

tivity constraints, valid upper bounds on all the other arcs, and modified inventory balances,

$$\sum_{i \in I} x_{i,j} \leq s_j^{\Delta} + \sum_{i \in I} x_{j,i} \qquad j \in I, \tag{5.30}$$

$$\sum_{i \in I} x_{j,i} \leq s_j^{\Delta} + \sum_{i \in I} x_{i,j} \qquad j \in I, \tag{5.31}$$

where $s_j^{\Delta} = s_j^U - s_j^L$. If the flow does not reach the current upper bound, we see that there is potential to tighten the big-M for the corresponding flow constraint in the mixed-integer scheduling formulation. The number of arcs in such a network would not exceed a few hundred, at most, and the solution of one LP per arc is is a relatively inexpensive procedure. The procedure guarantees that the tightest possible upper bounds are found for each arc as the LPs respect all of the other network flow considerations, and is used to generate tighter bounds on the flows (Table 5.7).

Rescheduling under disruptions An essential part of maintenance activities is the effect of disruptions on turnaround schedules. Discovery work from taking down a plant for a turnaround may yield the information that the shutdown may take several weeks more than originally anticipated. In this scenario, one may consider reconstructing the schedule from that point in time going forward, to see whether there may be other opportunistic benefits of performing other maintenance tasks at that time. Other major reasons to reschedule would be to potential savings from using a new schedule that takes into account the current state of inventory and maintenance, updated cost and price estimates, changes in the site network structure, and the site reliability situation.

Multiple kinds of turnarounds on a plant In some situations, a plant may need to undergo different kinds of turnarounds. The first kind of multiple turnaround scenario we address is when every second or every third turnaround on a particular plant possibly requires a different duration of shutdown and a different resource requirement. As a hypothetical situation, Unit 1 may need to undergo complete overhaul for six weeks and requiring 1000 personnel every five years. However, between two major turnarounds, it may require a smaller three-week turnaround

requiring 300 personnel. To model this, we use $Y_{i,t}$ as the Boolean variable that signals the start of a major turnaround, and the associated binary variable $y_{i,t}$. Similarly, we use $W_{i,t}$ and $w_{i,t}$ for the minor turnaround. The logic

$$Y_{i,t} \implies W_{i,t} \vee W_{i,t+1} \vee \cdots \vee W_{i,t+f_i-1},$$

ensures that there is a minor turnaround within f_i time periods of a major turnaround, and this is enforced for all t. This logic is easily converted into the inequality constraints

$$\sum_{\tau=t}^{t+f_i-1} w_{i,\tau} \geq y_{i,t} \qquad t \in T,\, i \in M, \tag{5.32}$$

where $w_{i,t} \in \{0,1\}$. Similarly, we enforce a major turnaround following a minor turnaround through

$$\sum_{\tau=t}^{t+f_i-1} y_{i,\tau} \geq w_{i,t} \qquad t \in T,\, i \in M. \tag{5.33}$$

The constraint

$$\sum_{\tau=t}^{t+2f_i-1} y_{i,\tau} \geq 1, \qquad t \in T,\, i \in M \tag{5.34}$$

ensures that major turnarounds have to take place at least once every $2f_i$ periods.

The other multiple turnaround scenario we consider is when there is a secondary turnaround, which is completely decoupled from the primary one (i.e., there is no requirement that this turnaround is to take place within some time of the primary turnaround). This may happen, for instance, when some parts of a plant need to be maintained at a completely different frequency from those parts undergoing maintenance through the primary turnaround. This is modeled by adding the constraints

$$\sum_{\tau=t}^{t+f_{M2_i}-1} v_{i,\tau} \geq 1 \qquad t \in T,\, i \in M_2, \tag{5.35}$$

where M_2 is the set of units that have a secondary turnaround and $v_{i,t} \in \{0,1\}$ represents the binary variable that represents when the unit is shut down for the secondary turnaround.

The corresponding big-M, manpower, and minimum operation level constraints as well as the objective function can be modified accordingly. Constraints on turnarounds similar to 5.27 can be written for the cases described by Equations 5.32–5.35.

Transitioning into a schedule A significant concern when implementing a schedule over a long term horizon as is the case in this paper, is the question of how to put it into motion. In the case of a brand new starting state, one may use the cyclic schedule formulation to begin with. In this case, a natural desire is to push the turnarounds of each of the units as much as possible to take advantage of the fact that the equipment have undergone relatively less wear and tear. For example, units that have to be maintained just once in the five year horizon have their turnaround at the end of the cycle. Figure 5.6 illustrates a schedule that follows this to some extent. Unit 1 undergoes maintenance only in Week 247, and Unit 8 in Week 62. Fewer turnarounds are scheduled for the entire first year, with just the units that have a turnaround time period of less than one year undergoing maintenance. However, one may also choose to use the rolling horizon formulation in this context.

In the case of an existing site transitioning into a new schedule, the rolling horizon formulation would apply directly. The setting is almost the same—the formulation accounts for the last time a unit was maintained along with current inventory levels, and plans its future turnarounds accordingly.

Seasonal constraints An important practical concern for turnaround scheduling is the productivity of manpower. For sites in locations which can have particularly harsh weather, there may be an entire season where it is desired that no turnarounds occur. This is a situation where a cyclic schedule may be particularly restrictive. In the cyclic schedule illustrated in Figure 5.10, for instance, the fourth quarter is constrained to have no turnarounds. However, if Unit A is scheduled to undergo a maintenance turnaround once every 1.25 years, or four times in the five year horizon at fixed intervals in a cyclic schedule, it is clear from the figure that its scheduled turnaround (which would occur within the intervals shown in red) would occur in the fourth quarter of one of the years in the horizon.

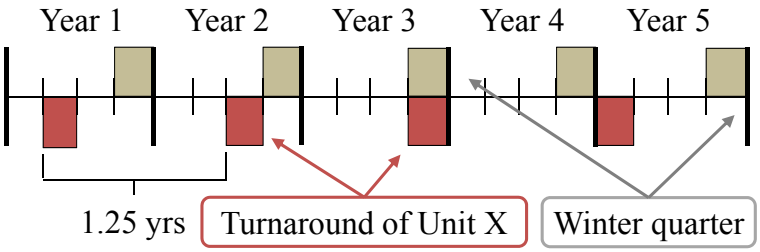


Figure 5.10: Illustration of inability of cyclic schedules in dealing with seasonal constraints

The rolling horizon formulation would face no such difficulties (assuming that no turnaround needed to occur more than once a quarter in the example above), and one would simply enforce

$$y_{i,t} = 0 \qquad i \in M,\, t \in \Gamma, \tag{5.36}$$

where Γ is the set of time periods where turnarounds must not occur.

Approximation of turnaround time periods In the cyclic schedule, it can be observed the the least common multiple of all the units turnaround time periods turned out to be five years. If, for instance, a unit had to be maintained every 1.5 years and we still wanted to retain a five-year horizon, then we rounded down this number to 1.25 years, so as to enforce four turnarounds every five years. This is a significant change, as each turnaround incurs a large cost. This approximation can be completely dispensed with in the rolling horizon formulation by ensuring that the horizon one solves for at each iteration is just a big enough multiple of the largest turnaround time period among the units.

As can be seen from the formulation, the main disadvantage of the rolling horizon approach is primarily from a computational standpoint, in that there is a significant growth in the complexity of the model. This results from both a direct increase due to the consideration of a much larger number of time periods, as well as the transition to flexibility in time between turnarounds on each unit as opposed to a fixed difference for the cyclic schedule, i.e., moving from equality to inequality constraints. However, the significant benefits from considering a rolling horizon schedule are clear from the above discussion.

5.6 CONCLUSIONS

The work demonstrated in this paper outlines a methodology for performing long-term turnaround optimization of continuous chemical plants that are part of a large integrated site. The problem is first motivated by discussing the large financial impact that maintenance tasks induce on integrated sites, and then the potential for optimization is illustrated using an industrial-size example site network.

We demonstrate the generation of potential schedules using mixed-integer programming technology through both cyclic and rolling horizon scheduling formulations. A thorough analysis of an output schedule is performed to reflect domain knowledge, and a number of potential analyses, including financial and sensitivity studies, are presented as capabilities of the modeling framework. The three major decisions that fall out of the framework are the: (1) maximization of net present value; (2) balancing of financial performance; and (3) reduction in peak manpower. Real-world issues such as the incorporation of seasonal constraints and the possibility to import intermediate products are discussed. Practical ways to screen input data to tighten the formulation, as well as a framework for modeling multiple kinds of turnarounds is put forth.

A potential extension of the work includes integrating this long-term maintenance strategy with a short-term scheduling model that considers production and inventory planning, and models maintenance tasks such as ramping up and down of plants and different kinds of manpower for different tasks. Another possible extension of this work is to complement the optimization model with a discrete event simulation. The simulation could be built to include variation in production, length of maintenance, failure rates of components, and could be used to validate the optimization solutions obtained through the mixed-integer model.

6

MEDIUM-TERM TURNAROUND PLANNING FOR INTEGRATED CHEMICAL SITES

6.1 INTRODUCTION

Chapter 5 introduced the turnaround planning problem, and illustrated the benefits from considering site-wide maintenance in an integrated manner. In this chapter, we deal with the problem of medium-term turnaround planning under uncertainty for integrated chemical sites.

With regard to turnarounds, there is uncertainty in what tasks need to happen, in the delivery of materials, as well as in weather conditions. Due to this, a major portion of the uncertainty lies in the duration of the turnaround, which affects the availability of maintenance manpower as well.

To quote Lenahan (1999), *there are only two types of work on a turnaround, routine and unexpected. If the routine is under control there is time to deal with the unexpected but if the routine becomes unexpected the unexpected may become catastrophic.*

In some sense, Chapter 5 addressed the weather condition issue by providing schedules that worked around specific undesirable times of the year due to weather conditions. Further, when plant units are taken off-line for maintenance turnarounds, a more accurate assessment of the state of the equipment is performed, which is termed as 'discovery work'. Discovery work may reveal that certain pieces of equipment are damaged worse than expected, resulting in an increase in maintenance duration that could even be on the order of weeks. Uncertainty in the delivery of materials also leads to a delay in the duration of a turnaround. Consequently, maintenance personnel, who are typically hired on contract several months in advance,

and the duration for which they are required to be made available is an important issue.

As far as we know, no prior work appearing in the literature addresses what we call the medium-term turnaround problem for integrated chemical sites, let alone include the consideration of uncertainty in this context. As mentioned in Chapter 5, most prior work deals with short-term scheduling of production and maintenance (for example, Dedopoulos & Shah (1995a); Sanmartí et al. (1997); Pistikopoulos et al. (2001); Megow et al. (2011)).

Very few papers exist on process scheduling under uncertainty (Lin et al. 2004; Li et al. 2012; Wittmann-Hohlbein & Pistikopoulos 2013), and fewer still where discrete-time formulations are considered, or on maintenance considerations under uncertainty. Vujanic et al. (2012) address the robust optimization of cement plant operation, where the uncertain parameter is in the time of required reserve dispatch in the context of energy storage. A review of work in the process scheduling under uncertainty area is covered by Li & Ierapetritou (2008).

In Section 6.2 we motivate the need for medium-term maintenance scheduling, and the nature of the uncertainty considered in the problem. Section 6.3 introduces two different modeling approaches to handle the uncertainty, in order to make scheduling and production planning decisions. Section 6.4 compares and contrasts the two approaches in terms of solution quality and solution time. In addition, several aspects of the production and inventory plan, turnaround schedule, and manpower utilization and availability are analyzed.

6.2 MOTIVATION AND PROBLEM DESCRIPTION

Chapter 5 considers maintenance scheduling over a time period that spanned multiple years, and recommended solutions usually involved a set of plant units undergoing their turnarounds with or close to each other. The medium-term scheduling problem deals with refining schedules recommended by the long-term scheduling model 6–9 months in advance of a particular set of unit turnarounds. There are two primary reasons that considering the problem on such a time scale is desirable: (1) Maintenance personnel are typically hired on contract, and contract negotiation for quantity, type, and duration of manpower requirement typically begins several months prior to turnarounds; and (2) the build-up of downstream inventory, to deal

with the satisfaction of product demand during the extended periods of interrupted production that turnarounds create, is known to take several weeks or even months. This provides motivation to investigate the medium-term maintenance scheduling problem, to simultaneously perform turnaround scheduling while ensuring availability of manpower of different skill sets and perform production planning across the integrated site network in preparation for disruptions in flow.

In summary, the main decisions to be made are the start times of each of the turnarounds, and the production and inventory decisions within the network. When considering these factors, we would like to incorporate the effect of uncertainty in our models. In this work, we focus on uncertainty in the duration of turnarounds.

From a planning perspective, several issues arise when maintenance turnarounds take longer than the nominal duration. The main issues we consider are the availability of maintenance manpower as well as the production and inventory levels over time across the entire site network. These two issues affect planning in two quite different ways.

As mentioned above, maintenance personnel are typically hired on contract several months in advance for a specific time period. If maintenance exceeds the nominal duration and maintenance personnel are not available, maintenance could be extended indefinitely and could have severe impacts on production, especially in a tightly integrated site. Replacement or on-demand manpower is usually not an option due to scarcity, so there is no immediate recourse action that can be effected. As a result, the availability of manpower for the nominal duration as well as for an extended duration is paramount. Having said this, it is extremely unlikely that all the turnarounds occurring in the time window concerned are affected. These characteristics indicate that we would like to have a model that is robust (Ben-Tal et al. 2009) to uncertain turnaround durations with respect to manpower availability. This is discussed further in Section 6.3.1.1.

An example of the need for analyzing this uncertainty more closely is illustrated below in Figure 6.1. From the point of view of production levels and demand satisfaction, the ideal situation may be, for example, to perform all of the maintenance work in the shortest time span possible in order to restore the site to normal operation quickly. However, from a robust perspective, possible delays and consequent manpower scarcity indicate that a provision for buffer time between scheduling tasks may result in more reliable maintenance turnaround times.

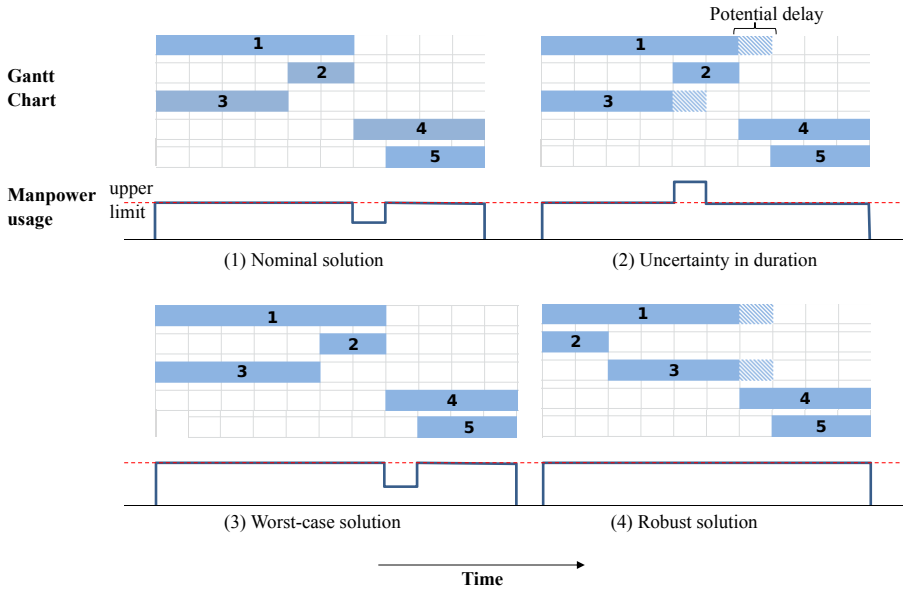


Figure 6.1: The effect of the manner of consideration of uncertainty on scheduling decisions and manpower requirement

Here, case (1) corresponds to when the assumption of exact maintenance durations is made. The optimal solution is illustrated, which keeps maximum manpower utilization at 100% of the available manpower while minimizing the total duration. Case (2) shows that delays are possible in Units 1 and 3, resulting in a potentially infeasible schedule where delays could force the available manpower limit to be exceeded. The worst-case solution is illustrated in case (3), where we just assume that all tasks take the upper bound on maintenance duration, and the figure shows that the schedule is feasible in this case, although we have increased the total duration of the turnarounds. Case (4) makes the argument that case (3) is too conservative, and that all random variables taking their worst case realizations is not realistic. Case (4) attempts to make a trade-off by allowing a certain number of tasks to exceed their expected duration time (here this is one task). Here, the schedule reflects this by rearranging the tasks to retain a good total duration, while allowing for either Unit 1 or Unit 3 to take longer than expected maintenance durations.

The other factor being affected is the set of production and inventory decisions to be made over time. Production down-time due to turnarounds can be managed by making use of production and inventory capacity to plan around turnaround periods.

Turnarounds are also often scheduled to take place during periods of lower demand, and this helps to alleviate the demand satisfaction issue as well. Uncertainty in the duration of a turnaround directly affects production and inventory levels, and appropriate planning of these is required to maximize some sort of profit objective, be it risk-neutral (expected value) or risk-adjusted (Rockafellar & Uryasev 2002). Unlike maintenance manpower, recourse actions may be taken on production and inventory once uncertainty has been realized. These characteristics suggest that stochastic programming (Birge & Louveaux 2011) may be a suitable approach from a profit maximization point of view.

The following sections investigate two approaches to reconciling these two aspects of the uncertainty. The first approach involves a **sequential** two-step procedure. First, a robust scheduling formulation is built which minimizes the makespan of the entire set of turnarounds, and all such solutions are found. Next, these recommended schedules are fixed, and a multi-stage stochastic linear programming model is solved that performs production planning, where each stage corresponds to the time interval between the start of adjacent turnarounds. For example, the first stage decisions are production planning decisions for the first six months. The first turnaround occurs at the six-month time point, and indicates the start of the second stage.

The second approach we look at involves a combined robust and two-stage mixed-integer stochastic program, which **simultaneously** schedules the turnarounds and performs production planning. The first-stage decisions consist of scheduling the turnarounds, which actually occur in subsequent stages, and the production planning decisions for the first stage (the first six months, for example). Here, an approximation is made, in that the duration of all turnarounds is assumed to be revealed at the start of the second stage. Following this, the recommended solution is embedded in the multi-stage stochastic linear program to refine the production plan.

The two schemes are shown in Figure 6.2.

Treatment of manpower Manpower may be grouped by craft, or by which contractor hires them out.

Manpower may also be classified based on tasks, where they may be grouped as a crew set with complementary skills. Three types of jobs normally occur during

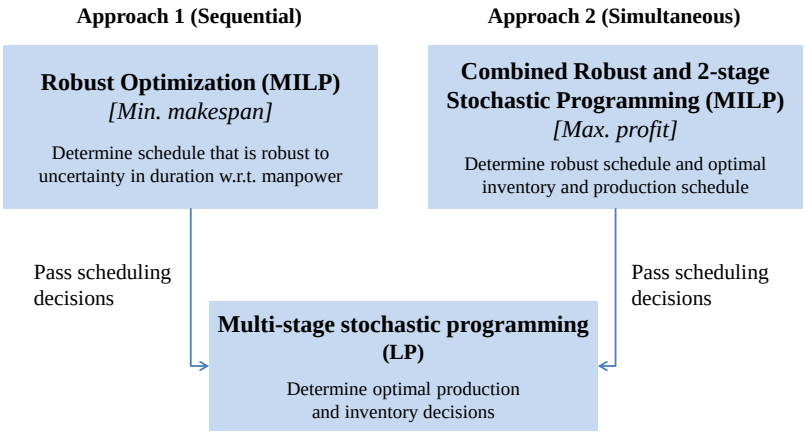


Figure 6.2: Proposed approaches for combined robust optimization and stochastic programming

a turnaround—major tasks, minor tasks, and bulk work. Major tasks may include time-consuming items such as distillation column re-traying, or project work such as installing a new control system, and these are usually planned up to the last detail, although progress can be beset by delays in delivery of equipment, for instance. Here, manpower with specialized training may be required.

Minor tasks involve items such as cleaning of heat exchangers, or inspection and repair of process vessels. Bulk work consists of the overhaul of a large number of equipment such as valves and small pumps. In both these cases, maintenance personnel may have transferable skill sets that allow them to work on miscellaneous minor tasks and bulk work. Here, delays may happen due to discovery work, or underestimation of variation in task duration during the planning phase.

We assume that an appropriate classification of manpower has been made, and that the different types of manpower considered are required through the nominal duration of the turnaround. Minor tasks and bulk work comprise of monitoring, testing, and attending to various pieces of plant equipment, each taking a portion of the entire turnaround time. Even if subtasks, and hence manpower requirements, are sequential for each piece of equipment, there are several such equipment units, and therefore the manpower is required more or less through the entire turnaround. Any remaining down-time due to other maintenance tasks being completed may be capitalized on to perform additional tasks. Short-term maintenance scheduling deals with details of specific crews and their respective shift scheduling, and this is beyond our current scope.

Note that as we focus on weekly time scales, we do not consider start-up and shutdown times, which typically take on the order of 2–3 days.

In the following sections, we discuss the formulations involved, the solution times, compare the solutions we obtain from the two approaches, and illustrate the various analyses that the model outputs enable.

6.3 MODEL FORMULATIONS

6.3.1 *Sequential robust scheduling and production planning*

6.3.1.1 *Model formulation for robust maintenance schedule generation*

This section deals with formulating a model to determine robust turnaround schedules. As mentioned above, it is typical to hire manpower on contract, and we assume that manpower requirements—which involves the quantity, skill, and specific time window of requirement—are required to be fixed six months ahead of time. However, the exact duration of a maintenance task may not be known ahead of time. This duration is often revealed only when plants are taken down for maintenance, through discovery work or an update on equipment delivery time. This may also be termed corrective maintenance, as it is not a task that is scheduled to be under the planned maintenance, but an item that may be acted upon due to unexpected deterioration in the equipment or other delays.

Through prior analysis, historical data, or expert opinion, one may ascertain a reasonably accurate window within which this uncertainty in duration may fall. In addition to the fact that it may be very unlikely in practice that all turnarounds occurring in the considered time window require corrective maintenance through discovery work, a maintenance policy may be enforced where only a pre-defined number of tasks deemed as most urgent or resource-intensive are put in as work orders. The remaining tasks may be deferred to future preventive maintenance, or may not affect operation as much and it may be preferred to run them down to breakdown maintenance. The nature of these considerations allow us to use the concept of budget uncertainty (Bertsimas & Sim 2004; 2003), where we ensure manpower availability for up to a pre-defined number of extended turnarounds.

For the formulation we use, notation is listed in Table 6.1.

Table 6.1: Set, parameter and variable notation for robust scheduling model

Entity	Set
Set of units to be maintained	M
Set of time periods	T
Set of different manpower skill types	K
Parameter	**Notation**
Nominal turnaround duration	d_i
Maximum extension of turnaround duration	p_i
Maximum number of turnarounds allowed to exceed nominal duration	Γ_t
Maximum number of units requiring manpower type k for the extended duration ($\leq \Gamma_t$)	$\gamma_{t,k}$
Manpower requirement during nominal duration	$q_{i,k}$
Manpower requirement during extended duration	$q^e_{i,k}$
Maximum available manpower of type k	$\widehat{m}^{\max}_{k,t}$
Maximum total allowable manpower	$m^{\max}_t$
Variable	**Notation**
End time period of last turnaround	T_{end}
Binary representing start of maintenance	$z_{i,t}$
Binary for maintenance during nominal duration	$y_{i,t}$
Binary for maintenance during extended duration	$u_{i,t}$

An objective that attempts to minimize makespan may be most suitable, as we would like to restore the site to full working condition in the minimum possible time. Another reason for this may be that the remoteness of the site location may require that maintenance crews be transported there. In this case, the crew may incur a cost for the entire duration of travel, which one may approximate to be the makespan of all the tasks.

$$\min \quad T_{\text{end}}, \tag{6.1}$$

where T_{end} represents the time at which the last maintenance ends. We assume, without loss of generality, that maintenance starts at time $t = 1$.

Here, it is interesting to note that the uncertain parameter is the duration of the turnaround, a parameter that has the unit of time, an entity that normally appears as an index in a discrete-time formulation. However, the framework of Bertsimas & Sim (2004; 2003) is geared to handle parameter uncertainty in the left-hand side coefficients and right-hand sides of the constraints. We describe how we transform this into an uncertain parameter in the sense of Bertsimas & Sim (2004) below.

We have a binary variable $z_{i,t}$ representing the starting turnaround period for unit i. We also introduce binary variables $y_{i,t}$ that represent the fact that a turnaround is taking place on unit i at time period t. Further, we introduce another binary variable $u_{i,t}$ that represents the period spanning the extended duration for a particular unit.

To ensure that maintenance for a particular unit takes place only once in the duration under consideration, we have

$$\sum_{t} z_{i,t} = 1 \qquad \forall i \in M. \tag{6.2}$$

We then enforce that the variables $y_{i,t}$ and $u_{i,t}$ take the appropriate values, we have

$$z_{i,t} \leq y_{i,t+k_i} \qquad \forall i \in M, t \in T, k_i = \{0, \ldots, d_i - 1\}, \tag{6.3}$$

$$z_{i,t} \leq u_{i,t+k_i} \qquad \forall\, i \in M, t \in T, k_i = \{d_i, d_i + 1, \ldots, d_i + p_i - 1\}. \tag{6.4}$$

Further, to prevent $y_{i,t}$ and $u_{i,t}$ that are outside the correspondent time interval of $z_{i,t}$ from taking the value one,

$$\sum_t y_{i,t} = d_i \qquad \forall\, i \in M, \tag{6.5}$$

$$\sum_t u_{i,t} = p_i \qquad \forall\, i \in M. \tag{6.6}$$

Note that the value of $u_{i,t}$, which represents the extended turnaround duration, takes the value 1 if any of the $z_{i,\tau} = 1$ where $\tau \in \{t - d_i - p_i + 1, \ldots, t - d_i\}$. This is true even if the particular unit's turnaround is not extended. The reason we model it in this way is that we can now think of the uncertainty as a parameter, namely the manpower requirement when we model the manpower resource constraint according to the framework of Bertsimas & Sim (2003) as

$$\begin{aligned} &\sum_{k\in K}\sum_{i\in M} q_{i,k} y_{i,t} \\ &+ \sup_{\substack{\{S_t\cup\{\tau_t\}|S_t\subseteq M,\\ |S_t|=\lfloor\Gamma_t\rfloor,\tau_t\in M\setminus S_t\}}} \left\{ \sum_{k\in K} \Big(\sum_{i\in S_t} q^e_{i,k} u_{i,t} + (\Gamma_t - \lfloor\Gamma_t\rfloor) q^e_{\tau_t,k} u_{i,t}) \Big) \right\} \le m_t^{\max} \quad \forall\, t \in T. \end{aligned} \tag{6.7}$$

This enforces that only a subset of the units undergo an extended turnaround in any given time period.

The function that protects the tth constraint from the uncertainty is

$$\beta_t(u^*, \Gamma_t) = \sup_{\substack{\{S_t\cup\{\tau_t\}|S_t\subseteq M,\\ |S_t|=\lfloor\Gamma_t\rfloor,\tau_t\in M\setminus S_t\}}} \left\{ \sum_{k\in K} \left(\sum_{i\in S_t} q^e_{i,k} u^*_{i,t} + (\Gamma_t - \lfloor\Gamma_t\rfloor) q^e_{\tau_t,k} u^*_{i,t}) \right) \right\},$$

which may be represented by the problem

$$\begin{aligned} \beta_t(u^*, \Gamma_t) = \max \ & \sum_{i\in M} u^*_{i,t} v_{i,t} \sum_{k\in K} q^e_{i,k} \\ \text{s.t.} \ & \sum_{i\in M} v_{i,t} \le \Gamma_t \\ & 0 \le v_{i,t} \le 1 \qquad \forall\, i \in M. \end{aligned} \tag{P1}$$

Writing the dual of Problem P1, we get

$$\begin{aligned}
&\min \sum_{i\in M} w_{i,t} + \Gamma_t r_t \\
&\text{s.t. } w_{i,t} + \Gamma_t \geq u^*_{i,t} \sum_{k\in K} q^e_{i,k} && \forall\, i \in M \\
&\quad w_{i,t} \geq 0 && \forall\, i \in M \\
&\quad r_t \geq 0, && \text{(P2)}
\end{aligned}$$

where $w_{i,t}$ and r_t are dual variables.

We then directly replace the protection function $\beta_t(u^*, \Gamma_t)$ in Equation 6.7 by the objective of Problem P2 and tag on the other constraints to the original problem. We repeat this procedure for each $t \in T$, and for each manpower skill type $k \in K$, as each is bounded by $\widehat{m}^{\max}_{k,t}$, and the usage of each is affected by turnaround durations. The constraints for the overall manpower limit are derived from Problem P2 and Equation 6.7, and can be written as

$$\sum_{k\in K}\sum_{i\in M} q_{i,k} y_{i,t} + \sum_{i\in M} w_{i,t} + \Gamma_t r_t \leq m^{\max}_t \qquad \forall\, t \in T \tag{6.8}$$

$$w_{i,t} + \Gamma_t \geq u^*_{i,t} \sum_{k\in K} q^e_{i,k} \qquad \forall\, i \in M, t \in T \tag{6.9}$$

$$w_{i,t} \geq 0 \qquad \forall\, i \in M, t \in T \tag{6.10}$$

$$r_t \geq 0 \qquad \forall\, t \in T. \tag{6.11}$$

Similarly, we can do this for each type of manpower, through

$$\sum_{i\in M} q_{i,k} y_{i,t} + \sum_{i\in M} \widehat{w}_{i,t,k} + \gamma_{t,k}\widehat{r}_{t,k} \leq \widehat{m}^{\max}_{k,t} \qquad \forall\, t \in T, k \in K \tag{6.12}$$

$$\widehat{w}_{i,t,k} + \gamma_{t,k} \geq u^*_{i,t} q^e_{i,k} \qquad \forall\, i \in M, t \in T, k \in K \tag{6.13}$$

$$\widehat{w}_{i,t,k} \geq 0 \qquad \forall\, i \in M, t \in T, k \in K \tag{6.14}$$

$$\widehat{r}_{t,k} \geq 0 \qquad \forall\, t \in T, k \in K. \tag{6.15}$$

Finally, to capture the makespan,

$$T_{\text{end}} \geq t\, u_{i,t} \qquad \forall\, i \in M, t \in T. \tag{6.16}$$

The entire robust optimization formulation consists of Equations 6.1–6.6, 6.8–6.15, and 6.16. Note that the robust formulation does not use any information from the site network structure, and hence the flows and inventory levels.

6.3.1.2 *Model formulation for multi-stage stochastic programming*

Once the above formulation is solved, we get a feasible robust schedule, which we then fix in the site network model that has inventory and mass balances, and unit ratio constraints. The problem is formulated as a multi-stage stochastic linear program for a time span of 9–10 months, where the first 6 months are devoted to production planning, and the first turnaround is scheduled to take place at the 6-month mark. This represents the start of the second stage, and appropriate flows are set to zero for the corresponding scenarios. The start of each subsequent turnaround represents the start of a stage, as uncertainty in turnaround duration is revealed at this point.

The stochastic programming formulation also includes holding costs and demand penalties and constraints. However, it is not concerned any longer with maintenance manpower, as the robust scheduling formulation specifies the manpower requirement at each time period and is no longer a variable.

The objective for the multi-stage stochastic program is to maximize expected profit through

$$\begin{aligned}\max \quad & \sum_{t\in T_1}\left(\sum_{i\in P} c_{i,t}x^1_{i,\infty,t} - \sum_{i\in S} h_i s^1_{i,t}\right) - \sum_{i\in P}\sum_{\substack{\delta:\delta\in\Delta,\\ D_\delta\cap T_1\neq\emptyset}} \rho_{i\delta}\ell^1_{i,\delta} \\ & + \sum_{a\in A}\left(\pi_a\left(\sum_{t\in T\setminus T_1}\left(\sum_{i\in P} c_{i,t}x_{i,\infty,t,a} - \sum_{i\in S} h_i s_{i,t,a}\right) - \sum_{i\in P}\sum_{\substack{\delta:\delta\in\Delta,\\ D_\delta\cap T_1=\emptyset}} \rho_{i,\delta}\ell_{i,\delta,a}\right)\right),\end{aligned} \tag{6.17}$$

where T_1 corresponds to the time periods in the first stage and the terms $x_{i,\infty,t}$ correspond to quantity sold to market from storage tank i in time period t. The objective function considers relative margins for sold product, holding costs, and demand penalties. Note that we consider quantity-based demand penalties, although event-based penalties may be added as well. The demand periods $\delta \in \Delta$ may

Table 6.2: Set, parameter and variable notation for multi-stage stochastic programming model

Entity	Set
Set of inventory tanks that store final products	P
Set of time periods	T
Set of first stage time periods	T_1
Set of producing and inventory units	I
Set of inventory tanks	S
Set of units that undergo a turnaround	M
Set of alternative scenarios	A
Set of demand periods	Δ
Set of time periods belonging to a demand period $\delta \in \Delta$	D_δ
Parameter	**Notation**
Holding cost	h_i
Relative profit margin of product	$c_{i,t}$
Probability of scenario alternative a	π_a
Maximum extension of turnaround duration	p_i
Demand for product i in a demand period δ	$\mu_{i\delta}$
Demand penalty for product i for demand period δ	$\rho_{i\delta}$
Binary parameter denoting start of turnaround	$\xi_{i,t}$
Duration of turnaround for unit i in scenario a	η_{ia}
Variable	**Notation**
Storage level in scenario a	$s_{i,t,a}$
Storage level for first stage	$s^1_{i,t}$
Flow rate from unit i to j in scenario a	$x_{i,j,t,a}$
Flow rate from unit i to j in first stage	$x^1_{i,j,t}$
Unmet demand	$\ell_{i,\delta,a}$
Unmet demand in first stage	$\ell^1_{i,\delta}$

correspond to a particular set of four weeks, for instance, and D_δ are the individual time periods in those four weeks.

We then have a number of constraints relating to network flows. First, we have the mass and inventory balance for the first stage

$$\sum_{i\in I} x^1_{i,j,t} + s^1_{j,t-1} = s^1_{j,t} + \sum_{i\in I} x^1_{jit} \quad \forall\, t \in T_1, j \in I, \tag{6.18}$$

and, for the subsequent stages,

$$\sum_{i\in I} x_{i,j,t,a} + s_{j,t-1,a} = s_{j,t,a} + \sum_{i\in I} x_{jita} \quad \forall\, t \in T\backslash T_1, j \in I, a \in A. \tag{6.19}$$

For units that do not store material,

$$s_{j,t,a} = 0, \quad \forall\, j \in I\backslash S, t \in T, a \in A. \tag{6.20}$$

Next we have the unit ratio constraints, which constrain the ratio of inputs or outputs to a particular unit due to stoichiometric considerations.

$$\begin{aligned} &x_{k,i,t,a} = r^{\text{in}}_{k,j}\, x_{j,i,t,a} \qquad \forall\, a \in A, t \in T,\ i \in I_{\text{in}},\ j,k \in I, \\ &\qquad \text{where } x^U_{j,i}, x^U_{k,i} \neq 0,\ j < k, \end{aligned} \tag{6.21}$$

$$\begin{aligned} &x_{i,j,t,a} = r^{\text{out}}_{j,k,a}\, x_{i,k,t,a} \qquad \forall\, a \in A, t \in T,\ i \in I_{\text{out}},\ j,k \in I, \\ &\qquad \text{where } x^U_{i,j}, x^U_{i,k} \neq 0,\ j < k, \end{aligned} \tag{6.22}$$

We have similar constraints for the first stage inventory level and flow variables.

In terms of inventory capacity constraints, there are

$$s_{j,t,a} \geq s^{\min}_j \qquad a \in A, j \in S,\ t \in T \tag{6.23}$$

$$s_{j,t,a} \leq s^{\max}_j \qquad a \in A, j \in S,\ t \in T. \tag{6.24}$$

There are also standard nonnegativity constraints on all of the flows in the network,

$$x_{i,j,t,a} \geq 0 \qquad i,j \in I,\ t \in T, a \in A. \tag{6.25}$$

We have minimum flow constraints on the flows, which is given by

$$x_{i,j,t,a} \geq x^L_{i,j,t,a} \qquad \forall\, i,j \in I, t \in T, a \in A, \tag{6.26}$$

where the lower bound on the flows is time-dependent only because we define the lower limit to be zero during maintenance.

Flows are forced to go to zero for the duration of the turnaround via

$$x_{i,j,t,a} \leq x^U_{ij} \left(1 - \sum_{\tau=t}^{t+\eta_{ia}-1} \xi_{i,t}\right) \quad \forall\, i \in M, j \in I, t \in T \backslash T_1, a \in A, \tag{6.27}$$

where we know the values of $\xi_{i,t}$ from the robust formulation.

Next, we enforce non-anticipativity constraints that make scenarios equivalent before the uncertainty is revealed, i.e., before they are distinguishable, by

$$x_{i,j,t,a} = x_{i,j,t,a'} \qquad \forall\, i,j \in I, a < a', t = \alpha_{aa'}, \tag{6.28}$$

where $\alpha_{aa'}$ is a two-dimensional parameter table that specifies at what time two scenarios become distinguishable from each other, and can be determined by examining the scenario tree. Similarly, we do the same for the inventory levels

$$s_{i,t,a} = s_{i,t,a'} \qquad \forall\, i \in S, a < a', t = \alpha_{aa'}. \tag{6.29}$$

The demand constraints for the first stage may be modeled as

$$\sum_{t \in D_\delta} x^1_{i,\infty,t} + \ell^1_{i,\delta} = \mu_{i,\delta} \qquad \forall\, i \in P, \{\delta : \delta \in \Delta, D_\delta \cap T_1 \neq \emptyset\}, \tag{6.30}$$

and for subsequent stages as

$$\sum_{t \in D_\delta} x_{i,\infty,ta} + \ell_{i,\delta,a} = \mu_{i,\delta} \qquad \forall\, i \in P, a \in A, \{\delta : \delta \in \Delta, D_\delta \cap T_1 = \emptyset\}. \tag{6.31}$$

We enforce nonnegativity constraints on $\ell^1_{i,\delta}$ and $\ell_{i,\delta,a}$.

Once we solve the multi-stage model for all the robust schedules with minimum makespan, it is then straightforward to judge which of the schedules corresponds to the highest expected profit, once manpower costs are added on.

6.3.2 *Simultaneous robust scheduling and production planning*

In this section, we formulate a two-stage stochastic MILP for simultaneously finding a schedule robust to duration uncertainty for the turnarounds, while making production planning decisions. The first stage variables include binary variables for the scheduling as well as the inventory and production decisions for the first six months of the horizon considered. The second stage comprises of production decisions for the scenarios for all realizations of the uncertain parameters (the duration of each of the turnarounds).

This two-stage approximation assumes knowledge of all the turnaround durations at the start of the second stage. Although this is an approximation, the hope is that the simultaneous decision-making for scheduling and production may either (1) be easier to solve in one shot, rather than solving robust and multi-stage linear problems for all possible minimum makespan schedules; or (2) reveal different solutions that result in higher expected profits. To do this comparison, we can inject the schedule recommended by this two-stage MILP into the multi-stage linear program. The notation we use is similar to that used in Sections 6.3.1.1 and 6.3.1.2.

The objective is formulated as the maximization of the expected profit, and its formulation is identical to the multi-stage objective in Equation 6.17 with the addition of a manpower cost. This cost is added as even though the manpower cost is fixed for the nominal duration of each of the turnarounds, the manpower guaranteed for the extended duration will change, depending on the schedule. Since the two-stage formulation has a cost-based objective and also tries to determine a schedule, we can add this cost to the objective function. We compute this cost by using the left-hand side from Equation 6.12 as

$$b_{k,t} = \sum_{i \in M} q_{i,k} y_{i,t} + \sum_{i \in M} \widehat{w}_{i,t,k} + \gamma_{t,k} \widehat{r}_{t,k} \qquad t \in T \backslash T_1, k \in K. \tag{6.32}$$

The modified objective becomes

$$\max \quad \sum_{t\in T_1}\left(\sum_{i\in P} c_{i,t}x^1_{i,\infty,t} - \sum_{i\in S} h_i s^1_{i,t}\right) - \sum_{i\in P}\sum_{\substack{\delta:\delta\in\Delta,\\ D_\delta\cap T_1\neq\emptyset}} \rho_{i\delta}\ell^1_{i,\delta} - \sum_{k\in K}\sum_{t\in T\setminus T_1}\lambda_{k,t}b_{k,t}$$
$$+\sum_{a\in A}\left(\pi_a\left(\sum_{t\in T\setminus T_1}\left(\sum_{i\in P} c_{i,t}x_{i,\infty,t,a} - \sum_{i\in S} h_i s_{i,t,a}\right) - \sum_{i\in P}\sum_{\substack{\delta:\delta\in\Delta,\\ D_\delta\cap T_1=\emptyset}}\rho_{i,\delta}\ell_{i,\delta,a}\right)\right), \tag{6.33}$$

where $\lambda_{k,t}$ is the price per unit of manpower of type k at time t.

For the scheduling decisions, the equations representing the robust counterpart are identical to those given in Section 6.3.1.1, namely Equations 6.1–6.6, 6.8–6.15, except that we control the set T to be restricted to the time periods beyond the first stage, i.e., $T\setminus T_1$.

Similar to the multi-stage formulation in Section 6.3.1.2, we have the typical inventory and mass balance constraints for the first stage.

$$\sum_{i\in I} x^1_{i,j,t} + s^1_{j,t-1} = s^1_{j,t} + \sum_{i\in I} x^1_{j,i,t} \quad \forall\, t\in T_1, j\in I, \tag{6.34}$$

and, for the second stage,

$$\sum_{i\in I} x_{i,j,t,a} + s_{j,t-1,a} = s_{j,t,a} + \sum_{i\in I} x_{j,i,t,a}$$
$$\forall\, t\in T\setminus\{T_1\cup\{|T_1|+1\}\}, j\in I, a\in A. \tag{6.35}$$

We connect the inventories across the stages through

$$\sum_{i\in I} x_{ij,|T_1|+1,a} + s^1_{j,|T_1|} = s_{j,|T_1|+1,a} + \sum_{i\in I} x_{j,i|T_1|+1,a} \quad \forall\, j\in I, a\in A. \tag{6.36}$$

We also include the unit ratio constraints from Equations 6.21, 6.22, inventory bounds from Equations 6.20, 6.23, 6.24, flow bounds from Equation 6.27 (albeit with $\xi_{i,t}$ now being a binary variable rather than a parameter), and demand constraints from Equations 6.30, 6.31. Minimum flow constraints as in Equation 6.26 are enforced only for the first stage time periods. These comprise the entire two-stage formulation.

The solution of this two-stage formulation provides a schedule, which we can then fix in the multi-stage formulation to better plan production.

The difference here is that we do not have the non-anticipativity constraints for the variables beyond the first stage since we are approximating the multi-stage situation with two stages. The reason we do this is because we can no longer formulate the problem as a multi-stage mixed-integer program as we do not know the structure of the scenario tree, which depends on when the maintenance decisions are made. To more accurately formulate this, we could treat this as a problem with endogenous uncertainty, and use formulations such as those proposed in Goel & Grossmann (2006). This is beyond the scope of the current study.

6.4 RESULTS

As a case study, we use one of the results obtained in Chapter 5, illustrated in Figure 6.3. We consider the set of units that have scheduled turnarounds in the first quarter of the second year, namely Units 4, 5, 9, 10, 11, and 13. These units have been highlighted in the network as well as in the Gantt chart.

We retain a weekly discretization scheme for the discrete-time mixed-integer formulations as (1) the inventory capacities in the network are able to accommodate a week's worth of production or more; (2) demands are considered to be monthly; and (3) turnaround durations are on the order of weeks.

We classify manpower by skill and assume that each unit requires a certain number of each type. We have a cap on the manpower of each skill type, as well as a cap on the total manpower allowed at any particular time at the site, due to safety considerations.

The base case that we consider is where we allow up to two turnarounds to exceed their nominal duration at any particular time period, and assume that demand is 90% of production capacity for all the products. We assume that the probability that turnarounds take the nominal duration is 80% and the probability that they take the full extended duration is 20%.

We compare schedules and discuss inventory levels, manpower usage in more detail for the base case for both the schemes described in Section 6.3. Further, we compare expected profits, problem sizes and solution times for the two approaches.

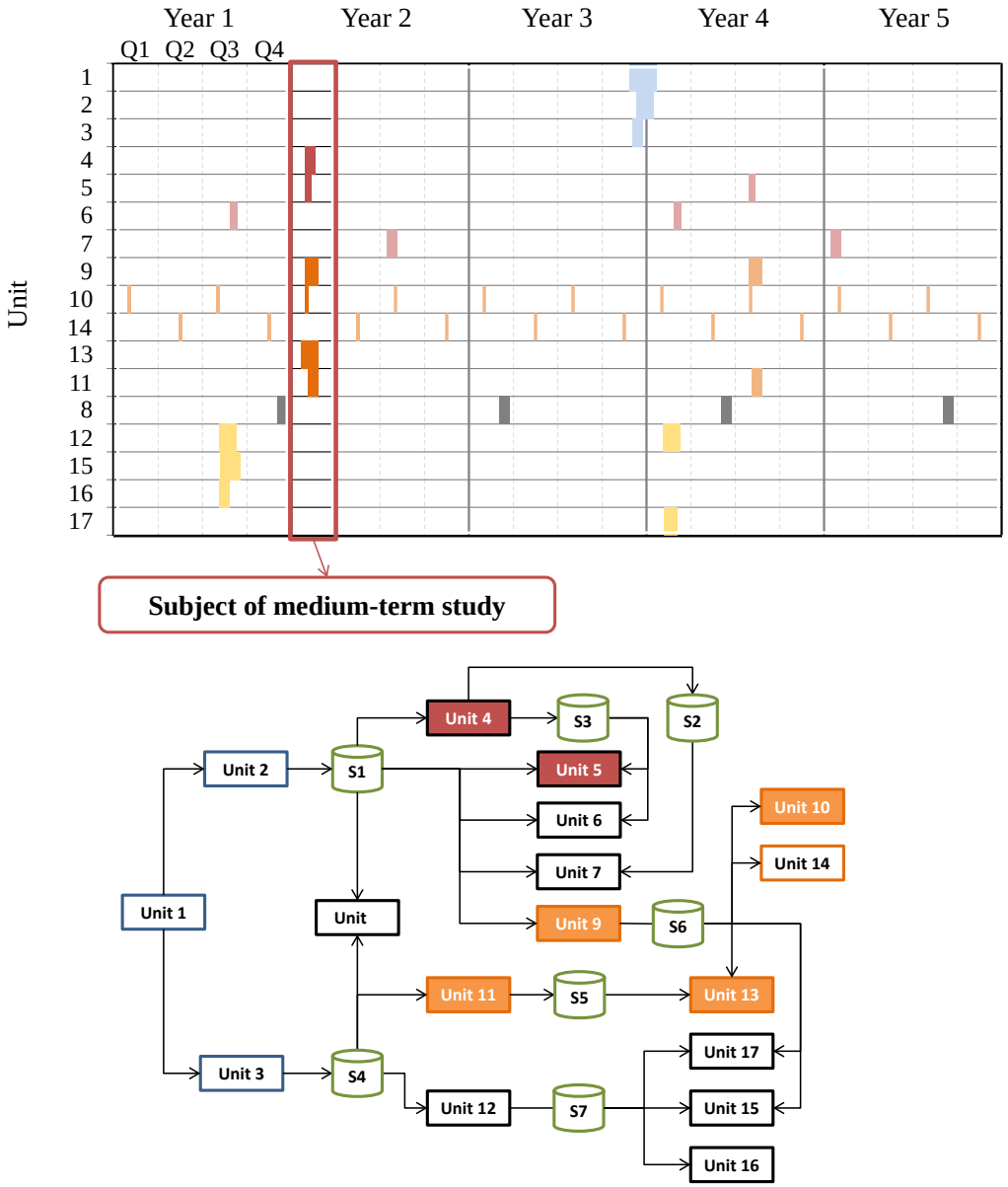


Figure 6.3: Illustration of the units being considered for the medium-term case study

For the base case, we first solve the robust scheduling problem and obtain all schedules with the minimum makespan, which is 11 weeks. We find 12 different solutions that correspond to this makespan. The problem sizes for the robust scheduling problem are small and solution is very quick ($\sim$ 1 s). The recommended schedules are then inserted into the multi-stage stochastic linear program, where the first stage lasts six months. Subsequent stages may be lumped together or separated, based on when units are shut down for maintenance.

6.4.1 *Robust solution*

Figure 6.4 illustrates a case where the robust scheduling formulation effectively avoids having to guarantee manpower availability for all units through the budget uncertainty concept. The figure shows a Gantt chart and the corresponding manpower utilization chart, broken down by each type of manpower. The hollow parts of bars indicate the manpower that would need to be provided if we considered the absolute worst case in the sense of Soyster (1973). At week 28, the worst case still does not violate the bounds, but still we need to guarantee availability of a fewer number of maintenance personnel than the worst case. At week 29, the worst case requirement actually violates the bounds on each type of manpower. However, the budget uncertainty constraint allows this schedule to be feasible, and we see that the constraints are not violated for a budget of two. Note that this is just an example schedule, and does not correspond to the cost-optimal schedule that we find.

6.4.2 *Comparison of sequential and simultaneous approaches*

We first discuss the quality of the schedules recommended by the sequential approach, and then compare the best one with the schedule obtained from the simultaneous approach.

Figure 6.5 illustrates the schedule that corresponds to the highest profit when used in the multi-stage production planning model. The figure recommends that the pairs of Units 4 and 5, Units 9 and 13, and Units 10 and 11 undergo their respective turnarounds together. An examination of the site network (Figure 6.3) shows that Units 4 and 5 are adjacent in the site network, and so are Units 9 and 13. If Units 9

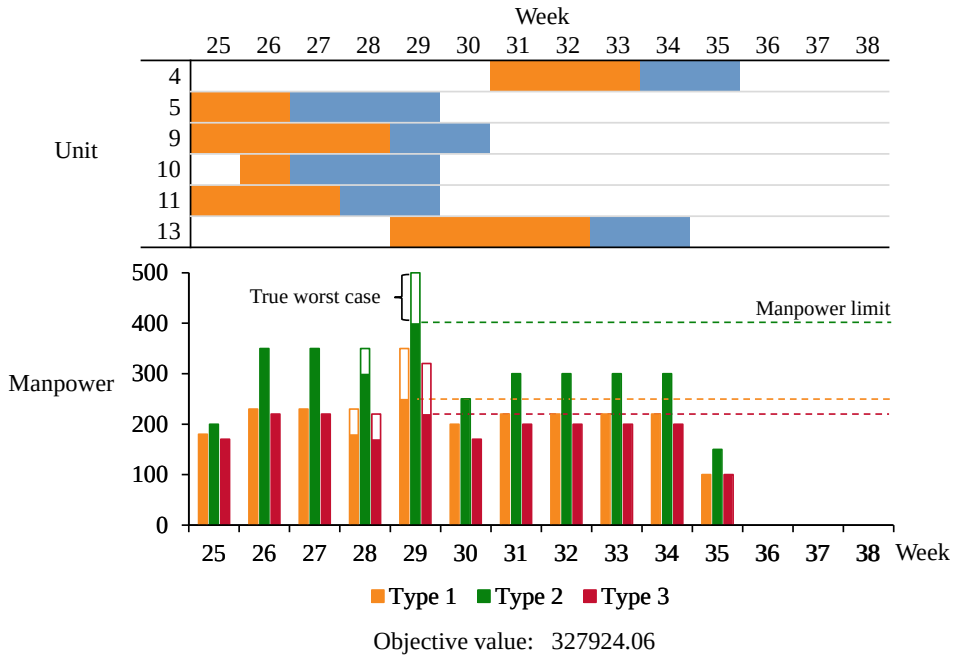


Figure 6.4: Schedule illustrating the effectiveness of the robust scheduling constraints

and 13 had their turnarounds decoupled, it would have meant that the inventory in storage tank S6 would have to supply Units 10, 13, 14, 15, and 17 with raw material when Unit 9 was undergoing a turnaround. Coupling the turnaround of Unit 9 with Unit 13 alleviates this burden on S6 to some extent.

Figure 6.6 corresponds to the schedule corresponding to the worst objective value when used in the multi-stage production planning model. The figure recommends performing turnarounds on Units 4, 9, and 10 together, and Units 5, 11, and 13 together. This is clearly less suitable for the site network, as Units 5 and 13 are decoupled in the network and do not have much effect on each others production. Similarly, Units 4 and 10 are decoupled. In addition, 4 and 9 both receive raw material from storage tank $S1$, and performing turnaround on both of these at once does not seem optimal, and this is vindicated by the fact that this is the worst-performing schedule.

The manpower chart shows the peak manpower limit, which is 900 personnel in the case studied. It also categorizes the manpower, where the orange part indicates the required manpower is accounted for during the nominal turnaround duration, and the blue indicates the manpower required to be available if at most two of the units have an extended turnaround. The pattern indicates the type of manpower

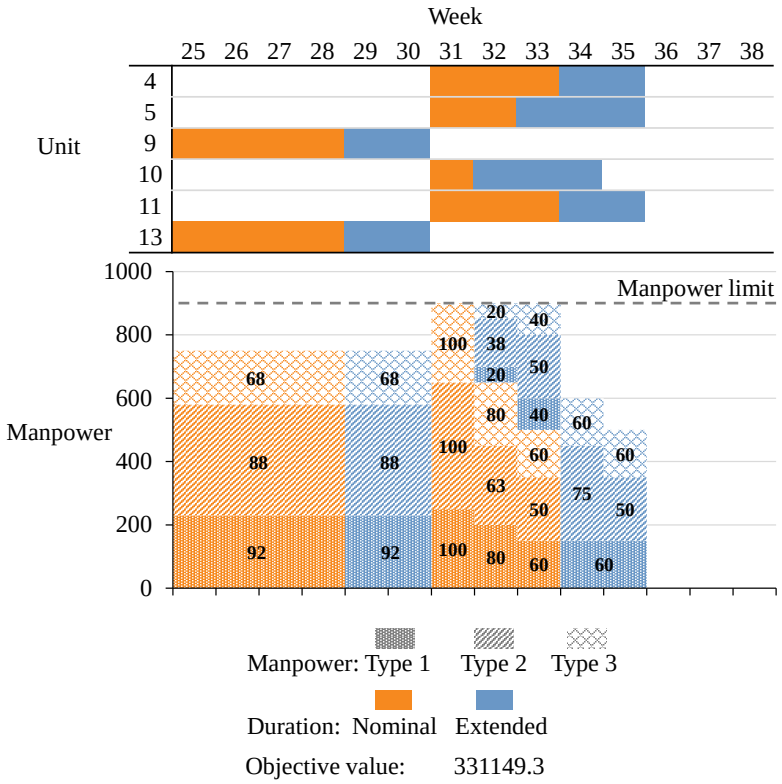


Figure 6.5: Schedule resulting in best objective with corresponding manpower requirement for the sequential robust optimization and multi-stage stochastic programming approach

involved. The number within the stacked bars indicate what percentage of the peak available manpower of a particular type is required to be made available in each week.

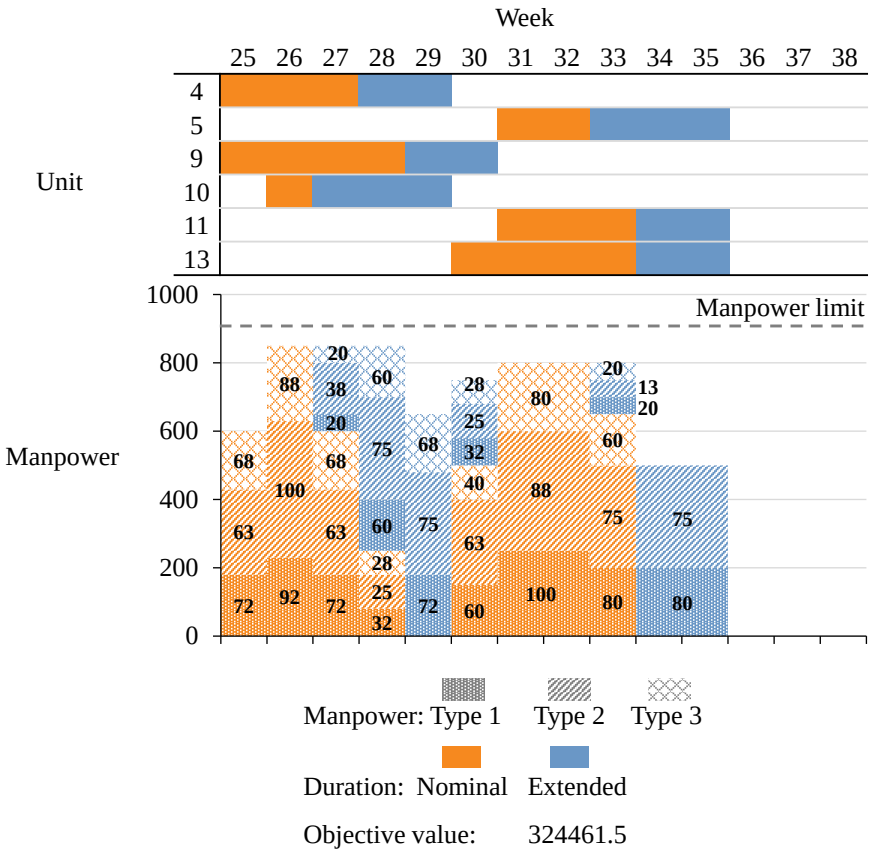


Figure 6.6: Schedule resulting in worst objective with corresponding manpower requirement for the sequential robust optimization and multi-stage stochastic programming approach

Although the difference in objective values is only about 2% in value, one must note that the objective measures profit over a 9-month horizon from a large integrated site and can, therefore, be significant.

Note that the robust optimization model has no access to the site network structure, and the ultimate performance of the stochastic programming model is at the mercy of the schedule recommended to it by the robust schedule solution. The combined robust and two-stage stochastic programming formulation does have access to the site network structure, and also contains the information to create a robust schedule. The solution for the combined two-stage formulation followed

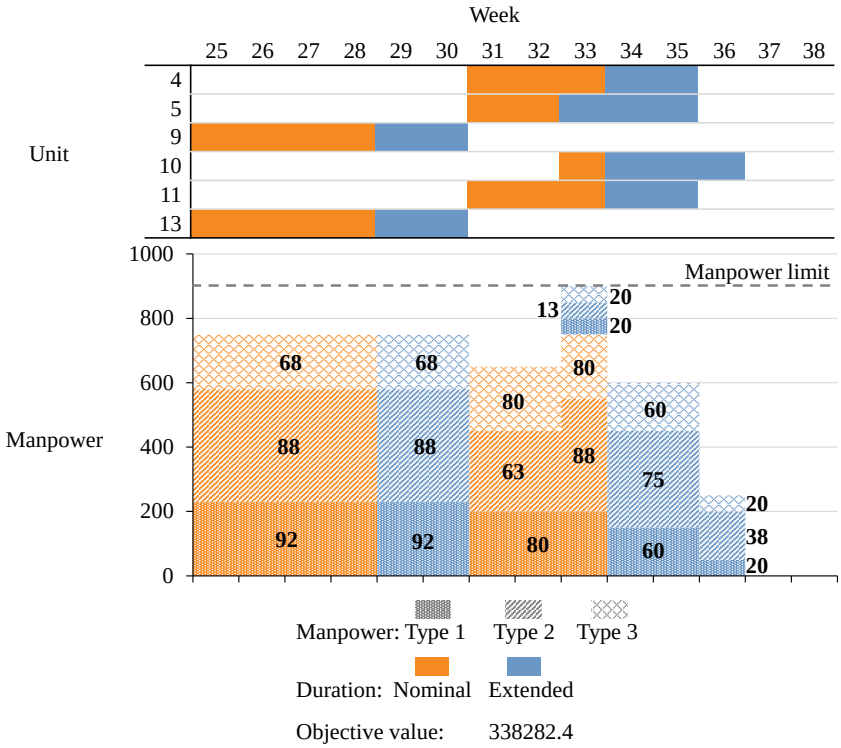


Figure 6.7: Manpower usage for best schedule from the simultaneous approach, that corresponds to a makespan that is not the smallest possible

by multi-stage stochastic program is illustrated in Figure 6.7. We see that this formulation is able to do better than the best solution offered by the sequential robust and stochastic programming formulations, again by about 2%. The solution offered by this formulation recommends a schedule whose duration is greater than the minimum makespan. This indicates that the best solution does not necessarily correspond to the minimum makespan solution, and the two-stage model makes use of the knowledge of the network structure to exploit this.

Note that in the objective values reported in Figures 6.5, 6.6 and 6.7, we do not include the cost of manpower. The cost of manpower is the same for the nominal duration, as all the Units considered have to undergo a turnaround, and each has a certain requirement for manpower. However, they each differ in the total amount of manpower they are required to make available across periods. This is because of the budget uncertainty, where the manpower required to be made available in a certain time period depends on which units may have an extended turnaround occurring in that time period, and on the corresponding manpower requirements.

For instance, if one computes the area under the graph in Figure 6.6, one can see that 3250 additional man-weeks are required to hedge against the uncertainty. If we do a similar computation for Figure 6.7, we see that 3100 additional man-weeks have to be guaranteed. This is interesting, as not only does the two-stage formulation give a better answer in terms of expected profit, but also has a reduced cumulative requirement for additional manpower. This translates into further savings, the amount of which is determined by the manpower costs, as reflected in Equation 6.33.

We do not need to include the cost of this additional manpower in the sequential robust and stochastic formulation, because once the schedule is fixed (by the robust formulation), we know how much additional manpower needs to be provided for. As discussed above, this cost is easily added to the two-stage simultaneous formulation, as can be seen in Equation 6.33. The two-stage formulation considers this cost when making scheduling decisions.

6.4.3 *Computational experience*

As far as modeling is concerned, we can use one of the two representations of the scenario tree, namely the explicit or implicit form. It should be noted that we work with the implicit form of the scenario tree, in that we do not formulate explicit non-anticipativity constraints. Instead, the non-anticipativity constraints are implicit, which means that variables corresponding to a certain stage are created only from the point when the scenarios become distinguishable. Given that there are six units undergoing a turnaround around the same time, we are now dealing with a seven-stage problem. An illustration of the two trees for our problem is shown in Figure 6.8.

If we assume two possibilities of turnaround duration for each unit and create variables for each of the $2^6 = 64$ scenarios and enforce relevant non-anticipativity constraints, we find that our model is too large to be loaded into memory. The largest problem we are able to solve is a four-stage problem in explicit form (Table 6.3). An alternative is to use a column generation technique to solve the problem. Instead, an implicit representation significantly cuts down on the number of variables created, and we are able to solve the seven-stage problem in an average of 20 s.

For the example case study we consider, this totals approximately $20 \times 12 = 240$ s. We simply pick the schedule which corresponds to the best expected profit.

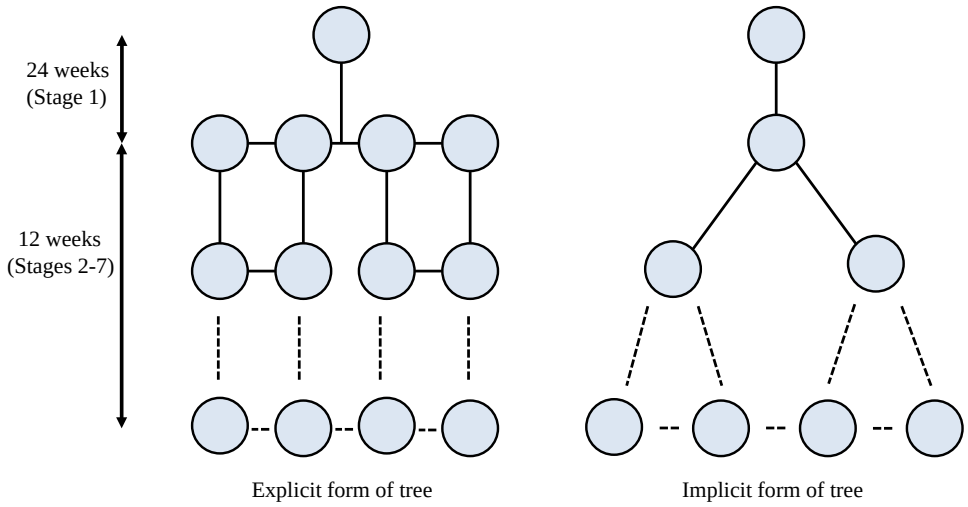


Figure 6.8: Implicit and explicit representations of scenario tree for the medium-term turnaround scheduling case study

Figure 6.5 shows the best schedule and the corresponding manpower utilization. Figure 6.6 corresponds to the worst schedule and corresponding manpower utilization.

Table 6.3 illustrates the solution times and sizes of the (1) robust optimization MILP; (2) multi-stage LP; and (3) combined robust optimization and two-stage stochastic MILP problems. All problems were solved using CPLEX 12.6 in GAMS 24.3. For comparison, the table also shows the size and solution time for the largest problem in explicit form that we are able to solve directly.

Table 6.3: Solution times and problem sizes of sequential and simultaneous strategies

	RO MILP	Multi-stage LP (implicit form)	RO+2-stage MILP	Multi-stage LP (explicit form; 4 stages)
Avg. Time (s)	< 0.1	19.4	$2,714$	33.32
Size				
Equations	2243	$87,409$	$120,212$	$4,601,645$
Continuous vars	1900	$42,736$	$65,032$	$712,485$
Binary vars	252	-	277	-

Two measures that illustrate the impact of uncertainty in a stochastic optimization problem are the value of stochastic solution (VSS) and the expected value of

perfect information (EVPI). The VSS provides a measure of how valuable it is to account for uncertainty in the optimization. It is harder to interpret in the multi-stage case, and there are multiple ways of computing this (Chiralaksanakul & Morton 2004; Maggioni et al. 2012). Another reason we avoid this is that the expected values of the uncertain parameter are fractional weeks, which does not allow us to evaluate this with our weekly discretized formulation. However, we focus on the EVPI measure here. The EVPI measures how different the expected profit is from performing stochastic optimization under uncertainty from when we have exact estimates of the uncertain parameters, or

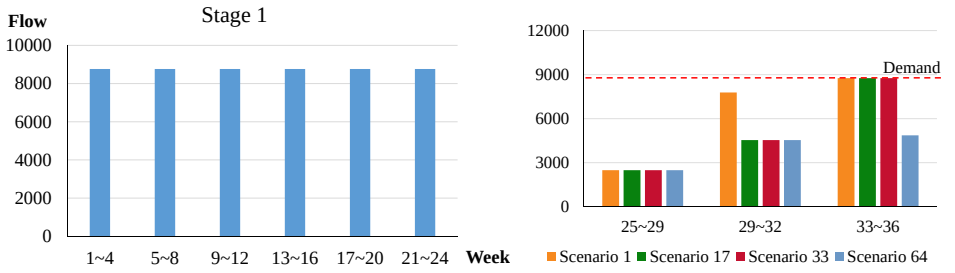
$$\begin{aligned}\text{EVPI} &= \text{Recourse problem objective} - \text{Wait-and-see objective} \\ &= 338,282.4 - 351,336.74 = -13,054.34\end{aligned}$$

6.4.4 *Analysis of inventory policy and demand satisfaction levels*

We use Unit 13 as an example to illustrate the nature of inventory policy and demand satisfaction that arises from our case study. According to Figure 6.7, Unit 13 undergoes its turnaround from Weeks 25–28. Figure 6.9a illustrates that first stage demands for product from Unit 13 can be satisfied.

In the second stage, as shown in Figure 6.9b, this is not always possible, even though we attempt to build downstream inventory in the time leading up to the turnaround. As demands are monthly, we show aggregated monthly demands in the figures. The demand that would be satisfied is illustrated for four scenarios. In scenario 1, all turnarounds achieve the nominal duration, and the recommended solution shows that demand satisfaction levels are far higher in Weeks 29–32 than for the other scenarios. In the worst case, when all turnarounds are assumed to take the extended duration, Scenario 64 (the most unlikely case) shows that we cannot satisfy the demands even in Weeks 33–36, even though Unit 13 undergoes its turnaround in Weeks 25–29.

Figure 6.10 shows how inventory build-up is planned for the downstream storage tank of Unit 13. As the inventory capacity for the storage tank is 2500, we see that it takes about 2.5 months to build up this inventory in the first stage. The entire inventory storage is only able to satisfy about 28% of the demands in Weeks 25–28. The demand is assumed to be 90% of the total capacity in this case. This result



(a) Demands catered to in Stage 1

(b) Demands catered to in subsequent stages for various scenarios

Figure 6.9: Demand level and demands catered to for Unit 13

shows that, at least from the point of view of Unit 13's products, it may be better to schedule the set of turnarounds when demands are lower.

The reason for the step-wise build up of the inventory is that demands are assumed to be monthly, and therefore only the excess production in the final week of the month gets stored in inventory.

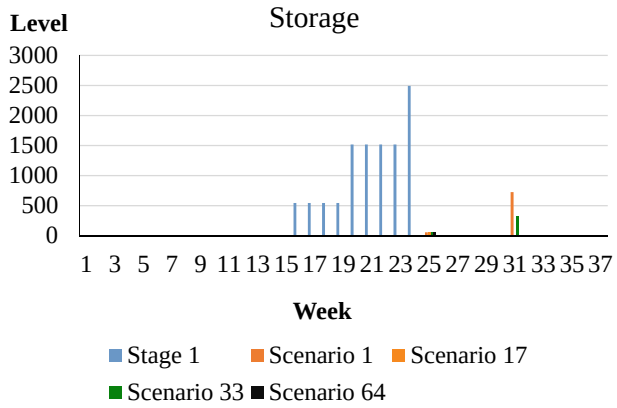


Figure 6.10: Inventory levels for downstream storage tank of Unit 13

6.5 CONCLUSIONS

In this chapter, we investigate the medium-term turnaround planning problem, which addresses both inventory build-up in storage tanks across the entire site in anticipation of a set of turnarounds, as well as maintenance personnel planning decisions while considering uncertainty in duration of turnarounds.

Two approaches were proposed to handle the uncertainty in turnaround duration. The first involved a sequential robust optimization to minimize the turnaround 'makespan' and multi-stage stochastic programming approach to maximize profit. The former determined the turnaround schedule, and the latter made production decisions. We then investigated a combined robust optimization and two-stage stochastic programming formulation to simultaneously make scheduling as well as production decisions. The scheduling decisions were then fixed in the multi-stage stochastic linear program to optimize for production planning.

We observe that the two-stage formulation, though an approximation, considers schedules that are not necessarily of the minimum makespan and also considers manpower costs for the additional manpower that needs to be guaranteed. The trade-off is that solutions are very quick to obtain from the sequential approach, but may not recommend the best performing schedules in terms of profit. The simultaneous approach is able to find superior results, but is also significantly more expensive to perform.

An advantage that the sequential approach has is that one may choose to enhance the manpower and subtask model with more detail, in which case the expense of the simultaneous approach may significantly increase.

A study was then performed on demand satisfaction and inventory planning, and the trends were explained through an analysis of the site network structure and relevant costs.

In summary, the consideration of uncertainty in turnaround duration significantly affects production, inventory, and manpower availability decisions. The consideration of medium-term turnaround planning under uncertainty is thus an important one, and we propose schemes to address this in this chapter.

It should also be noted that part from the specific problem we consider in this chapter, the modeling framework and the sequential and simultaneous solution methods we propose may be used in more general contexts of planning under uncertainty. In particular, it may be applicable in cases where the uncertain parameter affects different aspects of the problem differently, as it affected manpower (no recourse, requiring robust solution) and production decisions (full recourse) differently in the current context. We also demonstrate how the appearance of uncertainty in a parameter appearing as an index may be transformed into left-hand side parameter uncertainty—this may be relevant for other problems as well.

Part IV

CONCLUSIONS

Other people can talk about how to expand
the destiny of mankind. I just want to talk
about how to fix a motorcycle.
— Robert M. Pirsig

7.1 CONCLUSIONS

In this thesis, we explore the consideration of uncertainty in various optimization problems through different paradigms. The first part of the thesis addresses the simulation optimization problem, where we optimize directly over stochastic simulations. The second part of the thesis looks at uncertainty quantification for optimization, along with combined robust optimization and stochastic programming formulations. Connections between these different paradigms may run deeper. For instance, simulation optimization (Part II) may be linked with stochastic programming (addressed in Part III) through the sample path optimization technique, where a simulator is used to generate a number of sample paths that manifest as scenarios in the stochastic programming context.

In short, the various interactions of uncertainty and optimization lead to a rich class of problem-solving paradigms; we focus on the three particular approaches mentioned above, justify their suitability for certain contexts, and demonstrate their use through applications in chemical engineering and operations.

We outline the key contributions made in this thesis, as well as directions for potential future research in both simulation optimization as well as in maintenance turnaround planning under uncertainty.

7.2 KEY CONTRIBUTIONS

7.2.1 *Simulation optimization*

- We contribute a comprehensive literature review of the simulation optimization area, where we provide an introduction to the field, discuss its relationship to other fields, and outline existing algorithms. The work serves as an extensive reference to relevant and recent literature in methods as well as applications,

and it compiles a list of academic as well as commercial software for simulation optimization. No other existing literature provides such an encyclopedic and current treatment of the area.

- We outline a trust region-based simulation optimization algorithm that provides guarantees of global convergence to a stationary point. The extension of traditional DFO methods involving fully linear models to the stochastic context has not been done before, and our development extends the purview of DFO methods to a new class of problems. In this way, the development helps to bridge the derivative-free optimization literature with the simulation optimization literature.
- We demonstrate that the implementation of our algorithm, which is augmented by using nonparametric regression and global optimization techniques, outperforms existing codes in the academic literature.
- We foray into two application domains—one of which relates to a chemical supply chain problem from our industrial collaborators—to demonstrate the applicability of such methods.

7.2.2 *Turnaround planning*

- Although several prior papers in the process systems engineering literature consider both maintenance and production planning, none really addresses the planning of turnaround maintenance on long or medium-term scales. In this regard, we identify an application not addressed before in turnaround optimization for integrated chemical sites.
- We provide a long-term turnaround optimization formulation that we use to coordinate turnarounds across the site network over a multi-year horizon, while maximizing profit, balancing financial performance across time periods, and respecting peak manpower limits. We also provide a model to determine both turnaround frequencies and turnaround policies for a particular plant.
- We address the medium-term turnaround planning problem under uncertainty. In this regard, we address the uncertainty in the duration of the turnaround through a combined robust optimization and stochastic programming approach.

Robust optimization is used for the manpower planning decisions that have no recourse actions, and stochastic programming ideas are used as we allow full recourse for the production decisions in the network.

7.3 FUTURE WORK

With regard to simulation optimization, there are several directions that may be explored beyond this thesis, as listed below:

- Constraints, both stochastic and deterministic, appear in many applications in simulation optimization. The consideration of convergence and implementation issues in this case is a natural next step.

- The convergence proofs in Chapter 3 considered interpolation models, and these involved quantifying the discrepancy between model and function values, and model and function gradients (in the context of fully linear models). This, along with the sample replication scheme, was proposed with the intention of designing a descent-based algorithm using the concept of fully linear models.

  It may be interesting to investigate regression models, rather than interpolation models, as regression models naturally account for noise in observed values. However, the analysis becomes more complicated as one has to not only quantify the discrepancy between the observation (e.g., sample mean) and the true function value, but also between the regressed value and the observation. One path to explore would be to not necessarily guarantee descent at every iteration, not require sample replications, and, therefore, move away from relying on fully linear models. Proofs of probabilistic guarantees of convergence may be possible, and may lead to improved practical implementations.

- Another assumption that we make in Chapters 3 and 4 is that the expected or true response has a Lipschitz continuous derivative. In reality, we may not know the smoothness properties of the underlying function. Future research directions may look at dispensing with this assumption, while providing guarantees of convergence.

- In terms of applications, Chapter 4 identifies different applications for simulation optimization across various domains in chemical engineering. We use these

example problems to illustrate the performance of our algorithm in practical situations, but we do not pursue full-scale studies, nor tailor the algorithm for these sub-domain applications. We believe that further investigation of these examples will shed light on the effectiveness of simulation optimization in simultaneously considering uncertainty while providing better designs, operating decisions, and parameter estimates. In addition, the use of stochastic models to describe phenomena in engineering systems, such as stochastic kinetic models (Srivastava & Rawlings 2014), provides impetus to continue the development of relevant simulation optimization methods.

The SO-LVIT algorithm described in Chapter 4 is designed for black-box optimization. For an application, if additional knowledge is known regarding the nature of the underlying simulations, this may be embedded through the use of appropriate hypothesis tests and choice of Gaussian Process kernel.

A further consideration is the ability to tackle problems where maximizing the expectation alone may not be the objective, but perhaps some notion of risk which may require the estimation of the variance at a point.

- Another endeavor that we recommend is to encourage the development of simulation optimization methodologies by making available real-world examples from engineering through problem testbeds. In addition, we would like to continue to maintain suites of algorithms from the literature in an optimization toolbox, along the lines of Rios & Sahinidis (2012).

- Finally, as we see in Chapter 4, our implementation performs well, but is not able to perform a global search and is therefore not competitive with global solvers for certain classes of problems. The integration of a global optimization routine with this local search is another promising direction for research. For instance, we could optimize an improvement metric (cf. Section 2.3.2) to determine sample points, and in this sense trade-off exploitation and exploration. This would extend the paradigm of optimization under uncertainty, as this would represent an optimization over an estimate of the uncertainty in our surrogate model.

With regard to turnaround management and optimization, we also see several avenues through which one may extend the current work:

- The model for determining turnaround frequencies was developed to demonstrate how one may determine a turnaround policy for an entire plant, whether it be one type of turnaround, major and minor types of turnarounds, or decoupled turnarounds. We presented a case study involving simple cost functions, and two sets of process equipment, and this was used to illustrate the idea. It would be beneficial to look at this problem in more detail, i.e., with a more realistic set of turnarounds and cost functions, in order to better determine an optimal turnaround policy for a plant.

- A major concern for chemical companies is the management of financial risk associated with turnarounds. Future work should involve the quantification of risk of plant breakdown and its impact on profits, in the case when turnarounds are delayed by several months due to unfavorable market conditions, or other external factors.

- On the other end of the time-scale, it would be worth considering contract negotiations and other practical details while planning for maintenance manpower and spare equipment for a particular turnaround or set of turnarounds. This may be of significance, due to the knock-on effect that unplanned maintenance may have on an integrated site. Extended maintenance durations could impact manpower availability, which in turn further effects the duration of shutdown, the plant productivity, and consequently, the productivity of the entire site. Therefore, a more careful and detailed optimization model for planning for both manpower and replacement/spare parts is an interesting direction that may be pursued.

- It may also be worth considering the medium-term turnaround planning problem through a combined turnaround scheduling and multi-stage production planning framework. In this case, the structure of the scenario tree would depend on the scheduling decisions, and would require the consideration of optimization under endogenous uncertainty.

BIBLIOGRAPHY

Abramson, M. A. (2007). `NOMADm` *version 4.5 User's Guide*. Wright-Patterson AFB, OH: Air Force Institute of Technology. (Cited on page 41.)

Albrecht, J. C., Kotani, A., Lin, J. S., Soper, S. A., & Barron, A. E. (2013). Simultaneous detection of 19 k-*ras* mutations by free-solution conjugate electrophoresis of ligase detection reaction products on glass microchips. *Electrophoresis*, *34*, 590–597. (Cited on page 90.)

Albrecht, J. C., Lin, J. S., & Barron, A. E. (2011). A 265-base DNA sequencing read by capillary electrophoresis with no separation matrix. *Analytical Chemistry*, *80*, 509–515. (Cited on page 90.)

Alkhamis, T. M., Ahmed, M. A., & Tuan, V. K. (1999). Simulated annealing for discrete optimization with estimation. *European Journal of Operational Research*, *116*, 530–544. (Cited on page 31.)

Alrefaei, M. H. & Andradóttir, S. (1999). A simulated annealing algorithm with constant temperature for discrete stochastic optimization. *Management Science*, *45*, 748–764. (Cited on page 24.)

American National Standards Institute definition of preventive maintenance (accessed February 21, 2014). `http://webstore.ansi.org/preventive-maintenance/`. (Cited on page 97.)

Ammeri, A., Hachicha, W., Chabchoub, H., & Masmoudi, F. (2011). A comprehensive literature review of mono-objective simulation optimization methods. *Advances in Production Engineering & Management*, *6*(4), 291–302. (Cited on pages 11 and 13.)

Anderson, E. J. & Ferris, M. C. (2001). A direct search algorithm for optimization with noisy function evaluations. *SIAM Journal on Optimization*, *11*, 837–857. (Cited on page 29.)

Andradóttir, S. (1998). Chapter 9: Simulation optimization. In J. Banks (Ed.), *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. John Wiley & Sons, New York. (Cited on pages 11 and 12.)

Andradóttir, S. (2006). An overview of simulation optimization via random search. In S. G. Henderson & B. L. Nelson (Eds.), *Handbooks in Operations Research and Management Science: Simulation*, volume 13 chapter 20, (pp. 617–631). Elsevier. (Cited on page 24.)

Andradóttir, S. & Kim, S.-H. (2010). Fully sequential procedures for comparing constrained systems via simulation. *Naval Research Logistics*, *57*(5), 403–421. (Cited on page 22.)

Angün, E. (2004). *Black box simulation optimization: generalized response surface methodology*. PhD thesis, Tilburg University. (Cited on page 12.)

Angün, E., Kleijnen, J. P. C., Hertog, D. D., & Gurkan, G. (2009). Response surface methodology with stochastic constraints for expensive simulation. *Journal of the Operational Research Society*, *60*(6), 735–746. (Cited on page 26.)

Ayvaz, M. T. (2010). A linked simulation-optimization model for solving the unknown groundwater pollution source identification problems. *Journal of Contaminant Hydrology*, *117*(1–4), 46–59. (Cited on page 19.)

Azadivar, F. (1992). A tutorial on simulation optimization. In Swain, J. J., Goldsman, D., Crain, R. C., & Wilson, J. R. (Eds.), *Proceedings of the 1992 Winter Simulation Conference*, (pp. 198–204). (Cited on page 11.)

Azadivar, J. (1999). Simulation optimization methodologies. In Farrington, P. A., Nembhard, H. B., Sturrock, D. T., & Evans, G. W. (Eds.), *Proceedings of the 1999 Winter Simulation Conference*, (pp. 93–100). (Cited on pages 11 and 12.)

Baker, K. R. (1977). An experimental study of the effectiveness of rolling schedules in production planning. *Decision Science*, *8*, 19–27. (Cited on page 123.)

Balakrishna, R., Antoniou, C., Ben-Akiva, M., Koutsopoulos, H. N., & Wen, Y. (2007). Calibration of microscopic traffic simulation models: Methods and application. *Transportation Research Record: Journal of the Transportation Research Board*, *1999*(1), 198–207. (Cited on page 19.)

Bangerth, W., Klie, H., Matossian, V., Parashar, M., & Wheeler, M. F. (2005). An autonomic reservoir framework for the stochastic optimization of well placement. *Cluster Computing*, *8*(4), 255–269. (Cited on page 19.)

Barton, R. R. & Ivey, Jr., J. S. (1996). Nelder-Mead simplex modifications for simulation optimization. *Management Science*, *42*, 954–973. (Cited on page 29.)

Barton, R. R. & Meckesheimer, M. (2006). Metamodel-based simulation optimization. In S. Henderson & B. Nelson (Eds.), *Handbook in Operations Research and Management Science: Simulation 13* (pp. 535–574). Elsevier. (Cited on page 25.)

Bechhofer, R. E., Santner, T. J., & Goldsman, D. M. (1995). *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*. John Wiley & Sons. (Cited on page 23.)

Ben-Tal, A., Ghaoui, L. E., & Nemirovski, A. (2009). *Robust Optimization*. Princeton University Press. (Cited on pages 5 and 136.)

Bertsimas, D. & Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming*, *98*(1–3), 49–71. (Cited on pages 140, 142, and 143.)

Bertsimas, D. & Sim, M. (2004). The price of robustness. *Operations Research*, *52*(1), 35–53. (Cited on pages 140 and 142.)

Bertsimas, D. & Tsitsiklis, J. (1993). Simulated annealing. *Statistical Science*, *8*(1), 10–15. (Cited on page 30.)

Bettonvil, B., del Castillo, E., & Kleijnen, J. P. C. (2009). Statistical testing of optimality conditions in multiresponse simulation-based optimization. *European Journal of Operational Research*, *199*, 448–458. (Cited on page 20.)

Bhatnagar, S. (2005). Adaptive multivariate three-timescale stochastic approximation algorithms for simulation based optimization. *ACM Transactions on Modeling and Computer Simulation*, *15*(1), 74–107. (Cited on page 19.)

Bianchi, L., Dorigo, M., Gambardella, L. M., & Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, *8*(2), 239–287. (Cited on page 24.)

Birge, J. R. & Louveaux, F. (2011). *Introduction to stochastic programming* (Second ed.). Springer Science+Business Media. (Cited on pages 5, 15, and 138.)

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. (Cited on page 47.)

Box, G. E. P. & Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society*, *XIII*(1), 1–35. (Cited on page 25.)

Cachon, G. & Terwiesch, C. (2008). *Matching Supply with Demand: An Introduction to Operations Management* (2nd ed.). Mc-Graw Hill. (Cited on page 88.)

Carson, Y. & Maria, A. (1997). Simulation optimization: Methods and applications. In Andradóttir, S., Healy, K. J., Winters, D. H., & Nelson, B. L. (Eds.), *Proceedings of the 1997 Winter Simulation Conference*, (pp. 118–126). (Cited on pages 11 and 12.)

Castro, P. M., Grossmann, I. E., Veldhuizen, P., & Esplin, D. (2014). Optimal maintenance scheduling of a gas engine power plant using generalized disjunctive programming. *AIChE Journal*. (Cited on page 101.)

Chang, K.-H. (2008). *Stochastic trust region response surface convergent method for continuous simulation optimization*. PhD thesis, Purdue University. (Cited on page 12.)

Chang, K.-H. (2012). Stochastic Nelder-Mead simplex method–A new globally convergent direct search method for simulation optimization. *European Journal of Operational Research*, *220*, 684–694. (Cited on pages 29, 41, and 84.)

Chang, K.-H., Hong, L. J., & Wan, H. (2007). Stochastic trust region gradient-free method (STRONG)–A new response-surface-based algorithm in simulation optimization. In Henderson, S. G., Biller, B., Hsieh, M.-H., Shortle, J., Tew, J. D., & Barton, R. R. (Eds.), *Proceedings of the 2007 Winter Simulation Conference*. (Cited on pages 43 and 44.)

Chang, K.-H., Hong, L. J., & Wan, H. (2011). Stochastic trust-region response-surface method (STRONG)–A new response-surface framework for simulation optimization. (Cited on pages 50 and 84.)

Chang, K.-H., Hong, L. J., & Wan, H. (2013). Stochastic trust-region response-surface method (STRONG)—A new response-surface framework for simulation optimization. *INFORMS Journal on Computing*, *25*(2), 230–243. (Cited on pages 27, 41, 43, 44, 61, and 62.)

Chen, C.-H. (1995). An effective approach to smartly allocate computing budget for discrete event simulation. In *Proceedings of the 34th IEEE Conference on Decision and Control*, (pp. 2598–2605). (Cited on page 22.)

Chen, C.-H. (1996). A lower bound for the correct subset selection probability and its application to discrete event system simulations. *IEEE Transactions on Automatic Control*, *41*, 1227–1231. (Cited on page 24.)

Chen, C.-H. & Lee, L. H. (2010). *Stochastic Simulation Optimization: An Optimal Computing Budget Allocation*. System Engineering and Operations Research. World Scientific Publishing Company. (Cited on page 12.)

Chen, C.-H., Yücesan, E., Dai, L., & Chen, H.-C. (2009). Optimal budget allocation for discrete-event simulation experiments. *IIE Transactions*, *42*(1), 60–70. (Cited on page 22.)

Chen, H. & Schmeiser, B. W. (1994). Retrospective optimization algorithms for stochastic root finding. In Tew, J., Manivannan, S., Sadowski, D., & Seila, A. (Eds.), *Proceedings of 1994 Winter Simulation Conference*, (pp. 255–261). (Cited on page 29.)

Cheung, K.-Y. & Hui, C.-W. (2001). Total-site maintenance scheduling. In *PRES 2001, 4th Conference on Process Integration, Modelling, and Optimisation for Energy Saving and Pollution Reduction*, (pp. 355–363). (Cited on page 100.)

Cheung, K.-Y., Hui, C.-W., Sakamoto, H., Hirata, K., & O'Young, L. (2004). Short-term site-wide maintenance scheduling. *Computers & Chemical Engineering*, *28*, 91–102. (Cited on page 100.)

Chiralaksanakul, A. & Morton, D. P. (2004). Assessing policy quality in multi-stage stochastic programming. Stochastic Programming E-Print Series. Institut für Mathematik. (Cited on page 160.)

Cho, J. & Dorfman, K. D. (2010). Brownian dynamics simulations of electrophoretic DNA separations in a sparse ordered post array. *Journal of Chromatography A*, *1217*, 5522–5528. (Cited on pages 18 and 91.)

Christer, A. H. & Whitelaw, J. (1983). An operational approach to breakdown maintenance: Problem recognition. *Journal of the Operational Research Society*, *34*, 1041–1052. (Cited on page 98.)

Cohn, A. M. & Barnhart, C. (2003). Improving crew scheduling by incorporating key maintenance routing decisions. *Operations Research*, *51*(3), 387–396. (Cited on page 102.)

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, *4*, 129–145. (Cited on page 16.)

Collins, N. E., Eglese, R. W., & Golden, B. L. (1988). Simulated annealing–an annotated bibliography. *American Journal of Mathematical Management Sciences*, *8*, 209–308. (Cited on page 31.)

Conn, A. R., Gould, N. I. M., & Toint, P. L. (2000). *Trust-region methods*. MOS-SIAM Series on Optimization. (Cited on pages 26 and 49.)

Conn, A. R., Scheinberg, K., & Vicente, L. N. (2009). *Introduction to derivative-free optimization*. Philadelphia, PA: SIAM. (Cited on pages 16, 46, 52, and 67.)

de Angelis, V., Felici, G., & Impelluso, P. (2003). Integrating simulation and optimisation in health care centre management. *European Journal of Operational Research*, *150*, 101–114. (Cited on page 19.)

de Boer, P.-T., Kroese, D. P., Mannor, S., & Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of Operations Research*, *134*, 19–67. (Cited on page 34.)

Dedopoulos, I. T. & Shah, N. (1995a). Optimal short-term scheduling of maintenance and production for multipurpose plants. *Industrial & Engineering Chemistry Research*, *34*, 192–201. (Cited on pages 99 and 135.)

Dedopoulos, I. T. & Shah, N. (1995b). Preventive maintenance policy optimisation for multipurpose plant equipment. *Computers & Chemical Engineering*, *19*, S693–S698. (Cited on pages 106, 107, and 108.)

Deng, G. (2007). *Simulation-based Optimization*. PhD thesis, University of Wisconsin–Madison. (Cited on page 12.)

Deng, G. & Ferris, M. C. (2006). Adaptation of the UOBYQA algorithm for noisy functions. In Perrone, L. F., Wieland, F. P., Liu, J., Lawson, B. G., Nicol, D. M., & Fujimoto, R. M. (Eds.), *Proceedings of the 2006 Winter Simulation Conference*, (pp. 312–319). (Cited on pages 27 and 43.)

Deng, G. & Ferris, M. C. (2007). Extension of the DIRECT optimization algorithm for noisy functions. In Henderson, S. G., Biller, B., Hsieh, M.-H., Shortle, J., Tew, J. D., & Barton, R. R. (Eds.), *Proceedings of the 2007 Winter Simulation Conference*, (pp. 497–504). (Cited on page 36.)

Deng, G. & Ferris, M. C. (2009). Variable-number sample-path optimization. *Mathematical Programming Series B*, *117*, 81–109. (Cited on page 43.)

Dengiz, B. & Akbay, K. S. (2000). Computer simulation of a PCB production line: metamodeling approach. *International Journal of Production Economics*, *63*(2), 195–205. (Cited on page 19.)

Dhillon, B. S. (2002). *Engineering maintenance: A modern approach*. CRC Press. (Cited on page 97.)

Dhivya, M., Sundarambal, M., & Anand, L. N. (2011). Energy efficient computation of data fusion in wireless sensor networks using cuckoo-based particle approach (cbpa). *International Journal of Communications, Network and System Sciences*, *4*(4), 249–255. (Cited on page 19.)

Dolan, E. D. & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, *91*, 201–213. (Cited on page 38.)

Dorigo, M. & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, *344*(2–3), 243–278. (Cited on pages 24 and 35.)

Dorigo, M. & Stützle, T. (2004). *Ant colony optimization*. MIT Press, Cambridge, MA, USA. (Cited on pages 24 and 35.)

Doyle, P. S., Bibette, J., Bancaud, A., & Viovy, J.-L. (2002). Self-assembled magnetic matrices for DNA separation chips. *Science*, *295*(5563), 2237. (Cited on page 91.)

Driessen, L. T. (2006). *Simulation-based Optimization for Product and Process Design*. PhD thesis, Tilburg University. (Cited on page 12.)

Ernst, D., Glavic, M., Stan, G.-B., Mannor, S., & Wehenkel, L. (2007). The cross-entropy method for power system combinatorial optimization problems. In *Power Tech*, (pp. 1290–1295). IEEE. (Cited on page 19.)

Fahrenkopf, M. (2014). *Optimization, Dynamics and Stability of Non-linear Separation Processes*. PhD thesis, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15213, USA. (Cited on page 91.)

Feo, T. A. & Bard, J. F. (1989). Flight scheduling and maintenance base planning. *Management Science*, *35*(12), 1415–1432. (Cited on page 102.)

Ferris, M. C., Deng, G., Fryback, D. G., & Kuruchittham, V. (2005). Breast cancer epidemiology: Calibrating simulations via optimization. *Oberwolfach Reports*, *2*. (Cited on page 19.)

Figueira, G. & Almada-Lobo, B. (2014). Hybrid simulation-optimization methods: A taxonomy. *Simulation Modelling Practice and Theory*, *46*, 118–134. (Cited on page 14.)

Frazier, P., Powell, W., & Dayanik, S. (2009). The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, *21*(4), 599–613. (Cited on page 41.)

Frazier, P. I. (2009). *Knowledge-gradient methods for statistical learning*. PhD thesis, Princeton University. (Cited on page 12.)

Fu, M. C. (1994). Optimization via simulation: A review. *Annals of Operations Research*, *53*, 199–247. (Cited on pages 11, 12, 16, 18, 20, and 23.)

Fu, M. C. (2002). Optimization for Simulation: Theory vs. Practice. *SIAM Journal on Computing*, *14*(3), 192–215. (Cited on pages 11, 12, and 27.)

Fu, M. C., Andradóttir, S., Carson, J. S., Glover, F. W., Harrell, C. R., Ho, Y.-C., Kelly, J. P., & Robinson, S. M. (2000). Integrating optimization and simulation: Research and practice. In Joines, J. A., Barton, R. R., Kang, K., & Fishwick, P. A. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*. (Cited on pages 11, 12, and 39.)

Fu, M. C., Glover, F. W., & April, J. (2005). Simulation Optimization: A review, new developments, and applications. In Kuhl, M. E., Steiger, N. M., Armstrong, F. B., & Joines, J. A. (Eds.), *Proceedings of the 2005 Winter Simulation Conference*, (pp. 83–95). (Cited on pages 11, 12, and 72.)

Fu, M. C. & Hill, S. D. (1997). Optimization of discrete event systems via simultaneous perturbation stochastic approximation. *IIE Transactions*, *29*(233–243). (Cited on page 19.)

Fu, M. C., Hu, J., & Marcus, S. I. (1996). Model-based randomized methods for global optimization. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, (pp. 355–363)., Kyoto, Japan. (Cited on page 33.)

Fu, M. C. & Hu, J. Q. (1997). *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*. Kluwer Academic Publishers. (Cited on page 27.)

Gendreau, M. & Potvin, J.-Y. (2010). Tabu search. In *Handbook of Metaheuristics* (Second ed.)., volume 146 of *International Series in Operations Research & Management Science* (pp. 41–60). Springer. (Cited on page 31.)

Gerencsér, L., Kozmann, G., Vágó, Z., & Haraszti, K. (2002). The use of the spsa method in ecg analysis. *IEEE Transactions on Biomedical Engineering*, *49*(10), 1094–1101. (Cited on page 19.)

Gharbi, A. & Kenné, J.-P. (2005). Maintenance scheduling and production control of multiple-machine manufacturing systems. *Computers & Industrial Engineering*, *48*, 693–707. (Cited on page 102.)

Gittins, J. C. (1989). *Multi-armed bandit allocation indices*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons. (Cited on page 17.)

Glasserman, P. (1991). *Gradient Estimation via Perturbation Analysis*. Kluwer Academic Publishers. (Cited on page 27.)

Global Ethylene Database (accessed February 21, 2014). `http://www.industrialinfo.com/database/niche_markets/ethylene_database.jsp`. (Cited on page 98.)

Glover, F. (1990). Tabu search: A tutorial. *Interfaces*, *20*(4), 77–94. (Cited on page 31.)

Glover, F. & Hanafi, S. (2002). Tabu search and finite convergence. *Discrete Applied Mathematics*, *119*(1–2), 3–36. (Cited on page 24.)

Glover, F. & Laguna, M. (1997). *Tabu Search*. Kluwer Academic Publishers, Boston. (Cited on page 31.)

Glover, F. & Laguna, M. (2000). Fundamentals of scatter search and path relinking. *Control and Cybernetics*, *29*(3), 653–684. (Cited on pages 31 and 32.)

Goel, V. & Grossmann, I. E. (2006). A class of stochastic programs with decision dependent uncertainty. *Mathematical Programming*, *108*(2–3), 355–394. (Cited on page 151.)

Goldsman, D. & Nelson, B. L. (1998). Comparing systems via simulation. In J. Banks (Ed.), *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice* chapter 8. John Wiley & Sons. (Cited on page 23.)

Gong, W. B., Ho, Y. C., & Zhai, W. (49). Stochastic comparison algorithm for discrete optimization with estimation. *SIAM Journal on Optimization*, *10*, 384–404. (Cited on page 24.)

Graves, S., Kan, A. R., & Zipkin, P. (Eds.). (1992). *Logistics of Production and Inventory*, volume 4 of *Handbooks in Operations Research and Management Science*. Elsevier North-Holland, Amsterdam. (Cited on page 89.)

Grievink, J., Smit, K., Dekker, R., & van Rijn, C. F. H. (1993). Managing reliability and maintenance in the process industry. In *Foundations of Computer Aided Process Operations*, Crested Buttle Resort, CO. (Cited on page 98.)

Griewank, A. & Walther, A. (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation* (2nd ed.). Number 105 in Other Titles in Applied Mathematics. Philadelphia, PA: SIAM. (Cited on page 15.)

Gürkan, G., Ozge, A. Y., & Robinson, S. M. (1994). Sample path optimization in simulation. In Tew, J., Manivannan, S., Sadowski, D., & Seila, A. (Eds.), *Proceedings of 1994 Winter Simulation Conference*, (pp. 247–254). (Cited on page 29.)

Hajek, B. (1988). Cooling schedules for optimal annealing. *Mathematics of Operations Research*, *13*, 311–329. (Cited on page 31.)

Hall, J. D., Bowden, R. O., & Usher, J. M. (1996). Using evolution strategies and simulation to optimize a pull production system. *Journal of Materials Processing Technology*, *61*(1–2), 47–52. (Cited on page 19.)

Hansen, N. (2006). The CMA evolution strategy: A comparing review. In J. A. Lozano, P. Larrañaga, I. Inza, & E. Bengoetxea (Eds.), *Towards a new evolutionary computation. Advances on estimation of distribution algorithms* (pp. 75–102). Springer. (Cited on page 34.)

Hansen, N. (Current as of 8 June, 2011). *The CMA Evolution Strategy: A tutorial*. `http://www.lri.fr/~hansen/cmaesintro.html`. (Cited on pages 35 and 41.)

Healy, K. & Schruben, L. W. (1991). Retrospective simulation response optimization. In Nelson, B. L., Kelton, D. W., & Clark, G. M. (Eds.), *Proceedings of the 1991 Winter Simulation Conference*, (pp. 954–957). (Cited on pages 29 and 43.)

Hill, S. D. & Fu, M. C. (1995). Transfer optimization via simultaneous perturbation stochastic approximation. In Alexopoulos, C., Kang, K., Lilegdon, W. R., & Goldsman, D. (Eds.), *Proceedings of the 1995 Winter Simulation Conference*, (pp. 242–249). (Cited on page 19.)

Ho, Y.-C. (1999). An explanation of ordinal optimization: soft computing for hard problems. *Information Sciences*, *113*, 169–192. (Cited on page 23.)

Ho, Y. C. & Cao, X. R. (1991). *Discrete Event Dynamic Systems and Perturbation Analysis*. Kluwer Academic Publishers. (Cited on page 27.)

Hochberg, Y. & Tamhane, A. C. (1987). *Multiple Comparison Procedures*. John Wiley & Sons, New York. (Cited on page 23.)

Hong, L. J. & Nelson, B. L. (2006). Discrete optimization via simulation using COMPASS. *Operations Research*, *54*(1), 115–129. (Cited on page 24.)

Hong, L. J. & Nelson, B. L. (2009). A brief introduction to optimization via simulation. In Rossetti, M. D., Hill, R. R., Johansson, B., Dunkin, A., & Ingalls, R. G. (Eds.), *Proceedings of the 2009 Winter Simulation Conference*. (Cited on pages 11, 12, 20, 24, and 39.)

Hooke, R. & Jeeves, T. A. (1961). Direct search solution of numerical and statistical problems. *Journal of the Association for Computing Machinery*, *8*, 212–219. (Cited on page 29.)

Hsu, J. C. (1996). *Multiple Comparisons: theory and methods*. CRC Press, Boca Raton. (Cited on page 23.)

Hu, J., Fu, M. C., & Marcus, S. I. (2005). Stochastic optimization using model reference adaptive search. In Kuhl, M. E., Steiger, N. M., Armstrong, F. B., & Joines, J. A. (Eds.), *Proceedings of the 2005 Winter Simulation Conference*, (pp. 811–818). (Cited on pages 34 and 41.)

Hu, J., Fu, M. C., & Marcus, S. I. (2007). A model reference adaptive search method for global optimization. *Operations Research*, *55*(3), 549–568. (Cited on pages 34 and 41.)

Huang, D., Allen, T. T., Notz, W. I., & Zeng, N. (2006). Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization*, *34*, 441–466. (Cited on pages 26, 41, and 84.)

Humphrey, D. G. & Wilson, J. R. (2000). A revised simplex search procedure for stochastic simulation response-surface optimization. *INFORMS Journal on Computing*, *12*(4), 272–283. (Cited on page 29.)

Hunter, S. R. & Pasupathy, R. (2013). Optimal sampling laws for stochastically constrained simulation optimization on finite sets. *INFORMS Journal on Computing*, *25*(3), 527–542. (Cited on page 22.)

Hutchison, D. W. & Hill, S. D. (2001). Simulation optimization of airline delay with constraints. In Peters, B. A., Smith, J. S., Medeiros, D. J., & Rohrer, M. W. (Eds.), *Proceedings of the 2001 Winter Simulation Conference*, (pp. 1017–1022). (Cited on page 19.)

Huyer, W. & Neumaier, A. (2008). SNOBFIT – Stable noisy optimization by branch and fit. *ACM Transactions on Mathematical Software*, *35*, 1–25. (Cited on pages 41 and 84.)

Industrial Info Resources (accessed February 21, 2014). `www.industrialinfo.com`. (Cited on page 98.)

Irizarry, M. D. L. A., Wilson, J. R., & Trevino, J. (2001). A flexible simulation tool for manufacturing-cell design, II: response surface analysis and case study. *IIE Transactions*. (Cited on page 19.)

Jacobson, S. H. & Schruben, L. W. (1989). Techniques for simulation response optimization. *Operation Research Letters*, *8*, 1–9. (Cited on page 11.)

Jia, Q.-S., Ho, Y.-C., & Zhao, Q.-C. (2006). Comparison of selection rules for ordinal optimization. *Mathematical and Computer Modelling*, *43*(9–10), 1150–1171. (Cited on page 24.)

Jones, D. R., Perttunen, C. D., & Stuckman, B. E. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Application*, *79*, 157–181. (Cited on page 35.)

Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, *13*, 455–492. (Cited on pages 26 and 84.)

Jung, J. Y., Blau, G., Pekny, J. F., Reklaitis, G. V., & Eversdyk, D. (2004). A simulation based optimization approach to supply chain management under demand uncertainty. *Computers & Chemical Engineering*, *28*, 2087–2106. (Cited on pages 19 and 116.)

Kabirian, A. (2009). *Continuous optimization via simulation using Golden Region search*. PhD thesis, Iowa State University. (Cited on page 12.)

Kabirian, A. & Ólafsson, S. (2007). Allocation of simulation runs for simulation optimization. In Henderson, S. G., Biller, B., Hsieh, M.-H., Shortle, J., Tew, J. D., & Barton, R. R. (Eds.), *Proceedings of the 2007 Winter Simulation Conference*, (pp. 363–371). (Cited on page 41.)

Kabirian, A. & Ólafsson, S. (2011). Continuous optimization via simulation using golden region search. *European Journal of Operational Research*. (Cited on page 41.)

Kenne, J. P. & Gharbi, A. (2001). A simulation optimization approach in production planning of failure prone manufacturing systems. *Journal of Intelligent Manufacturing*, *12*, 421–431. (Cited on page 19.)

Khan, H. A., Zhang, Y., Ji, C., Stevens, C. J., Edwards, D. J., & O'Brien, D. (2006). Optimizing polyphase sequences for orthogonal netted radar. *Signal Processing Letters, IEEE*, *13*(10), 589–592. (Cited on page 19.)

Kiefer, J. & Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, *23*(3), 462–466. (Cited on page 27.)

Kim, J. M. & Doyle, P. S. (2007). Brownian dynamics simulations of a DNA molecule colliding with a small cylindrical post. *Macromolecules*, *40*(25), 9151–9163. (Cited on page 91.)

Kim, S.-H. (2005). Comparison with a standard via fully sequential procedures. *ACM Transactions on Modeling and Computer Simulation*, *15*(2), 155–174. (Cited on page 23.)

Kim, S.-H. & Nelson, B. L. (2006). Selecting the best system. In S. G. Henderson & B. L. Nelson (Eds.), *Handbooks in Operations Research and Management Science: Simulation* chapter 17, (pp. 501–534). Elsevier Science. (Cited on page 22.)

Kim, S.-H. & Nelson, B. L. (2007). Recent advances in ranking and simulation. In Henderson, S. G., Biller, B., Hsieh, M.-H., Shortle, J., Tew, J. D., & Barton, R. R. (Eds.), *Proceedings of the 2007 Winter Simulation Conference*, (pp. 162–172). (Cited on page 22.)

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680. (Cited on page 30.)

Kleijnen, J. P. C. (1993). Simulation and optimization in production planning: A case study. *Decision Support Systems*, *9*, 269–280. (Cited on page 19.)

Kleijnen, J. P. C. (2008). *Design and Analysis of Simulation Experiments*. Springer, New York. (Cited on pages 12 and 25.)

Kleijnen, J. P. C. (2009). Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, *192*(3), 707–716. (Cited on page 26.)

Kleijnen, J. P. C., Beers, W. C. M., & van Nieuwenhuyse, I. (2012). Expected improvement in efficient global optimization through bootstrapped kriging. *Journal of Global Optimization*, *54*(1), 59–73. (Cited on page 26.)

Kleijnen, J. P. C. & van Beers, W. C. M. (2005). Robustness of kriging when interpolating in random simulation with heterogeneous variances: Some experiments. *European Journal of Operational Research*, *165*, 826–834. (Cited on page 26.)

Kleinman, N. L., Hill, S. D., & Ilenda, V. A. (1997). Spsa/simmod optimization of air traffic delay cost. In *Proceedings of the 1997 American Control Conference*, volume 2, (pp. 1121–1125). (Cited on page 19.)

Köchel, P. & Nieländer, U. (2005). Simulation-based optimisation of multi-echelon inventory systems. *International Journal of Production Economics*, *93–94*, 505–513. (Cited on page 19.)

Kolda, T. G., Lewis, R. M., & Torczon, V. J. (2003). Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, *45*, 385–482. (Cited on page 29.)

Kondili, E., Pantelides, C. C., & Sargent, R. W. H. (1993). A general algorithm for short-term scheduling of batch operations. part i–mathematical formulation. *Computers & Chemical Engineering*, *17*, 211–227. (Cited on page 107.)

Kothandaraman, G. & Rotea, M. A. (2005). Simultaneous-perturbation-stochastic-approximation algorithm for parachute parameter estimation. *Journal of Aircraft*, *42*(5), 1229–1235. (Cited on page 19.)

Kroese, D. P., Hui, K.-P., & Nariai, S. (2007). Network reliability optimization via the cross-entropy method. *Reliability, IEEE Transactions on*, *56*(2), 275–287. (Cited on page 19.)

Kroese, D. P., Porotsky, S., & Rubinstein, R. Y. (2006). The cross-entropy method for continuous multi-extremal optimization. *Methodology and Computing in Applied Probability*, *8*(3), 383–407. (Cited on pages 34 and 41.)

Kulturel-Konak, S. & Konak, A. (2010). Simulation optimization embedded particle swarm optimization for reliable server assignment. In Johansson, B., Jain, S., Montoya-Torres, J., Hugan, J., & Yücesan, E. (Eds.), *Proceedings of the 2010 Winter Simulation Conference*, (pp. 2897–2906). (Cited on page 19.)

Laggoune, R., Chateauneuf, A., & Aissani, D. (2009). Opportunistic policy for optimal preventive maintenance of a multi-component system in continuous operating units. *Computers & Chemical Engineering*, *33*, 1499–1510. (Cited on pages 100 and 102.)

Larrañaga, P. & Lozano, J. A. (2002). *Estimation of distribution algorithms: A new tool for evolutionary computation*. Kluwer Academic Publishers. (Cited on page 32.)

Larson, J. & Billups, S. C. (2014). Stochastic derivative-free optimization using a trust region framework. In review. (Cited on pages 43 and 44.)

Larson, J. M. (2012). *Derivative-free optimization of noisy functions*. PhD thesis, University of Colorado Denver. (Cited on page 43.)

Lau, T. W. E. & Ho, Y.-C. (1997). Universal alignment probabilities and subset selection for ordinal optimization. *Journal of Optimization Theory and Applications*, *93*(3), 455–489. (Cited on page 23.)

Law, A. M. & Kelton, W. D. (2000). *Simulation modeling and analysis* (3rd ed.). McGraw-Hill, Singapore. (Cited on page 36.)

Lee, L. H., Pujowidianto, N. A., Li, L.-W., Chen, C.-H., & Yap, C. M. (2012). Approximate simulation budget allocation for selecting the best design in the presence of stochastic constraints. *IEEE Transactions on Automatic Control*, *57*(11), 2940–2945. (Cited on page 22.)

Lehmann, E. L. & Romano, J. P. (2005). *Testing Statistical Hypotheses* (Third ed.). Springer. (Cited on pages 50 and 65.)

Lenahan, T. (1999). *Turnaround Management*. Butterworth Heinemann. (Cited on pages 4 and 134.)

Li, Y. (2009). A simulation-based evolutionary approach to LNA circuit design optimization. *Applied Mathematics and Computation*, *209*(1), 57–67. (Cited on page 19.)

Li, Z. & Ierapetritou, M. (2008). Process scheduling under uncertainty: Review and challenges. *Computers & Chemical Engineering*, *32*, 715–727. (Cited on page 135.)

Li, Z., Tang, Q., & Floudas, C. A. (2012). A comparative theoretical and computational study on robust counterpart optimization. II. Probabilistic guarantees on constraint satisfaction. *Industrial & Engineering Chemistry Research*, *51*, 6769–6788. (Cited on page 135.)

Lin, X., Janak, S. L., & Floudas, C. A. (2004). A new robust optimization approach for scheduling under uncertainty: I. Bounded uncertainty. *Computers & Chemical Engineering*, *28*, 1069–1085. (Cited on page 135.)

Lucidi, S. & Sciandrone, M. (2002). On the global convergence of derivative-free methods for unconstrained minimization. *SIAM Journal on Optimization*, *13*, 97–116. (Cited on page 29.)

Lutz, C. M., Davis, K. R., & Sun, M. (1998). Determining buffer location and size in production lines using tabu search. *European Journal of Operational Research*, *106*, 301–316. (Cited on page 19.)

Maggioni, F., Allevi, E., & Bertocchi, M. (2012). Measures of information in multi-stage stochastic programming. Number 2 in Stochastic Programming E-Print Series. Institut für Mathematik. unpublished. (Cited on page 160.)

Martí, R., Laguna, M., & Glover, F. (2006). Principles of scatter search. *European Journal of Operational Research*, *169*(2), 359–372. (Cited on page 32.)

Maryak, J. L. & Chin, D. C. (2008). Global random optimization by simulataneous perturbation stochastic approximation. *IEEE Transactions on Automatic Control*, *53*, 780–783. (Cited on page 28.)

Meagher, R. J., Won, J., Coyne, J. A., Li, J., & Barron, A. E. (2008). Sequencing of DNA by free-solution capillary electrophoresis using a genetically engineered protein polymer drag-tag. *Analytical Chemistry*, *80*, 2842–2848. (Cited on page 91.)

Megow, N., Möhring, R. H., & Schulz, J. (2011). Decision support and optimization in shutdown and turnaround scheduling. *INFORMS Journal on Computing*, *23*(2), 189–204. (Cited on pages 100 and 135.)

Meketon, M. S. (1987). Optimization in simulation: A survey of recent results. In Thesen, A., Grant, H., & Kelton, W. D. (Eds.), *Proceedings of the 1987 Winter Simulation Conference*, (pp. 58–67). (Cited on pages 11, 12, and 14.)

Merhof, D., Soza, G., Stadlbauer, A., Greiner, G., & Nimsky, C. (2007). Correction of susceptibility artifacts in diffusion tensor data using non-linear registration. *Medical Image Analysis*, *11*(6), 588–603. (Cited on page 19.)

Merton, R. C. (1974). On the pricing of corporate debt: The risk structure of interest rates. *The Journal of Finance*, *29*(2), 449–470. (Cited on page 18.)

Mishra, V., Bhatnagar, S., & Hemachandra, N. (2007). Discrete parameter simulation optimization algorithms with applications to admission control with dependent service times. In *Proceedings of the 46th IEEE Conference on Decision and Control*, (pp. 2986–2991)., New Orleans, LA. (Cited on page 19.)

Mockus, J. (1989). *Bayesian Approach to Global Optimization*. Kluwer Academic Publishers. (Cited on page 26.)

Mockus, J., Tiesis, V., & Zilinskas, A. (1978). *Towards Global Optimisation*, volume 2, chapter The application of Bayesian Methods for Seeking the Extremum, (pp. 117–128). North-Holland, Amsterdam. (Cited on page 26.)

Moré, J. & Wild, S. (2009). Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, *20*, 172–191. (Cited on page 38.)

Moudani, W. E. & Mora-Camino, F. (2000). A dynamic approach for aircraft assignment and maintenance scheduling by airlines. *Journal of Air Transport Management*, *6*(4), 233–237. (Cited on page 102.)

Mukerji, R. (1991). Power plant maintenance scheduling: optimizing economics and reliability. *IEEE Transactions on Power Systems*, *6*(2), 476–483. (Cited on page 102.)

Myers, R. H., Montgomery, D. C., & Anderson-Cook, C. M. (2009). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Wiley Series in Probability and Statistics. John Wiley & Sons. (Cited on page 26.)

Narayan, V. (2004). *Effective Maintenance Management: Risk and Reliability Strategies for Optimizing Performance*. Industrial Press, Inc. (Cited on page 4.)

Neddermeijer, H. G., Oortmarssen, G. J. V., Piersma, N., & Dekker, R. (2000). A framework for response surface methodology for simulation optimization. In Joines, J. A., Barton, R. R., Kang, K., & Fishwick, P. A. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*, (pp. 129–136). (Cited on page 26.)

Nelder, J. A. & Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, *7*, 308–313. (Cited on page 29.)

Nelson, B. L. (2010). Optimization via simulation over discrete decision variables. *Tutorials in operations research*, *7*, 193–207. (Cited on page 24.)

Nelson, B. L. & Goldsman, D. (2001). Comparisons with a standard in simulation experiments. *Management Science*, *47*(3), 449–463. (Cited on page 23.)

Nicolai, R. & Dekker, R. (2009). Automated response surface methodology for simulation optimization models with unknown variance. *Quality Technology and Quantitative Management*, *6*(3), 325–352. (Cited on page 26.)

Nocedal, J. & Wright, S. (1999). *Numerical Optimization*. Springer-Verlag. (Cited on page 49.)

North American Chemical Processing Industry Maintenance Turnarounds (accessed February 21, 2014). 2008 north american chemical processing industry maintenance turnarounds. `http://www.industrialinfo.com/media/downloadMedia.jsp?mediaId=344733`. (Cited on page 4.)

Ö. Yalçinkaya, G. M. B. (2009). Modelling and optimization of average travel time for a metro line by simulation and response surface methodology. *European Journal of Operational Research*, *196*, 225–233. (Cited on page 19.)

Ólafsson, S. (2006). Metaheuristics. In S. Henderson & B. Nelson (Eds.), *Handbook in Operations Research and Management Science: Simulation 13* (pp. 633–654). Elsevier. (Cited on page 24.)

Olson, D. W., Dutta, S., Laachi, N., Tian, M., & Dorfman, K. D. (2011). Continuous-time random walk models of DNA electrophoresis in a post array: Part II. mobility and sources of band broadening. *Electrophoresis*, *32*, 581–587. (Cited on page 91.)

OR/MS today 2013 survey (2013). `http://www.orms-today.org/surveys/Simulation/Simulation.html`. (Cited on page 36.)

Osorio, C. & Bierlaire, M. (2010). A simulation-based optimization approach to perform urban traffic control. In *Proceedings of the Triennial Symposium on Transportation Analysis*. (Cited on pages 19 and 27.)

Ou, J., Cho, J., Olson, D. W., & Dorfman, K. D. (2009). DNA electrophoresis in a sparse ordered post array. *Physical Review E*, *79*, 061904. (Cited on page 91.)

Pasupathy, R. & Ghosh, S. (2013). Simulation optimization: A concise overview and implementation guide. *Tutorials in operations research*, *10*, 122–150. (Cited on page 11.)

Pasupathy, R. & Henderson, S. G. (2011). SIMOPT: A library of simulation-optimization problems. In Jain, S., Creasey, R. R., Himmelspach, J., White, K. P., & M. Fu, e. (Eds.), *Proceedings of the 2011 Winter Simulation Conference*. (Cited on pages 21 and 39.)

Pasupathy, R. & Kim, S. (2011). The stochastic root finding problem: Overview, solutions, and open questions. *ACM Transactions on Modeling and Computer Simulation*, *21*(3), 19:1–19:23. (Cited on page 28.)

Patel, P. D. & Shaqfeh, E. S. G. (2003). A computational study of DNA separations in sparse disordered and periodic arrays of posts. *The Journal of Chemical Physics*, *118*(6), 2941–2951. (Cited on page 91.)

Peters, J., Vijayakumar, S., & Schaal, S. (2003). Reinforcement learning for humanoid robotics. In *Third IEEE-RAS International Conference on Humanoid Robots*, (pp. 1–20)., Karlsruhe, Germany. (Cited on page 17.)

Pflug, G. C. (1996). *Optimization of stochastic models: The interface between simulation and optimization*. Kluwer Academic Publishers. (Cited on page 27.)

Pistikopoulos, E. N., Vassiliadis, C. G., Arvela, J., & Papageorgiou, L. G. (2001). Interactions of maintenance and production planning for multipurpose process plants—a system effectiveness approach. *Industrial & Engineering Chemistry Research*, *40*, 3195–3207. (Cited on pages 99, 102, and 135.)

Powell, M. J. D. (2002). UOBYQA: unconstrained optimization by quadratic approximation. *Mathematical Programming*, *92*, 555–582. (Cited on page 43.)

Powell, M. J. D. (2009). The BOBYQA algorithm for bound constrained optimization without derivatives. Technical report, Department of Applied Mathematics and Theoretical Physics, University of Cambridge. (Cited on page 44.)

Powell, W. B. (accessed October 23, 2013). `http://www.castlelab.princeton.edu/cso.htm`. (Cited on page 17.)

Powell, W. B. & Ryzhov, I. O. (2012). *Optimal Learning*. John Wiley & Sons. (Cited on page 17.)

Prakash, P., Deng, G., Converse, M. C., Webster, J. G., Mahvi, D. M., & Ferris, M. C. (2008). Design optimization of a robust sleeve antenna for hepatic microwave ablation. *Physics in Medicine and Biology*, *53*, 1057–1069. (Cited on page 19.)

Radac, M.-B., Precup, R.-E., Petriu, E. M., & Preitl, S. (2011). Application of ift and spsa to servo system control. *IEEE Transactions on Neural Networks*, *22*(12), 2363–2375. (Cited on page 19.)

Rall, L. B. (1981). *Automatic Differentiation: Techniques and Applications*, volume 120 of *Lecture Notes in Computer Science*. Berlin: Springer. (Cited on page 15.)

Ramanathan, S. P., Mukherjee, S., Dahule, R. K., Ghosh, S., Rahman, I., Tambe, S. S., Ravetkar, D. D., & Kulkarni, B. D. (2001). Optimization of continuous distillation columns using stochastic optimization approaches. *Transactions of the Institution of Chemical Engineers*, *79*, 310–322. (Cited on page 19.)

Rasmussen, C. E. & Nickisch, H. (2011). *The GPML Toolbox version 3.1*. Available at `http://www.gaussianprocess.org/gpml/code/matlab/doc/`. (Cited on page 77.)

Rasmussen, C. E. & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press. (Cited on pages 26, 47, 49, and 74.)

Reeves, C. R. (1997). Genetic algorithms for the operations researcher. *INFORMS Journal on Computing*, *9*(3), 231–250. (Cited on page 30.)

Ren, H., Karger, A. E., Oaks, F., Menchen, S., Slater, G. W., & Drouin, G. (1999). Separating DNA sequencing fragments without a sieving matrix. *Electrophoresis*, *20*(12), 2501–2509. (Cited on page 90.)

Renotte, C. & Vande Wouwer, A. (2003). Stochastic approximation techniques applied to parameter estimation in a biological model. In *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2003. Proceedings of the Second IEEE International Workshop on*, (pp. 261–265). IEEE. (Cited on page 19.)

Rios, L. M. & Sahinidis, N. V. (2012). `mydfo`—*A unifying interface to derivative-free optimization solvers*. Carnegie Mellon University. (Cited on page 167.)

Rios, L. M. & Sahinidis, N. V. (2013). Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization*, *56*, 1247–1293. (Cited on pages 20, 36, 39, 83, and 84.)

Robbins, H. & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, *22*(3), 400–407. (Cited on page 27.)

Rockafellar, R. T. & Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, *26*(7), 1443–1471. (Cited on page 138.)

Romero, P. A., Krause, A., & Arnold, F. H. (2013). Navigating the protein fitness landscape with gaussian processes. *Proceedings of the National Academy of Sciences (PNAS)*, *110*(3). (Cited on page 19.)

Roustant, O., Ginsbourger, D., & Deville, Y. (2012). Dicekriging, diceoptim: Two r packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, *51*(1), 1–55. (Cited on page 41.)

Rubinstein, R. (1999). The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, *1*, 127–190. (Cited on page 34.)

Rubinstein, R. Y. & Kroese, D. P. (2004). *The Cross-Entropy Method: A unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*. Springer-Verlag, New York. (Cited on pages 12, 33, and 34.)

Rubinstein, R. Y. & Shapiro, A. (1993). *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. John Wiley & Sons. (Cited on pages 27 and 29.)

Rudin, W. (1976). *Principles of Mathematical Analysis*. McGraw-Hill, New York. (Cited on page 62.)

Sacks, J., Schiller, S. B., & Welch, W. J. (1989). Designs for computer experiments. *Technometrics*, *31*, 41–47. (Cited on page 26.)

Safizadeh, M. H. (1990). Optimization in simulation: Current issues and the future outlook. *Naval Research Logistics*, *37*, 807–825. (Cited on page 11.)

Sahinidis, N. V. (2004). Optimization under uncertainty: State-of-the-art and opportunities. *Computers & Chemical Engineering*, *28*(6–7), 971–983. (Cited on page 15.)

Sahinidis, N. V. (2013). *BARON 12.1.0: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual. (Cited on page 82.)

Sanger, F., Nicklen, S., & Coulson, A. R. (1977). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, *74*(12), 5463–5467. (Cited on page 90.)

Sanmartí, E., Espuña, A., & Puigjaner, L. (1997). Batch production and preventive maintenance scheduling under equipment failure uncertainty. *Computers & Chemical Engineering*, *21*(10), 1157–1168. (Cited on pages 99, 102, and 135.)

Satoh, T. & Nara, K. (1991). Maintenance scheduling by using simulated annealing method [for power plants]. *IEEE Transactions on Power Plants*, *6*(2), 850–857. (Cited on page 102.)

Schwartz, J. D., Wang, W., & Rivera, D. E. (2006). Simulation-based optimization of process control policies for inventory management in supply chains. *Automatica*, *42*, 1311–1320. (Cited on page 19.)

Scott, W., Frazier, P. I., & Powell, W. (2011). The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. *SIAM Journal on Optimization*, *21*(3), 996–1026. (Cited on pages 26 and 41.)

Settles, B. (2010). Active learning literature survey. Technical report, University of Wisconsin–Madison. (Cited on page 16.)

Shapiro, A. (1991). Asymptotic analysis of stochastic programs. *Annals of Operations Research*, *30*, 169–186. (Cited on pages 29 and 43.)

Shapiro, A. (1996). Simulation based optimization. In Charnes, J. M., Morrice, D. J., Brunner, D. T., & Swain, J. J. (Eds.), *Proceedings of the 1996 Winter Simulation Conference*, (pp. 332–336). (Cited on page 29.)

Shi, L. & Ólafsson (2007). *Nested Partitions Optimization: Methodology and Applications*, volume 109 of *International Series in Operations Research & Management Science*. Springer. (Cited on page 24.)

Shi, L. & Ólafsson, S. (2000). Nested partitions method for stochastic optimization. *Methodology and Computing in Applied Probability*, *2*, 271–291. (Cited on page 24.)

Sloan, T. W. & Shanthikumar, J. G. (2000). Combined production and maintenance scheduling for a multiple-product, single-machine production system. *Production and Operations Management*, *9*(4), 379–399. (Cited on page 102.)

Song, Y. & Grizzle, J. W. (1995). The extended kalman filter as a local asymptotic observer for discrete-time nonlinear systems. *Journal of Mathematical Systems, Estimation, and Control*, *5*(1), 59–78. (Cited on page 18.)

Soyster, A. L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, *21*, 1154–1157. (Cited on page 153.)

Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, *37*, 332–341. (Cited on page 28.)

Spall, J. C. (2003a). Chapter 7: Simultaneous Perturbation Stochastic Approximation. In *Introduction to stochastic search and optimization: Estimation, simulation, and control*. Wiley-Interscience. (Cited on pages 27, 41, and 84.)

Spall, J. C. (2003b). *Introduction to stochastic search and optimization: Estimation, simulation, and control*. Wiley-Interscience. (Cited on page 11.)

Spall, J. C. (2009). Feedback and weighting mechanisms for improving Jacobian estimates in the adaptive simultaneous perturbation algorithm. *IEEE Transactions on Automatic Control*, *54*(6), 1216–1229. (Cited on page 28.)

Spall, J. C. (2012). Stochastic optimization. In J. E. Gentle, W. K. Härdle, & Y. Mori (Eds.), *Handbook of Computational Statistics: Concepts and Methods* (Second ed.). chapter 7, (pp. 173–201). Springer. (Cited on page 28.)

Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. (2012). Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, *58*(5), 3250–3265. (Cited on page 26.)

Sriram, C. & Haghani, A. (2003). An optimization model for aircraft maintenance scheduling and re-assignment. *Transportation Research Part A: Policy and Practice*, *37*(1), 29–48. (Cited on page 102.)

Srivastava, R. & Rawlings, J. B. (2014). Parameter estimation in stochastic chemical kinetic models using derivative-free optimization and bootstrapping. *Computers & Chemical Engineering*, *63*, 152–158. (Cited on page 167.)

Stadtler, H. (2000). Improved rolling horizon schedules for the dynamic single-level lot-sizing problem. *Management Science*, *46*(2), 318–326. (Cited on page 122.)

Stephens, C. P. & Baritompa, W. (1998). Global optimization requires global information. *Journal of Optimization Theory and Applications*, *96*, 575–588. (Cited on page 16.)

Sundararajan, S. & Keerthi, S. (2001). Predictive approaches for choosing hyperparameters in gaussian processes. *Neural Computation*, *13*(5), 1103–1118. (Cited on page 77.)

Swisher, J. R., Hyden, P. D., Jacobson, S. H., & Schruben, L. W. (2000). A survey of simulation optimization techniques and procedures. In Joines, J. A., Barton, R. R., Kang, K., & Fishwick, P. A. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*. (Cited on page 11.)

Syberfeldt, A. & Lidberg, S. (2012). Real-world simulation-based manufacturing optimization using cuckoo search. In Laroque, C., Himmelspach, J., Pasupathy, R., Rose, O., & Uhrmacher, A. (Eds.), *Proceedings of the 2012 Winter Simulation Conference*. (Cited on page 19.)

Tan, J. S. & Kramer, M. A. (1997). A general framework for preventive maintenance optimization in chemical process operations. *Computers & Chemical Engineering*, *21*(12), 1451–1469. (Cited on pages 98, 99, and 102.)

Tawarmalani, M. & Sahinidis, N. V. (2005). A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, *103*, 225–249. (Cited on page 82.)

Tein, L. H. & Ramli, R. (2010). Recent advancements of nurse scheduling models and a potential path. In *Proceedings of the 6th IMT-GT Conference on Mathematics, Statistics and its Applications*, (pp. 395–409). (Cited on page 19.)

Tekin, E. & Sabuncuoglu, I. (2004). Simulation optimization: A comprehensive review on theory and applications. *IIE Transactions*, *36*, 1067–1081. (Cited on pages 11, 12, 23, 27, 36, and 39.)

Teng, S., Lee, L. H., & Chew, E. P. (2007). Multi-objective ordinal optimization for simulation optimization problems. *Automatica*, *43*(11), 1884–1895. (Cited on page 24.)

The Gaussian Processes Website (Accessed on September 11, 2014). The Gaussian Processes Website. `http://www.gaussianprocess.org/#code`. (Cited on page 77.)

Trosset, M. W. (2000). On the use of direct search methods for stochastic optimization. Technical report, Rice University, Houston, TX. (Cited on page 29.)

van Beers, A. C. & Kleijnen, J. P. C. (2004). Kriging interpolation in simulation: A survey. *Proceedings of the 2004 Winter Simulation Conference*, *1*, 121–129. (Cited on page 26.)

Vande Wouwer, A., Renotte, Đ., Bogaerts, P., & Remy, M. (2001). Application of spsa techniques in nonlinear system identification. In *Proceedings of the European Control Conference*, (pp. 2835). (Cited on page 19.)

Volkanovski, A., Mavko, B., Boševski, T., Čaševski, A., & Čepin, M. (2008). Genetic algorithm optimisation of the maintenance scheduling of generating units in a power system. *Reliability Engineering & System Safety*, *93*(6), 779–789. (Cited on page 102.)

Volkmuth, W. D. & Austin, R. H. (1992). DNA electrophoresis in microlithographic arrays. *Nature*, *358*, 600–602. (Cited on page 91.)

Vujanic, R., Mariéthoz, S., Goulart, P., & Morari, M. (2012). Robust integer optimization and scheduling problems for large electricity consumers. In *2012 American Control Conference*, (pp. 3108–3113). (Cited on page 135.)

Wang, H., Pasupathy, R., & Schmeiser, B. W. (2012). Integer-ordered simulation optimization using r-spline: Retrospective search with piecewise-linear interpolation and neighborhood enumeration. *ACM Transactions on Modeling and Computer Simulation*, *23*, 17:1–17:24. (Cited on pages 24 and 41.)

Wang, Q. & Spall, J. C. (2011). Discrete simultaneous perturbation stochastic approximation on loss functions with noisy measurements. In *Proceedings of the American Control Conference*, (pp. 4520–4525)., San Francisco. IEEE. (Cited on page 28.)

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, *4*, 65–85. (Cited on page 30.)

Wild, S. M. (2009). *Derivative-free optimization algorithms for computationally expensive functions*. PhD thesis, Cornell University. (Cited on page 56.)

Wild, S. M., Regis, R. G., & Shoemaker, C. A. (2008). ORBIT: Optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing*, *30*, 3197–3219. (Cited on pages 43, 44, and 53.)

Wild, S. M. & Shoemaker, C. A. (2011). Global convergence of radial basis function trust region derivative-free algorithms. *SIAM Journal on Optimization*, *21*(3), 761–781. (Cited on pages 46 and 48.)

Wittmann-Hohlbein, M. & Pistikopoulos, E. N. (2013). Proactive scheduling of batch processes by a combined robust optimization and multiparametric programming approach. *AIChE Journal*, *59*(11), 4184–4211. (Cited on page 135.)

Wu, Y., Nyström-Lahti, M., Osinga, J., Looman, M. W. G., Peltomäki, P., Aaltonen, L. A., de la Chapelle, A., Hofstra, R. M. W., & Buys, C. H. C. M. (1997). *MSH*2 and *MLH*1 mutations in sporadic replication error-positive colorectal carcinoma as assessed by two-dimensional DNA electrophoresis. *Genes, Chromosomes and Cancer*, *18*(4), 269–278. (Cited on page 90.)

Xie, J. & Frazier, P. I. (2013). Sequential bayes-optimal policies for multiple comparisons with a known standard. *Operations Research*, *61*(5), 1174–1189. (Cited on page 23.)

Xie, J., Frazier, P. I., Sankaran, S., Marsden, A., & Elmohamed, S. (2012). Optimization of computationally expensive simulations with gaussian processes and parameter uncertainty: Application to cardiovascular surgery. In *50th Annual Allerton Conference on Communication, Control, and Computing*. (Cited on page 19.)

Xing, X. Q. & Damodaran, M. (2002). Assessment of simultaneous perturbation stochastic approximation method for wing design optimization. *Journal of Aircraft*, *39*, 379–381. (Cited on page 19.)

Xing, X. Q. & Damodaran, M. (2005a). Application of simultaneous perturbation stochastic approximation method for aerodynamic shape design optimization. *AIAA Journal*, *43*(2), 284–294. (Cited on page 19.)

Xing, X. Q. & Damodaran, M. (2005b). Inverse design of transonic airfoils using parallel simultaneous perturbation stochastic approximation. *Journal of Aircraft*, *42*(2), 568–570. (Cited on page 19.)

Xu, J., Nelson, B. L., & Hong, L. J. (2010). Industrial strength COMPASS: A comprehensive algorithm and software for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation*, *20*(1), 1–29. (Cited on pages 20 and 41.)

Xu, J., Nelson, B. L., & Hong, L. J. (2013). An adaptive hypberbox algorithm for high-dimensional discrete optimization via simulation problems. *INFORMS Journal on Computing*, *25*(1), 133–146. (Cited on page 24.)

Yamayee, Z. A. (1982). Maintenance scheduling: Description, literature survey, and interface with overall operations scheduling. *IEEE Transactions on Power Apparatus and Systems*, *PAS-101*(8), 2770–2779. (Cited on page 102.)

Yan, D. & Mukai, H. (1992). Stochastic discrete optimization. *SIAM Journal on Control and Optimization*, *30*, 594–612. (Cited on page 24.)

Yang, X.-S. & Deb, S. (2010). Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation*, *1*(4), 330–343. (Cited on page 19.)

Yang, Z., Djurdjanovic, D., & Ni, J. (2008). Maintenance scheduling in manufacturing systems based on predicted machine degradation. *Journal of Intelligent Manufacturing*, *19*(1), 87–98. (Cited on page 102.)

Yao, X. (2004). Optimal preventive maintenance scheduling in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, *17*(3), 345–356. (Cited on page 102.)

Yeomans, J. S. (2007). Solid waste planning under uncertainty using evolutionary simulation-optimization. *Socio-Economic Planning Sciences*, *41*, 38–60. (Cited on page 19.)

Yun, I. & Park, B. (2010). Application of stochastic optimization method for an urban corridor. In Perrone, L. F., Wieland, F. P., Liu, J., Lawson, B. G., Nicol, D. M., & Fujimoto, R. M. (Eds.), *Proceedings of the 2010 Winter Simulation Conference*, (pp. 1493–1499). (Cited on page 19.)