MASTER'S THESIS

Sensitivity Analysis and Discrete Stochastic Optimization for Semiconductor Manufacturing Systems

by Praveen V. Mellacheruvu
Advisor: Jeffrey W. Herrmann, Michael C. Fu

M.S. 2000-2

ABSTRACT

Title of Thesis: SENSITIVITY ANALYSIS AND DISCRETE STOCHASTIC OPTIMIZATION FOR SEMICONDUCTOR MANUFACTURING SYSTEMS

Degree candidate: Praveen V. Mellacheruvu

Degree and year: Master of Science, 2000

Thesis directed by: Assistant Professor Jeffrey W. Herrmann
Professor Michael C. Fu
Institute for Systems Research

The semiconductor industry is a capital-intensive industry with rapid time-to-market, short product development cycles, complex product flows and other characteristics. These factors make it necessary to utilize equipment efficiently and reduce cycle times. Further, the complexity and highly stochastic nature of these manufacturing systems make it difficult to study their characteristics through analytical models. Hence we resort to simulation-based methodologies to model these systems.

This research aims at developing and implementing simulation-based operations research techniques to facilitate System Control (through sensitivity analysis) and System Design (through optimization) for semiconductor manufacturing systems.

Sensitivity analysis for small changes in input parameters is performed using gradient estimation techniques. Gradient estimation methods are evaluated by studying the state of the art and comparing the finite difference method and simultaneous perturbation method by applying them to a stochastic manufacturing system. The results are compared with the gradients obtained through analytical queueing models. The finite difference method is implemented in a heterogeneous simulation environment (HSE) based decision support tool for process engineers. This tool performs heterogeneous simulations and sensitivity analyses.

The gradient-based techniques used for sensitivity analysis form the building blocks for a gradient-based discrete stochastic optimization procedure. This procedure is applied to the problem of allocating a limited budget to machine purchases to achieve throughput requirements and minimize cycle time. The performance of the algorithm is evaluated by applying the algorithm on a wide range of problem instances.

SENSITIVITY ANALYSIS AND DISCRETE STOCHASTIC OPTIMIZATION FOR SEMICONDUCTOR MANUFACTURING SYSTEMS

by

Praveen V. Mellacheruvu

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2000

Advisory Committee:

Assistant Professor Jeffrey W. Herrmann, Chairman/Advisor
Professor Michael C. Fu, Chairman/Advisor
Professor Gary W. Rubloff

Dedication

To my parents

Acknowledgments

I would like to thank Dr.Herrmann and Dr.Fu for giving me an opportunity to conduct research under their guidance. I would also like to thank them for helping me appreciate Industrial Engineering and Operations Research better. They will have a lifelong influence on many facets of my personality, especially my writing skills! I express my gratitude to Dr.Rubloff for providing me valuable advice and giving me the big picture of the fascinating semiconductor industry. I also acknowledge the support provided by NSF and SRC for this research. I express my appreciation for the input provided by my colleagues Laurent, Quan and Mandar. I would like to thank my roommates Kartik and Makesh for standing by me through the last year and the CIM Lab people, including Dr. Lin, for giving me a great work environment.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Simulation modeling is a powerful tool that can be effectively leveraged to model, analyze and optimize systems. When we consider manufacturing systems and their inherent stochastic nature, simulation is particularly useful to predict their behavior. This research focuses on developing and applying simulation-based operations research techniques to a specific class of manufacturing systems (semiconductor manufacturing systems) to analyze and study their behavior. This aim of this research is to facilitate System Control (through sensitivity analysis) and System Design (through optimization) of manufacturing systems through operations research.

In the semiconductor industry, a lot of research attention is focused on the effective use of equipment and reduction of cycle times as the industry invests large sums of money in chip-making equipment. Decision support tools which aid managers in such semiconductor manufacturing plants can provide savings in terms of both time and money.

Sensitivity analysis helps engineers who work with semiconductor processes understand the processes better by giving them an indication of the magnitude of change in the output metrics when input process parameters are changed. Engineers can use this information to identify the direction in which the system level metrics change when input parameters change and use it to control the process. These analyses can be used in the design stage, to identify parameters which are not sensitive to output metrics. Engineers can fine-tune processes using the other parameters which are much more sensitive and thus save time.

Sensitivity analysis of manufacturing systems has been used extensively to measure sensitivity with respect to drastic changes. This research proposes application of methods to measure sensitivity with respect to small changes, especially in a Heterogeneous Simulation Environment (HSE) using standard simulation tools. Though application of such methods with respect to manufacturing systems has been explored before, the development of such methods with a focus on implementation in a decision support tool is a novelty.

The second area of interest is stochastic optimization, which aids manufacturing systems design. We explore the problem of equipment selection in semiconductor manufacturing systems using discrete stochastic optimization methods.

This problem is of considerable interest to the semiconductor manufacturing industry as the industry spends a large amount of money on equipment. An optimization algorithm which allocates a given amount of money to the right equipment which will give the required throughput with the minimum cycle time possible will be of immense use in the design stage of the manufacturing systems life-cycle.

Machine allocation (or equipment selection) in a manufacturing system is a problem that has been analyzed in great detail using queueing theory and deterministic programming techniques. In this research, a simulation-based approach is taken to the problem and allocation of money to tools is considered.

1.2 Objectives of the Research

The objectives for this work include identifying and implementing techniques for sensitivity analysis and discrete stochastic optimization. The research addresses the following objectives:

- Evaluation of gradient estimation methods used for sensitivity analysis and identification of a suitable method for implementing in a heterogeneous simulation environment (HSE). We compare the finite difference method and simultaneous perturbation gradient estimation methods.

- Implementation of a sensitivity analysis method in a Heterogeneous Simulation Environment (HSE) based decision support tool.
- Development and implementation of a gradient-based discrete stochastic optimization method that provides quality solutions to the manufacturing systems design problem of equipment selection.

1.3 Outline of the Thesis

The thesis is organized as follows.

Chapter 2 gives an overview of semiconductor manufacturing and factory level simulation models of semiconductor wafer fabrication plants. The chapter also gives an overview of the current literature on the application of gradient estimation and simulation optimization methods and the applicability of these to manufacturing systems.

Chapter 3 compares two gradient estimation methods, the finite difference method and the simultaneous perturbation method. The methods described are applied to a stochastic manufacturing system and gradient estimates are obtained. A comparison of the results with analytical models is presented.

Chapter 4 describes the implementation of a gradient estimation method in a decision support tool for process engineers. The methodology behind the selection of the gradient estimation method for sensitivity analysis and the various features of the sensitivity analysis tool are elaborated.

Chapter 5 formulates a manufacturing system design problem and describes a gradient-based discrete stochastic optimization algorithm to obtain quality solutions. A series of experiments are conducted to evaluate the algorithm's performance.

Chapter 6 concludes the thesis, indicates the anticipated impact of methodologies researched in the work and gives avenues for future work.

Chapter 2

Background

2.1 Overview of Semiconductor Manufacturing

The semiconductor industry, when considered from the perspective of manufacturing, is very unique with characteristics that separate it from other manufacturing industries. It is a capital intensive industry with very high barriers to entry. The establishment of a fully equipped semiconductor manufacturing facility may cost more than two and a half billion dollars with possible revenues of more than twice that per year in 1996 [16]. It also has other characteristics like rapid time-to-market and short product development cycles. Some of the factors which make it difficult to apply operations research techniques to semiconductor manufacturing are complex product flows, random yields, diverse equipment characteristics, equipment down-time, production and development in shared facilities and data availability and maintenance [33].

The semiconductor manufacturing process can be subdivided into four basic process steps: wafer fabrication, wafer probe, assembly (or packaging) and final testing. Among these four steps, the most complex and important step is wafer fabrication.

During the wafer fabrication process the circuitry is built on to the chip by adding layers and patterns of metals with interconnects between the layers. There can be hundreds of processes, that the wafer undergoes before exiting the wafer manufacturing facility. The high-level process flow in wafer fabrication, shown in Figure 2.1 consists of cleaning, oxidation deposition, lithography, etching, ion implantation, photoresist strip, inspection and measurement.

In the wafer probe step, individual circuits in the wafer are checked and verified whether they are working properly. The fabrication and probe steps are called the "front-end" steps. In the assembly step the circuit is packaged and placed on PCBs (Printed Circuit Boards).

Finally in the testing phase, each and every integrated circuit is tested so that defect-free products are obtained. Downgrading or binning may also take place where a product that doesn't meet the required specifications but meets the specifications of another product is placed in the other category instead of being scrapped.

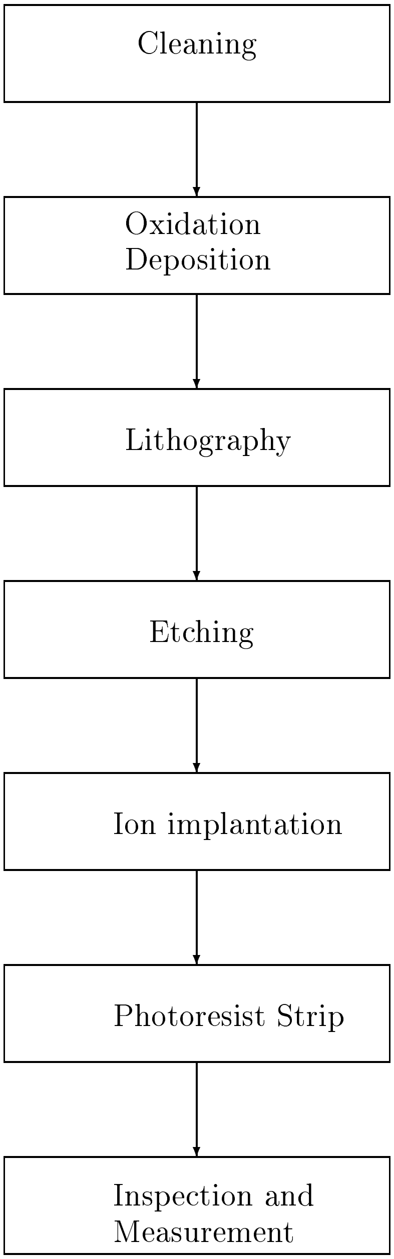


Figure 2.1: Overview of processes in semiconductor manufacturing

2.2 Simulation Modeling of Semiconductor Manufacturing Systems

Simulation models are developed at different stages in semiconductor manufacturing. The three important levels are process modeling, tool modeling and factory modeling.

2.2.1 Process Modeling

Modeling in the product development stage in semiconductor manufacturing can be classified into device modeling and process modeling. Device modeling is a technique used by the product designer to make better chip designs and circuits by studying the behavior of entities from transistors to full-blown computer architectures.

In process modeling, physical processes involved in the manufacture of the wafer are modeled through various techniques. The goal of simulation here will be to develop tools that can predict the outcome of physical processes and help in the development of process flow. Time-to-market will be reduced as the process engineer can reduce development time and cost by reducing physical trial-and-error experiments. Process modeling can be done by empirical modeling where we use solutions of differential equations governing the processes or by using simulation techniques like Monte Carlo simulation. The outputs obtained from process modeling include metrics like metal deposition rate, which form an input to the tool modeling and factory modeling stages in simulation modeling of manufacturing systems. Meyyappan [22] gives an introduction to the fundamentals of process modeling in semiconductor manufacturing.

2.2.2 Tool Modeling

In semiconductor manufacturing, a special type of tool called a cluster tool is used for some processes. According to the Semiconductor Equipment and Materials International (SEMI), a cluster tool [31] is defined as "an integrated, environmentally isolated manufacturing system consisting of process, transport, and cassettes of modules mechanically linked together". Typically, a cluster tool can perform different processes incorporating many process variabilities and intricacies of wafer moves between different chambers in the tool.

Cluster tool modeling addresses the scheduling of different processes in the cluster tool which also depends on the robot handler moves in the cluster tool. Push and pull scheduling rules [23] form the basis for cluster tool modeling while sophisticated scheduling algorithms are also being researched. The reader is referred to Wood [34], Srinivasan [31], and Nguyen [23] for cluster tool performance modeling and scheduling.

2.2.3 Factory Modeling

Some of the factors which lead to the development of simulation models are intractability of detailed analytical models of semiconductor processes, uncertainties inherent in manufacturing systems, improvement in computational power available to do simulation and the development of easy-to-use simulation tools like Arena, Promodel, AutoSched and Factory Explorer.

While some models are used in the planning and design stage, some models are also used in day-to-day operations, but these models address only system level changes. Hence a simulation model is useful in two situations: systems modeling and design and system control.

Factory modeling comes into play during different phases in the life-cycle of semiconductor factories. The different phases in the life-cycle of a semiconductor factory include design, production ramp, early high volume production (when demand is greater than supply) and late commodity production (when demand is falling).

In this work, the focus is on combining advanced operations research techniques with simulation models to evaluate the performance of semiconductor manufacturing facilities.

2.2.4 Factory Explorer

Factory Explorer®[1] [8] is the simulation tool which has been extensively used in this work. It is a discrete event simulation software which was developed specifically for modeling semiconductor manufacturing systems. Factory Explorer has an analytical engine, which predicts metrics like bottleneck resources, tool utilization and cost data, and a simulation engine, which estimates cycle time, work in process and other metrics. The input and output of data in Factory Explorer is through Microsoft Excel®[2] spreadsheets.

2.3 Equipment Selection in Manufacturing Systems Design

Equipment selection (or machine allocation) problems form a separate class of problems in the domain of manufacturing systems design. By equipment selection, we mean the selection of tools for workstations in a manufacturing system given a choice of tool types. Allocation and selection of tools in manufacturing systems is a widespread problem in manufacturing plants as there are systems like flexible manufacturing systems (FMS) and cellular manufacturing systems

[1] Registered trademark of Wright, Williams and Kelley Inc.

[2] Registered trademark of Miscrosoft Inc.

which are smaller manufacturing systems where tool selection has to be done and these systems themselves are components of the factory.

These problems have been addressed using analytical models, queueing theory and deterministic programming techniques like integer programming. The machine allocations were done with specific objectives like minimizing WIP, maximizing throughput, minimizing cost. Equipment selection problems can be generally classified using system and problem characteristics like having a fixed number of machines to allocate, using minimum cost allocation, obtaining an optimal output metric or obtaining a fixed output metric.

Frenk, Labbé, Van Vliet and Zhang [10] proposed algorithms for machine allocation problems where the WIP (work-in-process) was required to be less than a certain level. They developed system models using queueing networks analysis and proposed algorithms that will allocate machines for these systems. Shantikumar and Yao [28] formulated the server allocation problem as a deterministic nonlinear integer program and modeled the problem using a closed queueing network. They also developed a greedy heuristic to provide an approximate solution to the problem.

2.4 Sensitivity Analysis

One of the important applications of large-scale simulation models is sensitivity analysis, which can refer to either large-scale changes in the system or small changes made to some of the parameters governing the system. Some examples of drastic changes are changes in scheduling rules or changes to the number of tools in a workstation. These types of analyses can be performed using methods like design of experiments, which uses regression analysis to build a meta model of the system. The reader is referred to Kleijnen [18] for one such implementation. Some examples of small changes will be perturbations of processing times of workstations or changes in setup times at individual workstations. Some techniques that can be used to perform this type of analysis are gradient-based methods like finite differences or perturbation analysis. An overview of gradient estimation is presented next.

2.5 Gradient Estimation

2.5.1 Overview

Gradient estimation is an important technique that can be utilized to estimate the impact of change in input parameters on output metrics in stochastic processes. If the response of the output metrics with respect to the input parameters is continuous in nature, then the gradient of the output metric is obtained as a partial derivative of the response function. Gradient estimation for applications like optimization and sensitivity analysis can be done through a number of methods. For a more detailed overview of gradient estimation and the methods involved, the reader is referred to Banks [5], Fu [9] and L'Ecuyer [20].

Some of the methods for gradient estimation are finite difference method, perturbation analysis, likelihood ratio method, frequency domain experimentation and simultaneous perturbation method. While some methods like the perturbation analysis method require knowledge of the system being simulated which requires obtaining output or change in the input when the simulation is in progress, other methods like the finite difference methods take a black-box type approach to system being simulated for estimating the gradient.

In this overview, some of the methods that require knowledge of simulation internals are presented. The two methods to be compared are presented in Chapter 3.

2.5.2 Perturbation Analysis

Perturbation Analysis [14] includes methods such as Infinitesimal Perturbation Analysis (IPA) and Smoothed Perturbation Analysis (SPA). IPA [32] reformulates the problem of estimating the gradient with respect to the input parameters as the problem of estimating the gradient of an expected value involving a random variable whose distribution does not depend on the input vector, θ.

Perturbation analysis makes use of the concept of sample path analysis to estimate the gradient. The underlying assumption of IPA is that small changes in the metric being measured do not cause changes in the event schedule, unlike drastic changes like changes in dispatching rule. IPA estimates the gradient by accumulating the infinitesimal changes over the simulation. For example, an estimate of the gradient of system time with respect to the mean processing time in a G/G/1 queue,is the following:

$$\left(\frac{dT}{d\theta}\right)_{IPA} = \frac{1}{N}\sum_{m=1}^{M}\sum_{i=1}^{n_m}\sum_{j=1}^{i}\frac{dX_{(j,m)}}{d\theta} \tag{2.1}$$

With exponential service times,

$$\frac{dX}{d\theta} = \frac{X}{\theta} \tag{2.2}$$

for infinitesimal changes in X,

where

$X_{(j,m)}$ = Processing time of the j^{th} customer in the m^{th} busy period.

T = System time.

θ = Mean processing time.

N = Total number of customers served.

i = Counter for summation over customers.

n_m = Number of customers during the m^{th} busy period.

M = Number of busy periods.

2.5.3 Frequency Domain Experimentation

The frequency domain experimentation involves oscillating the value of the input parameter in a sinusoidal fashion during a single run, which will give an output function, a superposition function of the different inputs. This output function can be used for gradient estimation. This method is described in detail by Jacobson [15].

2.6 Stochastic Optimization

Stochastic optimization is implemented when the process that has to be optimized is stochastic in nature. The reader is referred to Banks [5] and Fu [9] for a review of simulation optimization techniques. Simulation-based stochastic optimization techniques are still complex, in spite of advances in computing. They have to be chosen carefully and adapted, according to the problem on hand.

Stochastic optimization methods can be classified based on the type of decision variables and the way the optimization process works. Stochastic optimization techniques can be classified into iterative and non-iterative. Based on the type of decision variables, they are classified into continuous and discrete stochastic optimization techniques. A description of the iterative process and non-iterative process is provided followed by a classification of some of the techniques based on the type of the decision variable.

2.7 Iterative Procedure

In the iterative stochastic optimization process, the optimizer uses the simulator iteratively to obtain the value of the objective, $f(t)$, which is being optimized, in order to evaluate the solution space of decision variables. The process is illustrated in Figure 2.2,where t is the initial feasible decision variable vector and t' is the new decision variable suggested by the optimizer, which is evaluated using the function measurements $f(t)$. Gradient-based stochastic approximation methods are examples of iterative procedures.

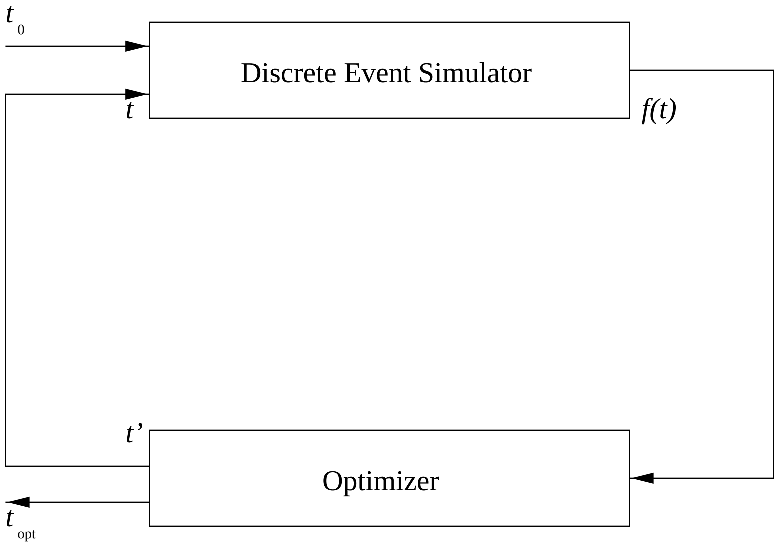


Figure 2.2: Iterative stochastic optimization process

2.8 Non-iterative Procedure

In a non-iterative stochastic optimization process (Figure 2.3), we make all the required function evaluations using simulation or other means before the optimization method is used. Using these function evaluations, we optimize the process. Sample path optimization is an example of a non-iterative procedure.

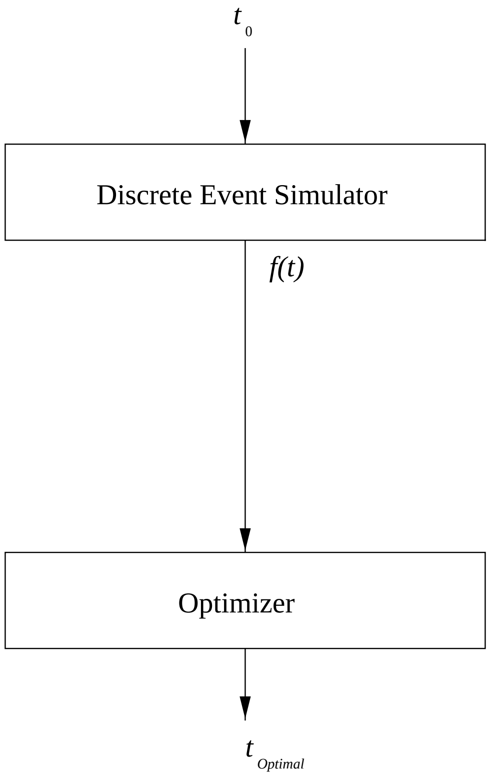


Figure 2.3: Non-iterative stochastic optimization process

2.9 Continuous Optimization Techniques

Continuous optimization problems are problems whose decision variables are continuous in nature. Two classes of methods, which are applied to continuous optimization problems, are stochastic approximation and sample path optimization.

2.9.1 Stochastic Approximation

Stochastic approximation is usually applied in conjunction with a gradient estimation method to choose the next set of values for the decision variables in an iterative process which will finally lead to an optimal solution.

Finite Difference Stochastic Approximation (FDSA), which uses finite differences to make gradient estimates, was introduced by Kiefer and Wolfowitz [17] and has been applied extensively for continuous optimization.

The reader is referred to Spall [29] for an overview of the implementation of the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm which uses simultaneous perturbation for gradient estimation. The reader is referred to Chapter 3 for a detailed description of the finite differences and the simultaneous perturbation gradient estimation methods.

2.9.2 Sample Path Optimization

In this technique, the original stochastic optimization problem is converted into an approximate deterministic optimization problem. This step is followed by application of regular deterministic programming methods to the deterministic problem in order to obtain an optimal solution. The reader is referred to Robinson [26] for an analysis of sample-path optimization and Rubinstein [27] for an overview on the application of sample path optimization using the LR method.

2.10 Discrete Stochastic Optimization Techniques

Discrete optimization has been predominantly carried out using random search techniques through a combinatorial solution space of discrete decision variables. Gradient-based methods have also been applied, where the decision to move in the solution space is based on the gradients of the objective function. This research explores the application of gradient-based techniques rather than random-search techniques since a gradient-based technique will be more suitable for the approach taken to the problem on hand. For discrete stochastic optimization, researchers have previously proposed random search techniques, conventional techniques like branch-and-bound [25] and variants of continuous optimization techniques.

2.10.1 Random Search Techniques

Random search techniques are discrete optimization techniques that move from one feasible point to another in search of the optimal solution. There are a variety of random search techniques that vary in the choice of the neighborhood structure, the decision strategy when moving from the current alternative θ_n to the next alternative θ_{n+1}, and the method for obtaining estimates of the optimal solution. Andradóttir proposed two algorithms, one of which converges to a local solution [3] while the other algorithm converges to a global solution [4]. Yan and Mukai [35] proposed the stochastic ruler algorithm where estimates of the objective function are compared with a uniform random variable U called the stochastic ruler. Andradóttir and Alrefaei [2] developed a variant of the stochastic ruler algorithm. They also developed a simulated annealing algorithm for the discrete stochastic optimization problem [1]. Genetic algorithms and tabu search are some other techniques which fall under this category.

2.10.2 Gradient-based Discrete Optimization Techniques

Gradient-based optimization techniques can be applied to discrete optimization problems. SPSA has been recently applied to discrete stochastic optimization problems by Gerencsér, Hill and Vágó [11]. They proposed a fixed gain version of the SPSA method and applied it to a class of discrete resource allocation problems formulated by Cassandras, Dai and Panayiotou [6].

2.11 Summary

An overview of simulation-based operations research techniques has been presented in the context of semiconductor manufacturing. Simulation modeling is a very integral part of the semiconductor manufacturing process at various stages. Some of the techniques discussed can be applied to semiconductor manufacturing to aid in decision making, process control and manufacturing systems design.

Chapter 3

Comparing Gradient Estimation Methods

3.1 Introduction

In the previous chapter, introduction to some of the gradient estimation methods was provided. This chapter presents the finite difference (FD) method and the simultaneous perturbation (SP) method. These two methods, unlike methods presented in the previous chapter, do not require knowledge of the underlying simulation and hence can be utilized with any discrete event simulation model.

Let us consider a stochastic process that has a certain number of input parameters and output metrics, which help us determine the performance of the process. The output metrics are obtained either through experiments, simulation or some other process as depicted in Figure 3.1.

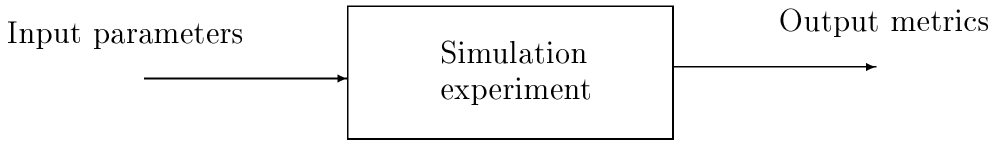


Figure 3.1: Simulation box

The sensitivity of the output metrics to the input processes is very helpful in determining the impact of the input parameters on the output processes. The output metric can be expressed as a function of the input parameters:

$$f = f(\theta_1, \theta_2, \ldots, \theta_n), \tag{3.1}$$

where f is the output metric written as a function of θ_i, $i = 1, 2, \ldots, n$, the input parameters.

The aim is to estimate the gradient of the output metric, g, where

$$g_i(\theta) = \frac{\partial f(\theta)}{\partial \theta_i} \tag{3.2}$$

gives the partial derivative of f with respect to the ith input parameter.

3.2 Finite Difference Method

In a one-dimensional case, the derivative of a function f is given by

$$g(\theta) = \lim_{c \to 0} \frac{f(\theta + c) - f(\theta - c)}{2c}. \tag{3.3}$$

When the step size c, is small, we can reasonably estimate the gradient by estimating the function f at $\theta + c$ and $\theta - c$.

The FD method of estimating the gradient is given by

$$\hat{g}_i(\theta) = \frac{\hat{f}(\theta + c_i e_i) - \hat{f}(\theta - c_i e_i)}{2c_i}, \tag{3.4}$$

where

c_i = step size,

e_i = unit vector in the i^{th} direction.

Thus we can estimate the gradient by conducting one simulation with input parameter $\theta + c_i e_i$ and obtain an estimate of $f(\theta + c_i e_i)$ and conduct another simulation at $\theta - c_i e_i$ and obtain an estimate of $f(\theta - c_i e_i)$. Equation 3.4 gives the gradient with respect to one input parameter. The gradient can be estimated for $i = 1, 2, \ldots, p$ parameters by $2p$ simulations with step size c_i and unit vector e_i for $i = 1, 2, \ldots, p$. One of the problems with the finite difference estimator is that when the step size is small, the variance of the estimators becomes large and when the step size increases, the bias of the estimate increases. So choice of simulation parameters like number of replications and choice of the estimator parameters like step size should be done carefully.

3.3 Simultaneous Perturbation Method

The SP gradient estimation method uses just two simulations for estimating all the gradients. The SP gradient estimator for a process with n input parameters and one output metric, f is given as follows:

$$\hat{g}_i(\theta) = \frac{\hat{f}(\theta + C\Delta) - \hat{f}(\theta - C\Delta)}{c_i \Delta_i}, \tag{3.5}$$

where

Δ = n-dimensional random perturbation i.i.d vector,

C = Diagonal matrix with step sizes for input parameters along the diagonal.

The reasoning behind the representation of the step size as a diagonal matrix is explained with the help of Equation 3.6, where c_i is the i^{th} diagonal element in C:

$$\hat{f}(\theta + C\Delta) = \hat{f}\left(\begin{bmatrix} \theta_1 \\ \theta_2 \\ .. \\ \theta_n \end{bmatrix} + \begin{bmatrix} c_1 & 0 & 0 & .. & 0 \\ 0 & c_2 & 0 & .. & 0 \\ .. & .. & .. & .. & .. \\ 0 & 0 & 0 & .. & c_n \end{bmatrix} \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ .. \\ \Delta_n \end{bmatrix}\right) \tag{3.6}$$

Each element of Δ is independently generated from a probability distribution with mean zero and finite second inverse moment, precluding a uniform or normal distribution. The rationale behind proper choice of Δ is explained in detail by Spall [30]. The method differs from the FD method in that all the input parameters are simultaneously perturbed during a single simulation. In the two simulation runs that are needed to estimate the gradient using the SP method, the perturbations in parameter values will be equal and opposite in sign. Hence, only the denominator of equation 3.5 differs for each component will be varying as Δ varies while the numerator will remain the same. Also the step size, c_i may remain the same for different input parameters or scaled for different input parameters, if the input parameters vary greatly in magnitude.

3.4 Problem Statement

We consider the problem of estimating sensitivity of the steady-state average cycle time (CT) to the processing times (PT) of each operation in the manufacturing system. The manufacturing system is a flow shop with no reentrant flow. The manufacturing system produces just one product. This problem is important, because the impact of processing times on total cycle time will give the people who work with the system information on the importance of the process parameters. The manufacturing system has seven workstations. The seven workstations are Coater, Stepper, Developer, Exposer, Printer, Reader and Writer. Table 3.1 gives the number of tools at each workstation and the mean processing time of that operation at that workstation.

The product, which is being manufactured is a wafer, which enters the factory in lots of 1 unit each. The lots enter with a mean interarrival time of 4 hours. The interarrival times and the processing times are exponentially distributed. This

Tool Group	Number of tools	Processing Time (in Hrs)
Coater	2	5
Stepper	1	1
Developer	2	5
Exposer	2	6
Printer	1	3
Reader	1	2
Writer	2	7

Table 3.1: Tool groups in the model and their parameters

aids in building simple analytical models to evaluate the system. The input model is depicted in Figure 3.2. We will use the Factory Explorer simulation tool [8] to simulate the system and obtain estimates of the average total cycle time of each tool.

3.5 Gradient Estimate using Finite Difference Method

Gradient measurement using FD method can be done through several submethods like the forward difference, backward difference and central difference methods. We will use the central difference method because the gradient estimate from the central difference method will usually have less bias than the forward or backward difference.

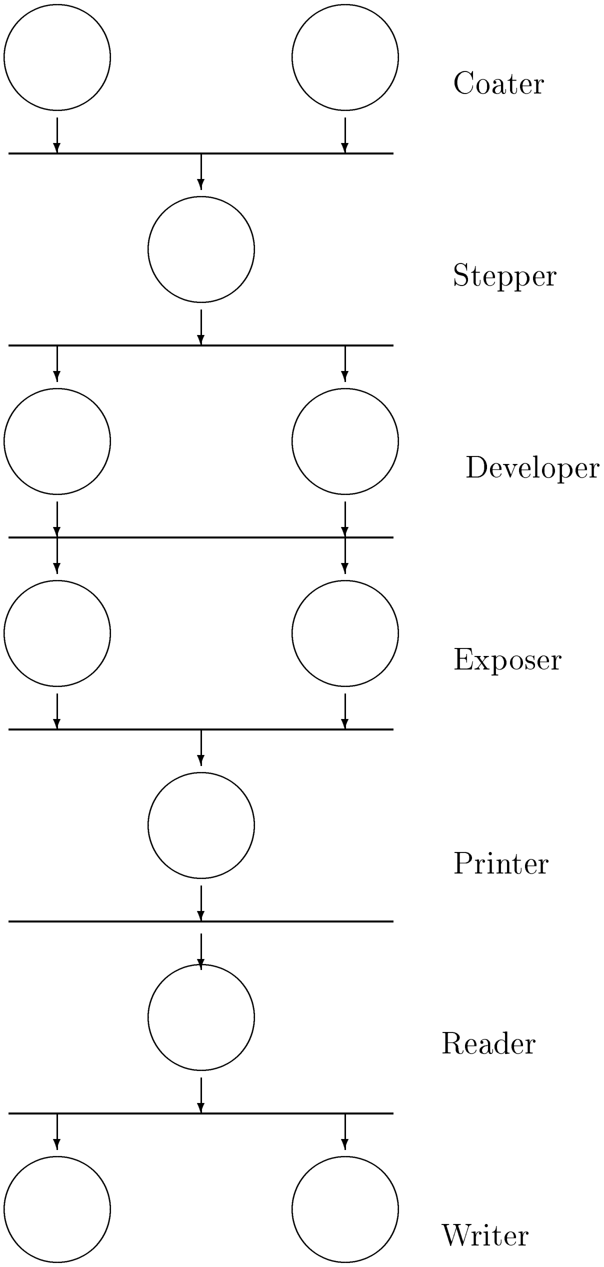


Figure 3.2: Input model - manufacturing system

The FD estimator for the above function is

$$(\hat{g}_i(\theta))_N = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{\hat{f}_j(\theta + c_i e_i) - \hat{f}_j(\theta - c_i e_i)}{2c_i}\right), \tag{3.7}$$

where

$\hat{g}_i$ = Estimate of the i^{th} component of the gradient vector,

$\hat{f}_j$ = j^{th} estimate of the function, which is obtained from simulation,

c_i = Step size for the i^{th} parameter,

θ = Vector of baseline input parameters,

N = Number of replications,

e_i = Unit vector in direction i.

The simulation tool used for conducting simulations considers the time duration for which the system is simulated rather than the number of customers, so we simulate the system for 87600 hours (10 years). To obtain an estimate of precision, we perform $N = 20$ replications. The cycle time and gradient estimates, along with standard error, for one parameter over 20 replications are given in Table 3.2. Since the model has seven input parameters we need a total of 280 simulation runs.

An important parameter in the FD method is the step size. In the FD method, a large step size yields estimates with high bias, but a small step size yields estimates with high variances. For this model, we consider a step size $c_i = \theta_i/100$. This step size is relatively small, but using a higher number of replications can reduce the variance.

Replication	Average cycle time-lower	Average cycle time-higher	Gradient
1	81.164	81.539	3.75
2	77.400	77.779	3.79
3	78.186	78.566	3.80
4	83.612	83.938	3.26
5	74.415	74.898	4.83
6	77.952	78.677	7.25
7	77.179	77.394	2.15
8	78.973	80.158	11.85
9	79.944	80.447	5.03
10	76.042	76.326	2.84
11	78.114	79.271	11.57
12	77.876	77.857	-0.19
13	73.183	73.015	-1.68
14	79.941	79.987	0.46
15	78.657	79.044	3.87
16	75.986	77.418	14.32
17	78.534	78.605	0.71
18	76.034	76.623	5.89
19	75.508	75.391	-1.17
20	76.261	76.851	5.90
		Average	4.411
		Standard error	0.955
Half width of confidence interval			2.415
	Confidence interval (1.986,6.837)		

Table 3.2: Table showing cycle time and gradient estimates estimated by finite difference method for the process coater

Based on the chosen values that define the logistics of the simulation and input parameters, the gradients for the average total cycle time with respect to the mean processing parameters are estimated and 99% confidence intervals are constructed as $(\bar{X} - h, \bar{X} + h)$, where

$$S = \sqrt{\frac{\sum_{i=1}^{N} X_i^2 - N\bar{X}^2}{N - 1}} \tag{3.8}$$

where

X_i = Gradient estimate at replication i.

$\bar{X}$ = Mean gradient estimate over N replications. $(\bar{X} = (\sum_{i=1}^{N} X/N))$

$$h = t_{N-1,1-\alpha/2}\frac{S}{\sqrt{N}} \tag{3.9}$$

where

$t_{N,\alpha}$ = The critical value from a t-distribution with n degrees of freedom,

$\alpha = 0.01$, if 99% is the confidence needed in the estimate.

The confidence interval is given by $(\bar{X} - h, \bar{X} + h)$. Table 3.3 gives the summary data for the FD method including the cycle time and gradient estimates. Figure 3.3 gives a graphical representation of the gradient results compared with the gradients from the analytical method.

An important conclusion that can be obtained from the finite difference method is that changes in average cycle times of the manufacturing system when mean processing times of a particular machine are varied are almost equal to the change in the average cycle time of that particular machine. This is facilitated

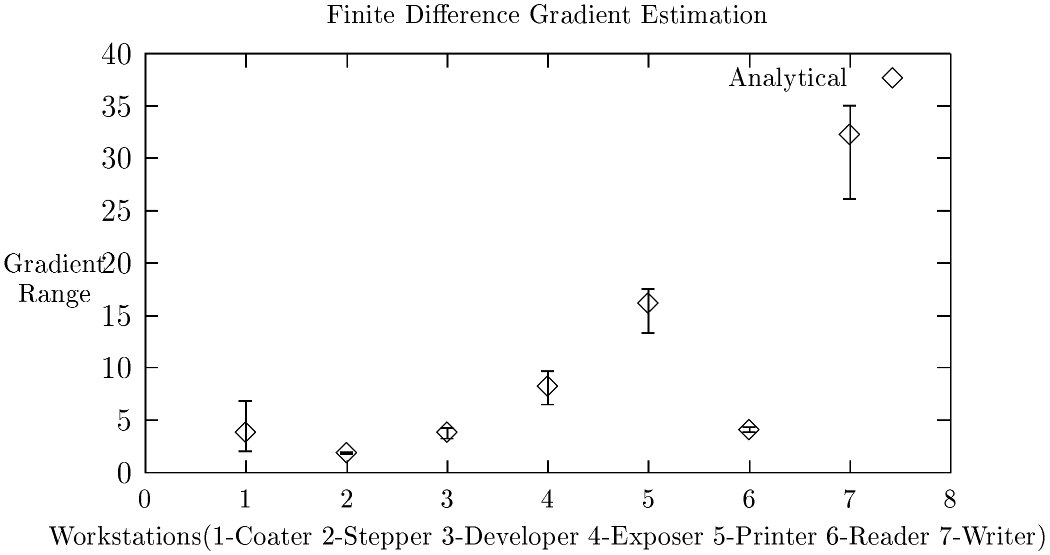


Figure 3.3: Plot comparing finite difference method and analytical models

by the finite difference method where we estimate the average cycle time on a per-parameter basis. This is reflected in the data in Table 3.4.

3.6 Gradient Estimate using Simultaneous Perturbation Method

The SP gradient is estimated for the ith parameter as follows. g is the n-dimensional vector of gradients

$$(\hat{g}_i(\theta))_N = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{\hat{f}_j(\theta + c\Delta_j) - \hat{f}_j(\theta - c\Delta_j)}{2c_i\Delta_{ji}}\right), \tag{3.10}$$

where

$\hat{g}_i$ = Estimate of the gradient for the i^{th} input parameter,

N = Number of replications,

Tool	Coater	Stepper	Developer	Exposer	Printer	Reader	Writer
Number of tools	2	1	2	2	1	1	2
Mean processing time-baseline	5	1	5	6	3	2	7
Processing time-lower	4.95	0.99	4.95	5.94	2.97	1.98	6.93
Average cycle time-lower	77.748	77.990	77.795	77.538	77.558	77.926	76.015
Mean processing time-upper	5.05	1.01	5.05	6.06	3.03	2.02	7.07
Average cycle time - upper	78.189	78.026	78.171	78.504	78.482	78.091	80.300
Gradient	4.411	1.840	3.762	8.052	15.398	4.109	30.602
Confidence Interval	(1.986, 6.837)	(1.753, 1.927)	(3.262, 4.262)	(6.447, 9.657)	(13.312, 17.483)	(3.852, 4.365)	(26.121, 35.083)

Table 3.3: Summary data for finite difference method

Tool	Change in average cycle time at tool	Change in total average cycle time	Difference
Coater	0.381	0.441	0.060
Stepper	0.036	0.037	0.001
Developer	0.379	0.376	-0.002
Exposer	0.913	0.966	0.053
Printer	0.982	0.924	-0.058
Reader	0.161	0.164	0.003
Writer	4.285	4.284	0.000

Table 3.4: Table comparing the change in CT at a tool and the change in total CT when the PT for that tool is varied between the upper and lower levels

$\theta =$ Baseline mean processing time vector of size n,

$\Delta_j =$ a n-dimensional random perturbation i.i.d vector, which keeps changing for every j^{th} replication,

$c_i =$ step size for the i^{th} input parameter.

The implementation of the gradient estimator has been studied in depth in [29] as part of a study on stochastic optimization using simultaneous perturbation stochastic approximation. The gradient for all input parameters can be calculated with only two simulations with one replication of the estimation process.

The Δ vector considered here is obtained from a Bernoulli distribution. It consists of n i.i.d symmetric Bernoulli random variables X_i. $P\{X_i = \pm 1\} = 0.5$. The step size considered here is $c_i = \theta/100$. Setting c_i as a function of θ_i takes care of the differences in the magnitudes of the processing times. Table 3.5 gives

Tools	Number of Tools	Mean processing time (Hours)			Gradient	Confidence Interval
		Lower		Upper		
Coater	2	4.95	5	5.05	5.574	(-4.780,15.928)
Stepper	1	0.99	1	1.01	-22.743	(-74.512,29.027)
Developer	2	4.95	5	5.05	3.809	(-6.545,14.162)
Exposer	2	5.94	6	6.06	11.631	(3.003,20.259)
Printer	1	2.97	3	3.03	13.664	(-3.592,30.921)
Reader	1	1.98	2	2.02	11.146	(-14.738,37.031)
Writer	2	6.93	7	7.07	34.979	(27.583,42.374)

Table 3.5: Summary data for simultaneous perturbation method

the summary data for the SP method including the cycle time and gradient estimates.

The SP gradient estimation is done for $N = 140$ replications. This facilitates comparison between SP and FD. While the SP method take two simulations to estimate the gradient for seven parameters, the FD method needs 14 simulations. Hence when we have $N = 20$ replications for the finite difference method, we can have $N = 140$ replications for the SP method.

Figure 3.4 gives a graphical representation of the gradient results compared with the gradients from the analytical method, to be described in the next section.

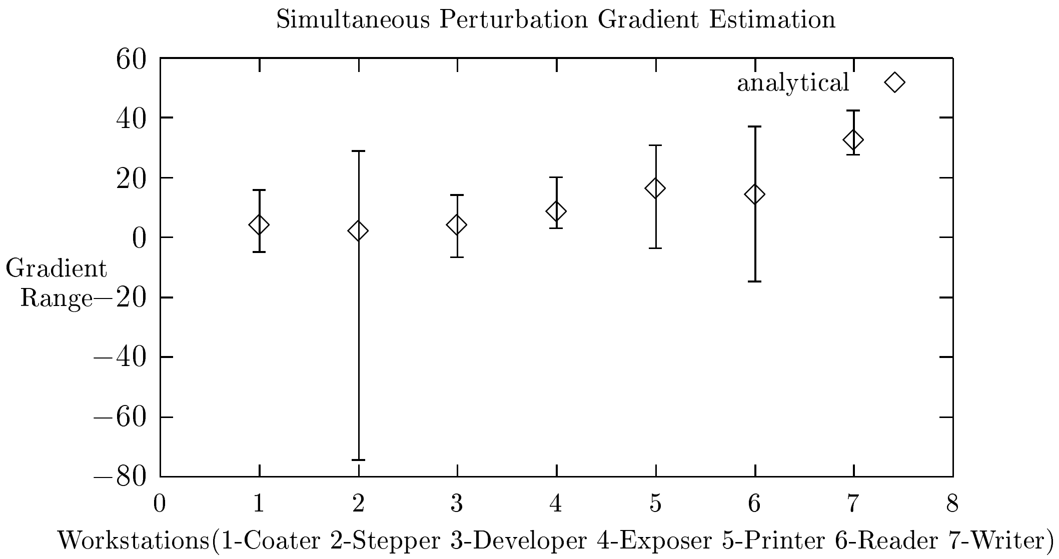


Figure 3.4: Plot comparing simultaneous perturbation method and analytical models

3.7 Analytical Verification

The gradients obtained by the FD and SP methods were compared to partial derivatives, which are calculated exactly. In the model considered, we have workstations with one or two tools each. Each station acts as an M/M/1[1] or M/M/2 queuing system. The cycle times at each tool can be calculated using exact models for M/M/1 and M/M/2 systems, since both interarrival times and processing times are exponential. The utilization, cycle time and gradient formulae for M/M/1 and M/M/2 queues are given below. For a derivation of

[1] M/D/1,M/U/1,M/D/2 are also analytically tractable.

the formulae, the reader can refer to [7] and [24].

M/M/1:

$$\begin{aligned} u_i &= r_a t_i, && (3.11) \\ CT_i &= \frac{t_i}{(1-u_i)}, && (3.12) \\ \frac{\partial(CT_i)}{\partial(t_i)} &= \frac{1}{(1-u_i)^2}, && (3.13) \end{aligned}$$

where

u = Utilization of tool at workstation i,

r_a = Arrival rate = (1/Mean inter-arrival time),

t_i = Mean processing time of operation at workstation i,

M/M/2:

$$\begin{aligned} u_i &= \frac{r_a t_i}{2}, && (3.14) \\ CT_i &= \frac{t_i}{(1-u_i^2)}, && (3.15) \\ \frac{\partial(CT_i)}{\partial(t_i)} &= \frac{(1+u_i^2)}{(1-u_i^2)^2}, && (3.16) \end{aligned}$$

where

u = Utilization of the tool,

r_a = Arrival rate = (1/Mean inter-arrival time),

t_i = Mean processing time of operation at workstation i,

Table 3.6 gives the analytical cycle times and gradients.

Tool	Arrival Rate (Jobs/Hour)	Number of tools	Mean Processing Time(Hours)	Utilization	Average Cycle time (Hours)	Partial Derivative
Coater	0.25	2	5	62.5	8.205	3.745
Stepper	0.25	1	1	25.0	1.333	1.778
Developer	0.25	2	5	62.5	8.205	3.745
Exposer	0.25	2	6	75.0	13.714	8.163
Printer	0.25	1	3	75.0	12.000	16.000
Reader	0.25	1	2	50.0	4.000	4.000
Writer	0.25	2	7	87.5	29.867	32.142

Table 3.6: Summary data for analytical models

3.8 Discussion

The FD method provided reasonably good estimates for the gradient of average total cycle time with respect to the mean processing times, while the SP method did not perform as well as the FD method. It gave poor confidence limits for the gradients, though the mean gradient was quite accurate for some of the parameters. This could be due to the fact that the estimate for one value depends on the way in which one variable affects the others during cycle time estimation, which results in the high noise levels in the measurements of the gradients. When we compare the gradient estimates of both the methods against the exact method we can see that, the FD method has performed significantly better than the SP method.

3.9 Summary

Two gradient estimation techniques, the finite differences and the simultaneous perturbation method, are described. The methods described are used to analyze a stochastic manufacturing system, and gradients are estimated. The results are compared to the gradients calculated from analytical queueing system models.

These gradient methods are of significant use in complex manufacturing systems like semiconductor manufacturing systems where we have a large number of input parameters that affect the cycle time. Gradient estimation methods will help us estimate the impact of these input parameters on average cycle time and identify the parameters that have the maximum impact on average cycle time.

Chapter 4

Implementing Gradient Estimation

4.1 Introduction

In the semiconductor industry, the ratio of number of process engineers to number of industrial engineers is around 10:1. The process engineer has to go to the industrial engineer to consult on decision making regarding changes in the process parameters with which he is working. This may turn out to be a time-consuming process leading to loss of productivity. If the process engineer is provided with a decision support tool which will enable him deal with such situations, he will be able to make decisions faster regarding operational process parameters which will eventually result in higher production.

Implementation of sensitivity analysis as part of a software tool for semiconductor manufacturing systems allows process engineers to gauge the impact of the process parameters they are working with on the overall performance metrics of the semiconductor fabrication facility. Such a software tool will aid in intelligent decision making.

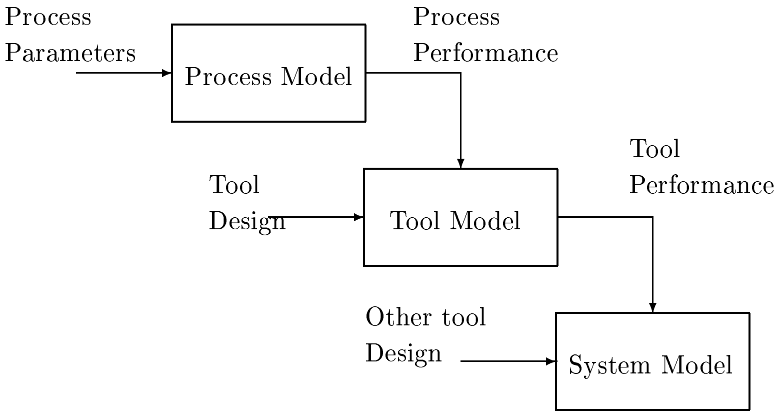


Figure 4.1: Hierarchy of simulation models

4.2 Motivation for a Sensitivity Analysis Tool

Sensitivity analysis, as described in Chapter 2, for stochastic systems like semiconductor manufacturing systems can either mean impact of high level changes like changes in scheduling rules or can mean perturbations in process parameters. Design of experiments and other methodologies can be utilized to estimate the effect of large changes in process metrics. These techniques do not perform well for small changes in process metrics. Here the aim is to look at perturbations in process parameters and their effect on system level output metrics like cycle time. The added motivation for gradient estimation methods, integral to sensitivity analysis, is their use in simulation optimization, which will ultimately provide more value.

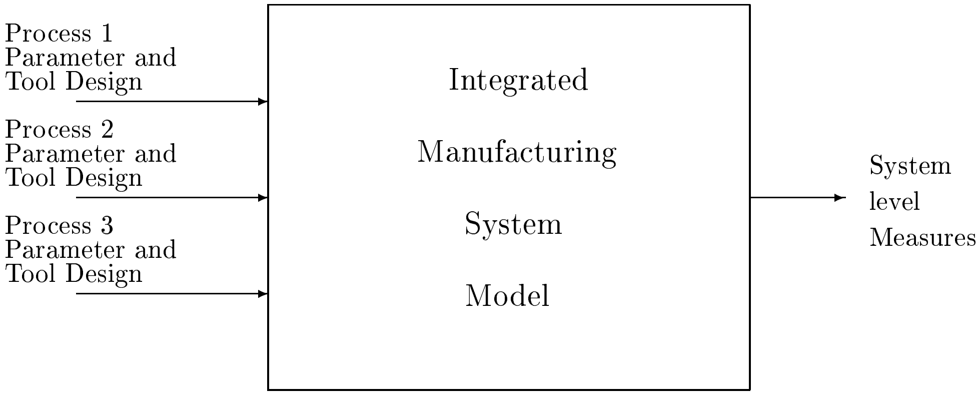


Figure 4.2: An integrated model

4.3 Architecture of Factory Administrator

When we consider modeling today in the semiconductor industry, we normally have three layers of different systems (Figure 4.1) which are modeled independently. The process models look at the material processes. The raw process times are calculated from these process models. The next layer is the cluster tool layer. Modeling of cluster tools is done where each cluster tool can perform one or more processes. We obtain the lot process times from these models. The final model will be the system level model where we include all the individual tool models to form a system level model, which will provide us with system level metrics.

An integrated model (Figure 4.2) called the Factory Administrator has been developed which will directly output the system level metrics when changes are made either to the tool design parameters or to the process parameters.

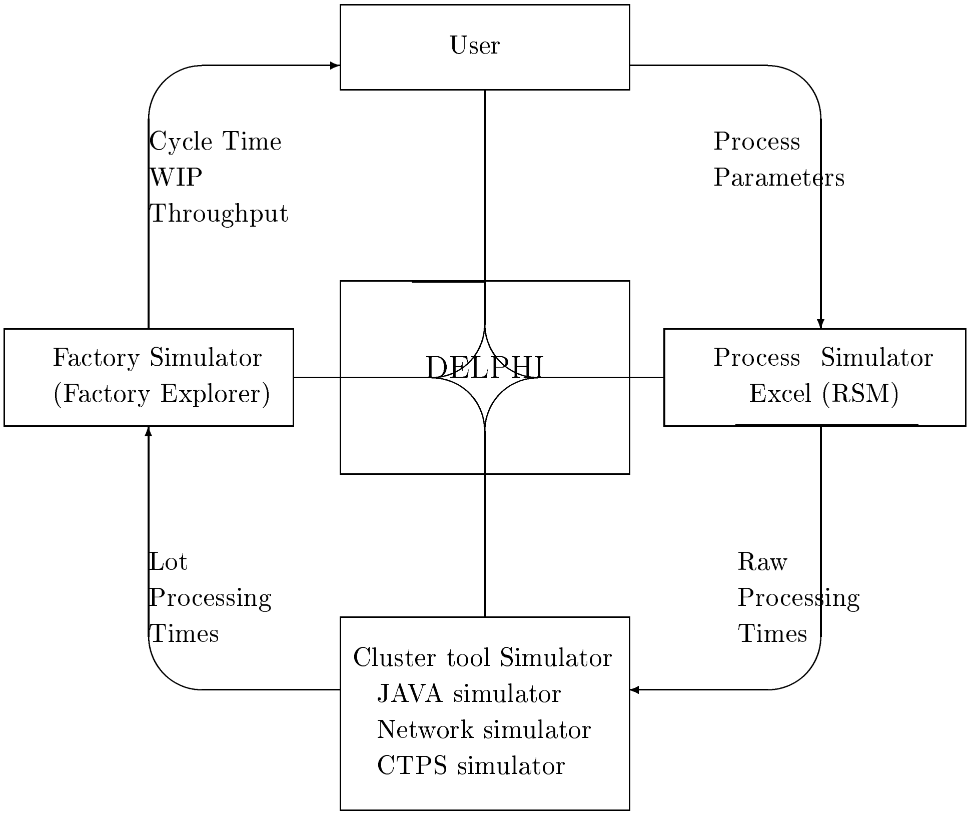


Figure 4.3: Architecture of the factory administrator

Integration gives us the power of viewing system level performance while having control over all input parameters.

An integrated model is very essential for sensitivity analysis because sensitivity analysis involving process models is done by running multiple simulations using the integrated model with process, tool and factory parameters.

4.4 Description of Factory Administrator

The Factory Administrator [13] has a front-end GUI (Graphical User interface) which has been developed using Delphi®[1]. The architecture is presented in Figure 4.3. The user does all transactions on the model using this front end shown in Figure 4.4. The process models are Response Surface Models (RSM) which have been embedded in Excel. The process parameters for each process are displayed in a worksheet in Excel, which is read by Delphi and displayed to the user, when he wants to use the process parameters for any particular process. Any change to be performed on the process parameters is done in the Delphi front-end which then updates the Excel spreadsheets. The raw processing times are calculated using the updated process parameters according to Response Surface Models (RSM) and updated in the spreadsheet.

The lot processing time for each tool is estimated using the cluster tool simulators. There are five types of cluster tool simulators available (Push, Pull, Optimal, Cyclic and Fixed Sequence). The JAVA®[2] cluster tool simulator has been integrated in the Factory Administrator. The lot processing times can thus be obtained from the Delphi front end itself.

The lot processing times are input into the factory simulation model which is a workbook in Excel. The factory model is developed with Factory Explorer,

[1] Registered trademark of Inprise Inc.

[2] Registered trademark of Sun Microsystems Inc.

which communicates with the user using Excel and simulates using a back-end discrete event simulation engine. Factory Explorer gives its output as a text file, which is read by the Delphi front-end. The front-end GUI then picks the required output metrics and displays them to the user.

Thus the user can make changes to the process parameters and then run the factory simulator directly to see the system level performance measure changes.

4.5 Selection of the Gradient Estimation Method

Several methods were considered for gradient estimation including finite difference method, simultaneous perturbation method, perturbation analysis and frequency domain experimentation. Some of the constraints that are applicable to the integration of gradient estimation methods in the Factory Administrator include

- **Computing efficiency** - The method used should not be computationally very intensive. Optimal use of computing power should be made so that accurate results are obtained within less time.

- **Knowledge of simulation** - The simulation tool which is being used does not provide data when the simulation is being run. Real time data cannot be obtained from the Factory Explorer simulation tool. Hence the methodology should have a black-box type approach towards the simulation tool.

- **Consistency and unbiasedness of the estimate** - The estimates of the output which we obtain from the machine should be unbiased and should provide reliable estimates of the output metric. This will be aided by the ability of the tool to use common random numbers which will reduce variability.

The advantages and disadvantages of some of the methods used are illustrated in Table 4.1. After comparing the advantages and disadvantages of the various methods, finite differences and the simultaneous perturbation were compared for performance (as Chapter 3 describes). Since finite differences gave tighter confidence intervals for the gradient obtained, the finite differences method was chosen for implementation in the Factory Administrator.

4.6 Implementation

The finite difference (FD) method depends on a number of parameters like the process parameter on which sensitivity analysis is being performed, the time duration for which the simulation is being run, the number of replications for which the simulation is run, the step size and other factors.

Gradient measurement using FD method can be done through several submethods like the forward difference, backward difference and central difference methods. We will use the central difference method, because the gradient esti-

Method	Advantages	Disadvantages
Finite Difference	Easy to use Doesn't require knowledge of the simulation	Gives biased estimates Requires several runs of the simulation
Perturbation Analysis	Requires one run of the simulation Gives estimates with consistency and unbiasedness	Requires knowledge of the simulation May require variations (SPA, IPA)
Simultaneous Perturbation	Requires only two runs of the simulation Doesn't require knowledge of the simulation	May give estimates with large variances
Frequency Domain	Requires only one run of the simulation	May give biased estimates

Table 4.1: Comparison of gradient estimation methods

	Fixed	Scaled
Step	5	1%
Actual	470	470
Upper	475	474.7
Lower	465	465.3

Table 4.2: Step sizes for process parameter - temperature in Celsius

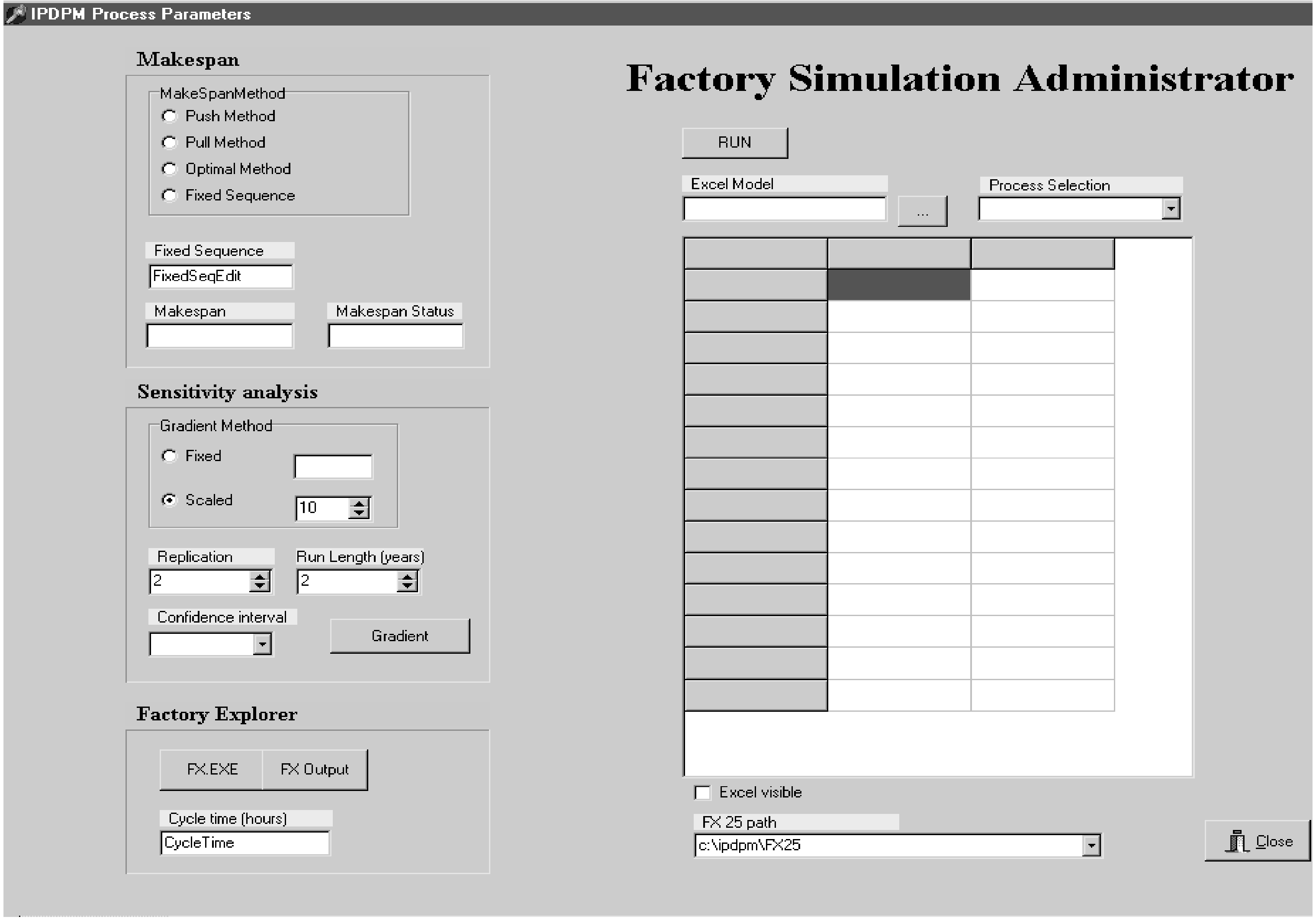
mate from the central difference method will usually have lesser bias than the forward or backward difference.

The user first chooses the parameter on which he wishes to perform sensitivity analysis, then he chooses the way the step size is determined. There are two ways in which the step size is determined.

- **Fixed step size** The user can choose a fixed step size which he sets himself.
- **Scaled step size** Here the user can select the percentage of the process parameter to be used as the step size.

Table 4.2 provides an example for step size selection. Then the user has to choose the number of replications and the time period (Number of years) for which the simulation is being run. The user also chooses the percentage confidence level (95%, 97.5% and 99%), for the gradient estimate. These factors have an impact on the precision of the gradient estimate obtained. A snapshot of the Factory Administrator is shown in figure 4.4.

After the user has selected the necessary parameters, which are needed to conduct sensitivity analysis, the process is started. The simulation is conducted



Makespan

MakeSpanMethod
○ Push Method
○ Pull Method
○ Optimal Method
○ Fixed Sequence

Fixed Sequence: FixedSeqEdit

Makespan:

Makespan Status:

Sensitivity analysis

Gradient Method
○ Fixed
● Scaled 10

Replication: 2

Run Length (years): 2

Confidence interval:

Factory Explorer

Cycle time (hours): CycleTime

Excel Model:

Process Selection:

□ Excel visible

FX 25 path: c:\ipdpm\FX25

Figure 4.4: Snapshot of the factory administrator

with many replications with the process parameter at the upper level. The cycle time for each replication is obtained and stored in an array. Then the simulation is run with the process parameter at the lower level. Again the cycle time for each replication is obtained. The gradient estimate is now made for each individual replication. The confidence interval is built using the gradient estimates obtained over replications. The mean gradient estimate along with the half width of the confidence interval is displayed to the user. In a single click, the user can run the process simulation, cluster tool simulation and the factory simulation together and thus perform analysis on the sensitivity of system level measures to the process parameters.

4.7 Example

In this section, an example simulation model used to demonstrate the HSE-based decision support tool is described. The simulation model has one product, Wafer1. The initial input rate of 2000 wafers/week, which is ramped to 5000 wafers/week by the end of two years. The wafers enter the manufacturing system in lots of twenty. There are three types of tools CLEAN, TI_LINER and W_CVD. The wafer goes through these three tools seven times, for one contact layer and six via layers. The process parameters for TI_LINER include thickness, pressure, power and spacing. The process parameters for W_CVD include thickness, pressure, temperature, mass flow of H_2 and mass flow of WF_6. Some of the cluster tool parameters are pump-down time, OD (Orient and Degas) time and Robot move time. Sensitivity analysis can be performed for performance metrics like Cycle time over any of these input parameters.

4.8 Summary

The implementation of a gradient estimation method for sensitivity analysis as a decision support tool has been described in this chapter. This tool will help evaluate the impact of process parameters on system-level outputs in a system and hence effect improvements in the utilization of tool resources.

Chapter 5

Stochastic Optimization

5.1 Introduction

This chapter describes a methodology to acquire quality solutions for the problem of allocating machines in a manufacturing system. The goal is to leverage the existing gradient estimation techniques used for sensitivity analysis and build an optimization algorithm which will find optimal allocation of machines in the system. The machine allocation problem studied here is one of vital importance to the semiconductor industry, which invests a great deal of money in equipment. Selecting the proper set of tools is important to satisfying throughput requirements and budget requirements and minimizing average cycle time.

5.2 General Formulation

We formulate the problem as follows. The objective is to minimize $E[T]$, the average cycle time of wafers through the factory. The decision variables X_{ij} are the number of tools of type j purchased at each workstation W_i. X_{ij} must be a non-negative integer.

We have a manufacturing system with n workstations $W_i, i = 1, 2, \ldots, n$. For each workstation W_i, there are z_i types of tools available. Each workstation can have tools from one or more types. The total number of decision variables is p.

$$p = \sum_{i=1}^{n} z_i \tag{5.1}$$

The cost of one tool of type j for workstation W_i is C_{ij}, and the capacity of one such tool is μ_{ij}(wafers per unit time). The decision-maker has a fixed budget of M dollars for purchasing tools, so that the total tool cost cannot exceed M. Also, the manufacturing system must achieve a throughput of λ (wafers per unit time). If μ_i is the capacity at workstation i, then $\mu_i = \sum_{j=1}^{z_i} X_{ij}\mu_{ij}$ and μ_i must be greater than λ. We can write the constraints as follows:

$$\sum_{j=1}^{z_i} X_{ij}\mu_{ij} > \lambda \text{ for all } i, \tag{5.2}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{z_i} X_{ij}C_{ij} \leq M. \tag{5.3}$$

Note that finding a solution that satisfies Equations (5.2) and (5.3) is equivalent to solving the knapsack problem.

	Workstations		
Tool Configuration	CLEAN $i = 1$	Ti_LINER $i = 2$	W_CVD $i = 3$
$j = 1$	\$10,000	\$20,000	\$44,000
$j = 2$	\$5,500	\$28,000	\$31,000
$j = 3$	\$6,000	\$30,000	\$30,000
$j = 4$	\$11,000	\$19,000	\$46,000

Table 5.1: Tool costs C_{ij}

	Workstations		
Tool Configuration	CLEAN $i = 1$	Ti_LINER $i = 2$	W_CVD $i = 3$
$j = 1$	6.25	12.5	25
$j = 2$	3.125	18.125	18.75
$j = 3$	3.75	20	17.5
$j = 4$	6.875	12.5	24.375
Required throughput 50			
All numbers in wafers/hour			

Table 5.2: Tool capacities μ_{ij}

5.3 Example

The factory has three workstations CLEAN, TI_LINER and W_CVD. Table 5.1 lists the costs for each tool type. Table 5.2 lists the single-tool capacity of each tool type.

The required throughput from the system is 50 wafers/hour and the budget constraint is \$400,000.

	Workstations		
Tool Configuration	CLEAN $i = 1$	Ti_LINER $i = 2$	W_CVD $i = 3$
$j = 1$	3	0	0
$j = 2$	0	0	4
$j = 3$	5	5	0
$j = 4$	6	0	0

Table 5.3: A Feasible solution matrix X_{ij}

The solution θ must satisfy throughput constraints:

$$6.25X_{11} + 3.125X_{12} + 3.75X_{13} + 6.875X_{14} \quad > \quad 50, \tag{5.4}$$

$$12.5X_{21} + 18.125X_{22} + 20X_{23} + 12.5X_{24} \quad > \quad 50, \tag{5.5}$$

$$25X_{31} + 18.75X_{32} + 17.5X_{33} + 24.375X_{34} \quad > \quad 50, \tag{5.6}$$

In addition, θ must satisfy the budget constraint:

$$\begin{aligned} 10,000X_{11} + 5500X_{12} + 6000X_{13} + 11000X_{14} + 20000X_{21} + \\ 28000X_{22} + 30000X_{23} + 19000X_{24} + 44000X_{31} + \\ 31000X_{32} + 30000X_{33} + 46000X_{34} \leq 400,000, \end{aligned} \tag{5.7}$$

Table 5.3 describes one feasible solution for this example. The total cost of the manufacturing system is \$400,000, and the workstations have the following capacities:

$\mu_1 = 3\mu_{11} + 5\mu_{13} + 6\mu_{14} = 78.75$ wafers/hour,

$\mu_2 = 5\mu_{23} = 100$ wafers/hour,

$\mu_3 = 4\mu_{32} = 75$ wafers/hour.

The average cycle time of the system can be estimated by simulation. The wafers arrive in one lot of 25 wafers every 0.5 hours. The interarrival times and the processing times are exponentially distributed. The mean processing time on a tool of type j at workstation i is $25/\mu_{ij}$. The number of lots that visit each tool in a workstation is proportional to the tool's capacity. The manufacturing system is simulated for a period of one year with ten replications. Using this model, the mean cycle time for each lot has been estimated to be 7.70 hours, and the 99% confidence interval is 7.699 ± 0.038.

5.4 Solution Approach

The budget and throughput constraints bound the set of feasible solutions. Purchasing too few tools will give insufficient capacity, but the budget constraint means that the tools must be selected carefully. To find a good solution to the problem, we will begin by using a heuristic to find a low-cost, feasible solution. Then, we will use a gradient-based search procedure to find better solutions. The gradient gives us information about adding tools that reduce the cycle time the most.

We have developed a search algorithm that uses gradient information to direct the search through the discrete solution space. The emphasis here is on using gradient estimation methods as they build on sensitivity analysis modules that were developed. The gradient, estimated by forward differences, provides a search direction. The gradient estimation uses forward differences to avoid violating the throughput constraints. For example, if $X_{ij} = 0$ at some point of the iteration, then we cannot use central differences as we have to estimate cycle times at $X_{ij} = -1$ and $X_{ij} = 1$. Though simultaneous perturbation method's advantage of being computationally efficient is useful, it was not applicable for the same reason. The search algorithm proposed also doesn't allow increasing one tool while simultaneously decreasing another tool. The gradient can be estimated through forward differences, where we have to estimate cycle times at $X_{ij} = 0$ and $X_{ij} = 1$. The search modifies the search direction to avoid reducing the number of tools or trying to add any tools that are too expensive. The search then determines the maximum step that remains feasible with respect to the budget constraint. Finally, the search moves to a nearby integer point that is feasible.

5.4.1 Notation

The following notation is used for the algorithm:

k = Iteration number,

c = Size of the perturbation,

$\hat{f}_r(\theta_k)$ = Average cycle time at point θ_k obtained by the r^{th} simulation run,

N = Number of replications,

e_q = Unit vector in direction q,

θ_k = Solution after iteration k. $\theta_k = (X_{11}, \ldots, X_{nz_n})$,

U_{ij} = Capacity per dollar of tools of type j at workstation i,

a_k = Step size at iteration k,

$\lfloor\!\lfloor x \rfloor\!\rfloor$ = Greatest integer less than or equal to x,

$\lceil\!\lceil x \rceil\!\rceil$ = Smallest integer greater than or equal to x.

5.4.2 Description of the Algorithm

The algorithm follows five steps.

Step 1: Initialization.

$k = 0$.

$c = 1$.

Initialize the solution vector θ_0 according to the following heuristic:

Heuristic for initial feasible solution vector:

For each workstation $i = 1, \ldots, n$:

Calculate $U_{ij} = \mu_{ij}/C_{ij}$ for each tool type.

Let $U_i^* = \max\{U_{i1}, \ldots, U_{iz_i}\}$.

Let y_i equal the number of tool types j such that $U_{ij} = U_i^*$.

For these y_i tool types, let $X_{ij} = \lceil\!\lceil \lambda/(y_i\mu_{ij}) \rceil\!\rceil$.

	Workstations		
Tool Configuration	CLEAN $i = 1$	Ti_LINER $i = 2$	W_CVD $i = 3$
$j = 1$	6.25	6.25	5.68
$j = 2$	5.68	6.47	6.05
$j = 3$	6.25	6.67	5.83
$j = 4$	6.25	6.58	5.30

Table 5.4: U_{ij} in 10^{-4} wafers/(dollar hours)

	Workstations		
Tool Configuration	CLEAN $i = 1$	Ti_LINER $i = 2$	W_CVD $i = 3$
$j = 1$	3	0	0
$j = 2$	0	0	3
$j = 3$	5	3	0
$j = 4$	3	0	0

Table 5.5: Solution matrix X_{ij} after step 1a

For the other $z_i - y_i$ tool types, let $X_{ij} = 0$.

If $\sum_{i=1}^{n} \sum_{j=1}^{z_i} X_{ij} C_{ij} > M$, stop.

Step 2: Gradient Estimation.

For each component of θ_k,$q = 1, \ldots, p$, estimate $\hat{g}_q(\theta_k)$ as follows.

With $N = 10$,

$$(\hat{g}_q(\theta_k))_N = \frac{1}{N} \sum_{r=1}^{N} \left(\frac{\hat{f}_r(\theta_k + ce_q) - \hat{f}_r(\theta_k)}{c} \right). \tag{5.8}$$

Note that this will require $N(p + 1)$ simulation runs.

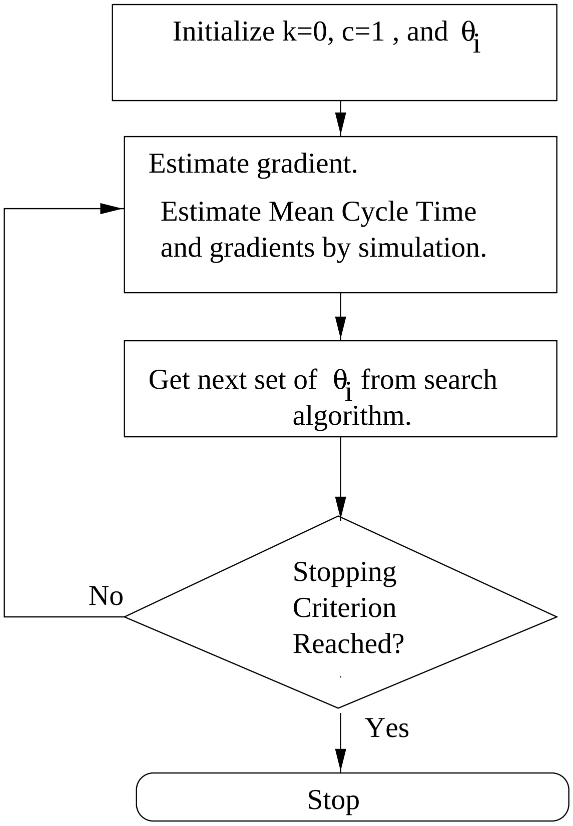

Initialize k=0, c=1 , and θi
Estimate gradient.
Estimate Mean Cycle Time
and gradients by simulation.
Get next set of θi from search
algorithm.
Stopping
Criterion
Reached?
No
Yes
Stop

Figure 5.1: Representation of the search algorithm

Step 3: Solution update.

Let $B = M - \sum_{i=1}^{n}\sum_{j=1}^{z_i} X_{ij}C_{ij}$.

Let $d_{ij} = \hat{g}_q(\theta_k)$ where X_{ij} is the q-th component of θ_k. [1]

If $d_{ij} > 0$, let $d_{ij} = 0$. This avoids reducing any X_{ij}.

If $C_{ij} > B$, let $d_{ij} = 0$. This avoids adding any tools that are too expensive.

Let

$$a = \frac{-B}{\sum_{i=1}^{n}\sum_{j=1}^{z_i} d_{ij}C_{ij}} \tag{5.9}$$

[1] Note $\hat{g}_q$ is a vector representation of the gradient whereas d_{ij} is the matrix representation. $q = \sum_{l=1}^{i-1} z_l + j$

if some $d_{ij} < 0$. Otherwise $a = 0$.

Create θ_{k+1} by adding $\llcorner\!\llcorner -ad_{ij} \lrcorner\!\lrcorner$ to X_{ij}.

If all $\llcorner\!\llcorner -ad_{ij} \lrcorner\!\lrcorner = 0$, then identify the smallest (most negative) d_{ij} and create θ_{k+1} by adding 1 to X_{ij}.

θ_{k+1} is feasible with respect to the throughput and budget constraints, since all $d_{ij} \leq 0$ and

$$\sum_{i=1}^{n} \sum_{j=1}^{z_i} (X_{ij} + \llcorner\!\llcorner -ad_{ij} \lrcorner\!\lrcorner) C_{ij} \leq M - B - a \sum_{i=1}^{n} \sum_{j=1}^{z_i} d_{ij} C_{ij} = M. \tag{5.10}$$

Step 4: If $\theta_{k+1} = \theta_k$, then stop. Else, add 1 to k and go to Step 2.

5.4.3 Example

The algorithm is applied to the example manufacturing system considered.

Step 1: Initialize the solution vector θ_0 as follows.

Table 5.4 shows U_{ij} for each tool type.

For workstation 1,

$U_1^* = 6.25$.

$U_{11} = U_{13} = U_{14} = U_1^*$.

$y_1 = 3$.

$X_{11} = \ulcorner\!\ulcorner \lambda / 3\mu_{11} \urcorner\!\urcorner = \ulcorner\!\ulcorner 50/18.75 \urcorner\!\urcorner = 3$.

$X_{13} = \ulcorner\!\ulcorner \lambda / 3\mu_{13} \urcorner\!\urcorner = \ulcorner\!\ulcorner 50/11.25 \urcorner\!\urcorner = 5$.

$X_{14} = \ulcorner\!\ulcorner \lambda / 3\mu_{14} \urcorner\!\urcorner = \ulcorner\!\ulcorner 50/20.625 \urcorner\!\urcorner = 3$.

For workstation 2,

$U_2^* = 6.67.$

$U_{23} = U_2^*.$

$y_2 = 1.$

$X_{23} = \lceil\!\lceil \lambda/\mu_{23} \rceil\!\rceil = \lceil\!\lceil 50/20 \rceil\!\rceil = 3.$

For workstation 3,

$U_3^* = 6.05.$

$U_{32} = U_3^*.$

$y_3 = 1.$

$X_{32} = \lceil\!\lceil \lambda/\mu_{32} \rceil\!\rceil = \lceil\!\lceil 50/18.75 \rceil\!\rceil = 3.$

All other $X_{ij} = 0.$

$\sum_{i=1}^{n} \sum_{j=1}^{z_i} X_{ij} C_{ij} = 276,000$, which is less than M.

Step 2: We use the forward difference formula to estimate the gradients.

The estimated gradients are illustrated in Table 5.6.

The number of simulation runs for this gradient computation will be $(12+1)10 = 130$.

Step 3: Update the solution.

B = 400,000 - 276,000 = 124,000.

$d_{ij} = 0$ for $X_{12}, X_{13}, X_{21}, X_{24}$, since $\hat{g}_q(\theta_k) > 0$.

Tool type	Cycle time higher(in hours)	Cycle time lower(in hours)	Gradient
X_{11}	12.627	13.923	-1.296
X_{12}	13.355	13.923	-0.568
X_{13}	13.147	13.923	-0.776
X_{14}	12.555	13.923	-1.368
X_{21}	12.811	13.923	-1.112
X_{22}	12.480	13.923	-1.443
X_{23}	12.342	13.923	-1.581
X_{24}	12.703	13.923	-1.220
X_{31}	11.125	13.923	-2.798
X_{32}	11.250	13.923	-2.673
X_{33}	11.320	13.923	-2.603
X_{34}	11.075	13.923	-2.848

Table 5.6: Gradient estimation

d_{ij} =$\hat{g}_q(\theta_k)$ for every other tool because $\hat{g}_q(\theta_k) \leq 0$ and $C_{ij} < B$.

$$a = \frac{124,000}{\begin{array}{c}(1.296(10,000) + 0.568(5,500) + 0.776(6,000) + 1.368(11,000) \\ +1.112(20,000) + 1.443(28,000) + 1.581(30,000) + 1.220(19,000) \\ +2.798(44,000) + 2.673(31,000) + 2.603(30,000) + 2.848(46,000))\end{array}} \tag{5.11}$$

$a = 0.212$

The approximated gradients and the updated solution vector are shown in Table 5.7.

Tool type	θ_k	$-ad_{ij}$	$\llcorner -ad_{ij} \lrcorner$	θ_{k+1}
X_{11}	3	0.275	0	3
X_{12}	0	0.121	0	0
X_{13}	5	0.165	0	5
X_{14}	3	0.290	0	3
X_{21}	0	0.236	0	0
X_{21}	0	0.306	0	0
X_{21}	3	0.336	0	3
X_{21}	0	0.259	0	0
X_{31}	0	0.594	0	0
X_{32}	3	0.567	0	3
X_{33}	0	0.553	0	0
X_{34}	0	0.605	1	1

Table 5.7: Solution update

Since all $\llcorner -ad_{ij} \lrcorner$ are zero, we increment X_{ij} with the highest gradient by one.

Step 4: Since $\theta_{k+1} \neq \theta_k$, $k = k + 1 = 2$ and we go to Step 2.

5.5 Experiments

5.5.1 Architecture

The administrator, the input template files, the output files and the simulation model files are the four components of the experimental architecture. The administrator controls the other three components. It also executes the search algorithm. This architecture along with the simulation engine (Factory Explorer) is depicted in Figure 5.2.

The input template files contain the input data for a number of simulation

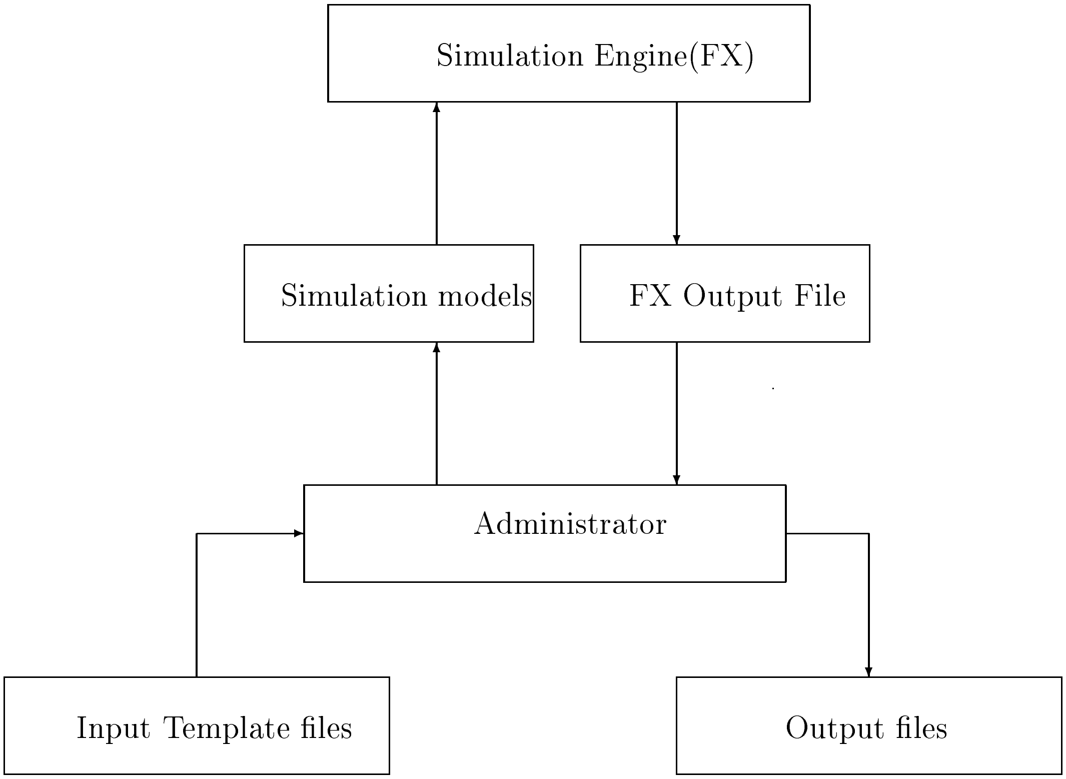


Figure 5.2: Experimental architecture

models. The administrator reads the input data from the input template file and runs the heuristic to find the initial feasible solution. This is used to populate the simulation model file. During the search algorithm, the administrator updates the simulation models, executes the simulation engine, and reads the simulation output files. When the search ends, the administrator outputs the search results.

5.5.2 Description of the Input Template Files

The input template file contains input data for a series of experiments. There are two such input template files. Each file specifies 160 problem instances. Two different methods were used to create the problem instances. The primary difference is the correlation of capacity and cost. In practice, we would expect a faster (high capacity) tool to be more expensive. In the real world, we would ideally expect the cost of a tool type to increase when it can process faster. In Problem Set 1, the capacity is not correlated to the cost, while in Problem Set 2, they are correlated and the capacity is chosen based on the cost.

Problem Set 1

The input is built using the following generation parameters:

P = Cost factor for tool types = \$1000,

λ = 100 wafers/hours,

n = Number of workstations = 5,

r = Expected number of tools per workstation = 2 or 10,

z_i = Number of tool types per workstation = 2 or 5,

α = Lower bound of cost range = 0.5 or 0.8,

β = Multiplier for budget = 1 or 3.

For each combination of parameter values, we generate ten instances. For each instance, the tool capacities and costs are generated by using the following procedure.

```
For $i = 1$ to $n$
    For $j = 1$ to $z_i$
        Choose $a_{ij} \in [0, 2]$
        Let $\mu_{ij} = a_{ij}(\lambda / r)$
        Choose $b_{ij} \in [\alpha, 1]$
        Let $C_{ij} = b_{ij} P$
$M = \beta n r P$
```

We have four parameters to vary. They are r, z_i, α and β. Each of these parameters can take two values. Hence we have sixteen combinations of these parameters.

Problem Set 2

The input is built using the following generation parameters:

P = Cost Factor for tool type = \$1000,

λ = 100 wafers/hours,

n = Number of workstations = 5,

r = Expected number of tools per workstation = 2 or 10,

z_i = Number of tool types per workstation = 2 or 5,

e = Shape of the correlation = 0.5 or 1,

α = Lower bound of cost range = 0.5,

β = Multiplier for budget = 1 or 3.

For each combination of parameter values, we generate ten instances. For each instance, the tool capacities and costs are generated by using the following procedure.

```
For $i = 1$ to $n$
    For $j = 1$ to $z_i$
        Choose $b_{ij} \in [\alpha, 1]$
        Let $a_{ij} = 2(b_{ij})^e$
        Let $\mu_{ij} = a_{ij}(\lambda/r)$
        Let $C_{ij} = b_{ij}P$
$M = \beta nrP$
```

5.5.3 Description of the Optimization Process

The administrator takes the μ_{ij} and C_{ij} values from the input template file along with n and z_i. Using these parameters the administrator runs the heuristic for the initial solution vector and determines the X_{ij} values. All the parameters generated are now used to populate the simulation models. The simulation models require parameters like name of the process step, name of the tool used, number of tools used and the percentage of lots which visit each tool in a workstation.

The nomenclature of different process steps is done using the workstation numbers. For example, the step at workstation 1 is named as “n1”. Similarly the tool type $j = 2$ at workstation 1 is named as “n1j2”. Further the tool processing times are generated using μ_{ij}. After the model has been populated, the administrator uses the simulation engine to run simulations using the model. The simulation engine then outputs the performance metrics to a text file. The text file is read by the administrator to obtain the necessary output metrics and based on the output metrics, the administrator decides the next iteration step.

5.5.4 Description of the Simulation Model

We have one product, Wafer, which enters the system at one lot of 25 wafers every 0.25 hours. The interarrival times and the processing times are exponentially distributed. The mean processing time on a tool of type j at workstation i is $25/\mu_{ij}$. The number of lots that visit each tool in a workstation is proportional to the tool's capacity. μ_{ij} and C_{ij} are obtained from the input files. While the initial number of tools at each workstation is obtained from the heuristic, the updated number of tools are obtained from the search algorithm. Each lot will visit each workstation starting with workstation 1 and ending with workstation

5. Each replication in a simulation run is conducted for one year, which means that approximately 35000 lots are processed in every replication.

5.5.5 Output Files

After an instance has been solved, the administrator outputs a few important metrics: total cost of the tools, the bottleneck workstation and its capacity and the estimated cycle time of that configuration (The bottleneck is the workstation with the smallest total capacity). These statistics are gathered after the initial heuristic has been completed and after the search algorithm completes its run. Three performance metrics are calculated to estimate the performance of the algorithm:

$$\begin{aligned} Cost\,Metric &= \frac{Cost_x - Cost_y}{M} && (5.12)\\ Capacity\,Metric &= \frac{Capacity_x - Capacity_y}{\lambda} && (5.13)\\ Cycle\,Time\,Metric &= \frac{Cycle\,Time_x}{Cycle\,Time_y} && (5.14) \end{aligned}$$

where x means after the search and y means after the heuristic (before the search).

Experiment Parameters					Instances with	Cost	Capacity	Cycle Time
n	r	z_i	α	β	Feasible Solutions	Metric	Metric	Metric
5	2	2	0.5	1	10	0.168	0.288	0.601
5	2	2	0.5	3	10	0.620	1.552	0.496
5	2	2	0.8	1	0			
5	2	2	0.8	3	10	0.641	1.640	0.465
5	2	5	0.5	1	10	0.348	0.681	0.498
5	2	5	0.5	3	10	0.740	2.585	0.681
5	2	5	0.8	1	9	0.090	0.183	0.751
5	2	5	0.8	3	10	0.655	2.123	0.684
5	10	2	0.5	1	9	0.377	0.310	0.370
5	10	2	0.5	3	10	0.762	1.467	0.451
5	10	2	0.8	1	5	0.185	0.162	0.527
5	10	2	0.8	3	10	0.667	1.096	0.478
5	10	5	0.5	1	10	0.528	0.565	0.399
5	10	5	0.5	3	10	0.824	1.681	0.501
5	10	5	0.8	1	10	0.365	0.434	0.337
5	10	5	0.8	3	10	0.801	1.471	0.494

Table 5.8: Results for problem set 1

Experiment Parameters					Instances with	Cost	Capacity	Cycle Time
n	r	z_i	e	β	Feasible Solutions	Metric	Metric	Metric
5	2	2	0.5	1	0			
5	2	2	0.5	3	10	0.501	0.827	0.381
5	2	2	1	1	0			
5	2	2	1	3	10	0.297	0.354	0.485
5	2	5	0.5	1	0			
5	2	5	0.5	3	10	0.596	0.915	0.236
5	2	5	1	1	0			
5	2	5	1	3	10	0.397	0.666	0.409
5	10	2	0.5	1	3	0.122	0.093	0.508
5	10	2	0.5	3	10	0.668	0.771	0.375
5	10	2	1	1	0			
5	10	2	1	3	10	0.482	0.412	0.467
5	10	5	0.5	1	6	0.112	0.057	0.625
5	10	5	0.5	3	10	0.688	0.733	0.230
5	10	5	1	1	1	0.172	0.110	0.504
5	10	5	1	3	10	0.625	0.505	0.327

Table 5.9: Results for problem set 2

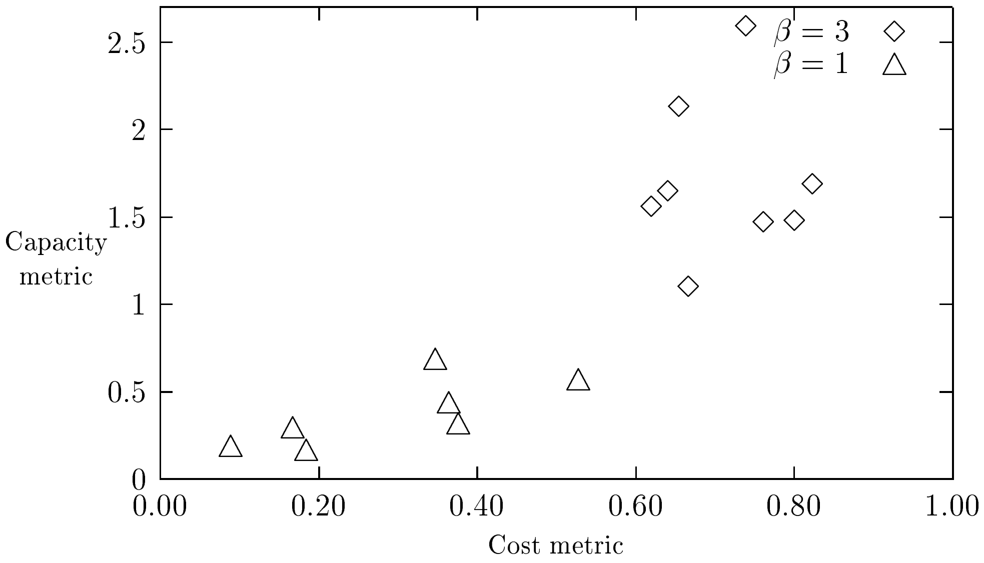


Figure 5.3: Cost metric vs capacity metric for problem set 1

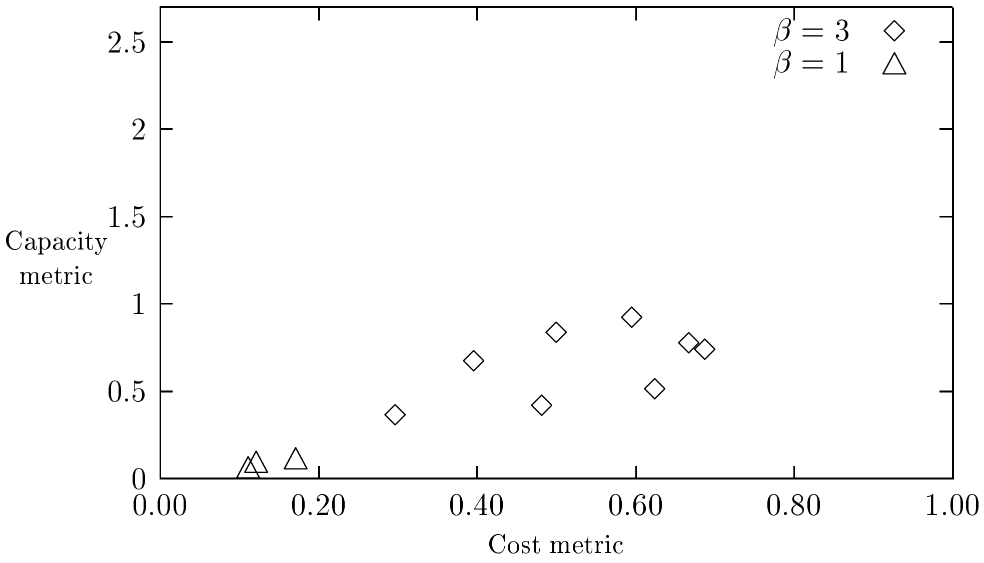


Figure 5.4: Cost metric vs capacity metric for problem set 2

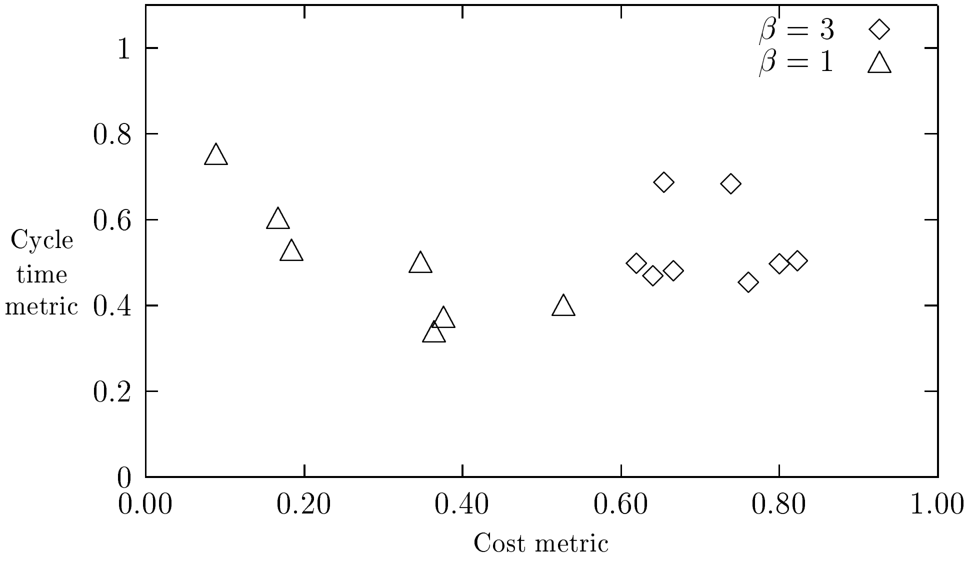


Figure 5.5: Cost metric vs cycle time metric for problem set 1

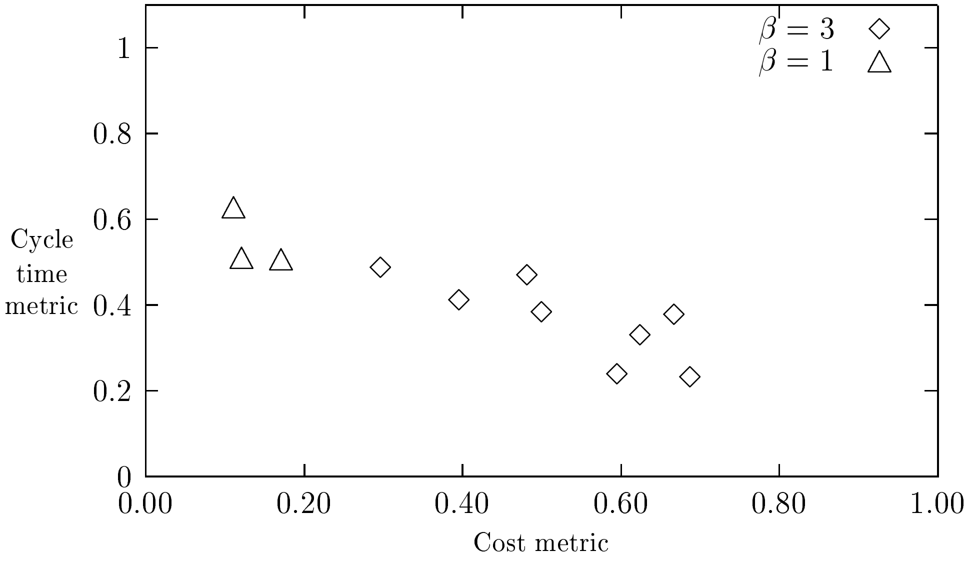


Figure 5.6: Cost metric vs cycle time metric for problem set 2

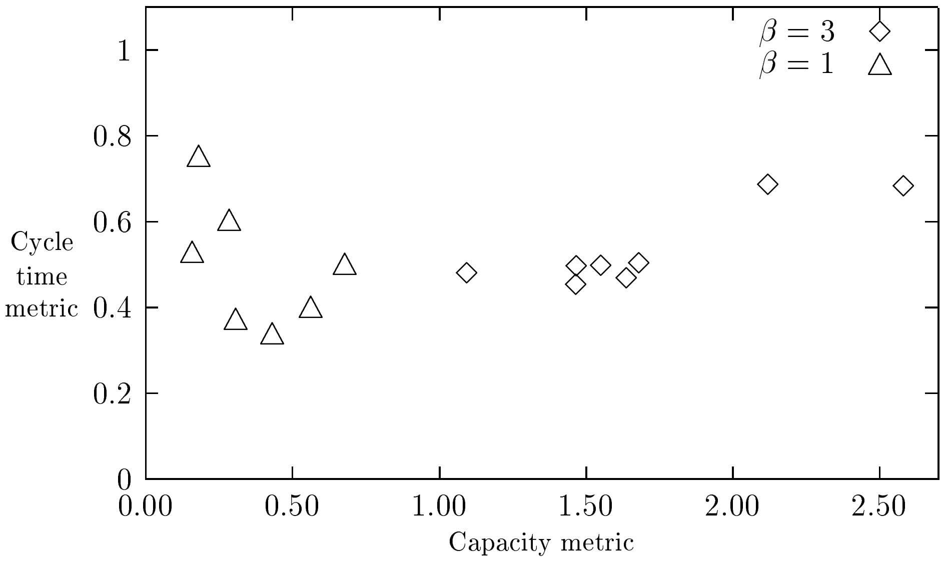


Figure 5.7: Capacity metric vs cycle time metric for problem set 1

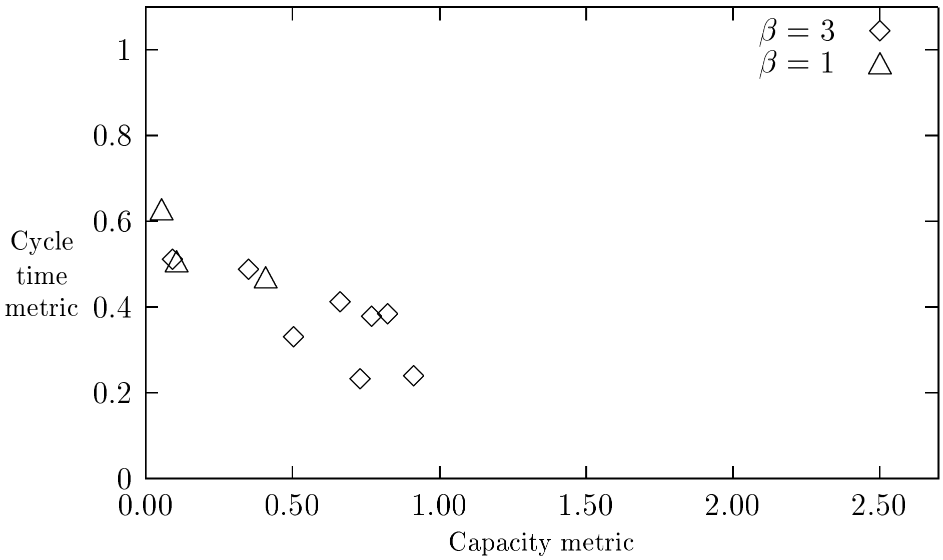


Figure 5.8: Capacity metric vs cycle time metric for problem set 2

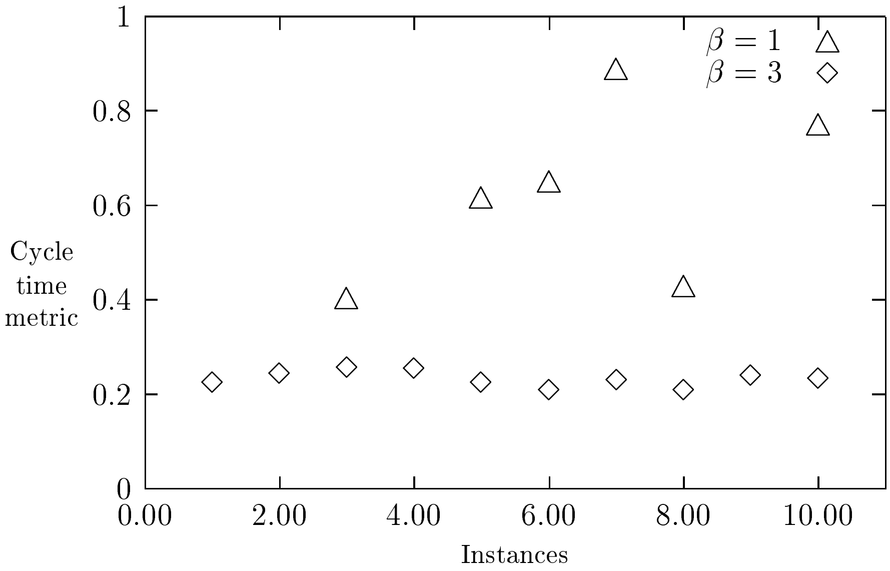


Figure 5.9: Cycle time metric over replications

5.6 Experiment Results

The performance metrics discussed above provide insight into the performance about the algorithm. The cost metric describes how much more money has been spent to purchase extra capacity and reduce cycle time. The bottleneck capacity metric describes of how much capacity has been added with respect to λ. Similarly cycle time reduction is described by the cycle time metric.

The correlation between capacity and cost is an important factor as we can see in Tables 5.8 and 5.9. Each row in these tables has the average performance for the instances with feasible solutions. We can see that when the capacity is correlated to the cost, we get significant reduction in cycle time with less additional capacity and less money compared to the case when the capacity and the cost are not correlated.

From Figures 5.3, 5.4, 5.5, 5.6, 5.7 and 5.8, we can observe that the search algorithm is able to reduce cycle time significantly. On some problem sets the average reduction is over 70%.

An important trend of interest is the impact of β on the performance of the algorithm. Even with $\beta = 1$, we can see good improvement in cycle time performances, while $\beta = 3$ did not improve very much on $\beta = 1$ for Problem Set 1.

Figure 5.9 illustrates the sensitivity of cycle time reduction to the budget constraint. The figure uses the cycle time metric for 10 instances obtained using Problem set 2 for a specific configuration ($n = 5$, $r = 10$, $z_i = 5$, $e = 0.5$). Note that, for $\beta = 1$, the heuristic found feasible solutions in only six instances. When $\beta = 1$, the average capacity metric is 0.057, and the cycle time metric ranges from 0.4 to 0.9. However, when $\beta = 3$, the average capacity metric is 0.733, and the cycle time metric is always near 0.233. The additional funds are able to purchase more equipment and reduce cycle times dramatically.

5.7 Summary

An equipment selection problem was formulated with minimizing cycle time as the objective and with constraints on the budget and minimum throughput on the system. A search algorithm has been presented where we find an initial solution through a heuristic and then develop the solution further by using a gradient-based search. The search algorithm was then evaluated using test cases generated using an experimental design architecture. It can be seen from the experiments that the algorithm performs quite well over a range of problem instances.

Chapter 6

Summary and Conclusions

6.1 Summary

The analysis of various performance metrics of stochastic systems such as semiconductor manufacturing fabs with respect to input parameters is a complicated process. Analytical models can be developed when the number of workstations is small and process flows are simple. But in many manufacturing systems, especially semiconductor manufacturing systems, complex product flows and a large number of process steps make analytical models intractable and simulation models inevitable. In this research, simulation models have been integrated with operations research techniques to provide valuable insight into the characteristics of the manufacturing system and help design these systems. Comparison of two gradient estimation techniques, finite differences and simultaneous perturbation, was performed using an analytical queueing system model as a benchmark. One of these techniques, finite differences, is then implemented in a decision support tool with a Heterogeneous Simulation Environment (HSE) for process engineers.

A gradient-based stochastic optimization procedure was then implemented to obtain quality solutions to the manufacturing system design problem of equipment selection.

6.2 Anticipated Impact

Advanced simulation modeling tools will have impact in the future as computing power increases and simulation-based optimization and sensitivity analysis techniques become more easily applicable.

Sensitivity analysis aids process control by providing a better picture of process parameters with respect to output metrics and thus helping the process engineer evaluate the status of the process he is controlling with regard to the semiconductor fabrication plant. It facilitates interactive decision making involving both industrial and process engineers.

Discrete stochastic optimization can be used to optimize real world stochastic systems. When we consider the application of the optimization technique described here, it is applicable to equipment selection for any manufacturing system. It can help reduce costs in the design stage itself by providing savings both in terms of budget and reduced cycle time.

6.3 Future Work

Future work can involve integration of optimization techniques in decision-making tools for managers, which can be an advanced version of the decision support tool for process engineers. The application of the stochastic optimization procedure to a general class of resource allocation procedures could be studied. Further, the algorithm can be benchmarked for performance by making a comparison with other techniques including random search and simulated annealing.

Bibliography

[1] Alrefaei, M.H. and S.Andradóttir., "Temperature for discrete stochastic optimization," *Management Science*, pp.748-764, 1999.

[2] Alrefaei, M.H. and S.Andradóttir., "A new search algorithm for discrete stochastic optimization," *Proceedings of the 1995 Winter Simulation Conference*, ed. C. Alexopoulos, K. Kang,W.R. Lilegdon, and D.Goldsman, pp.236-241, 1995.

[3] Andradóttir, Sigrún, "A method for discrete stochastic optimization," *Management Science*, Vol. 41, pp.1946-1961, 1995.

[4] Andradóttir, Sigrún, "A global search method for discrete stochastic optimization," *SIAM Journal on Optimization*, Vol. 6, pp.513-530, 1996.

[5] Banks, Jerry, Ed., *Handbook of Simulation*, Wiley Interscience., New York, NY, 1998.

[6] Cassandras, Christos G., L.Dai and C.G.Panayiotou, "Ordinal optimization for a class of deterministic and stochastic discrete resource allocation prob-

lems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 7 pp.881-900, 1998.

[7] Cassandras, Christos G., *Discrete event systems : modeling and performance analysis*, Irwin, Homewood, IL., 1993.

[8] *Factory Explorer, User's Manual* Wright Williams and Kelley, 1998.

[9] Fu, Michael C., "Optimization via simulation : a review," *Annals of Operations Research*, Vol.53, pp.199-247, 1994.

[10] Frenk, Hans, M.Labbé, M.Van Vliet and S.Zhang, "Improved algorithms for machine allocation in manufacturing systems," *Operations Research*, Vol. 42, No. 3, pp.523-530, 1994.

[11] Gerencsér László., S.D.Hill, and Z.Vágó, "Optimization over discrete sets via SPSA," *Proceedings of the 1999 Winter Simulation Conference*, pp.466-470, 1999.

[12] Glynn, P. W., "Likelihood ratio gradient estimation of stochastic systems," *Communications of the ACM*, Vol. 33, pp.75-84, 1990.

[13] Herrmann, J.W. et al. "Understanding the impact of equipment and process changes with a heterogeneous semiconductor manufacturing simulation environment,"*Submitted for publication to the Winter Simulation Conference 2000.*

[14] Ho, Y.-C., and X.-R.Cao, *Perturbation Analysis of Discrete Event Dynamical Systems*, Kluwer Academic Publishers, Norwell, MA., 1991.

[15] Jacobson, S.H., "Convergence results for harmonic gradient estimators," *ORSA Journal of Computing*, Vol.6, pp.381-397, 1994.

[16] Kempf, Karl G., "Simulating semiconductor manufacturing systems: successes, failures, and deep questions," *Proceedings of the 1996 Winter Simulation Conference*, pp.3-11, 1996.

[17] Kiefer, J. and J.Wolfowitz,"Stochastic estimation of the maximum of a regression function," *Annals of Mathematical Statistics*, Vol. 23, pp.462-466, 1952.

[18] Kleijnen Jack P. C., and Willem J. H. van Groenendaal "Regression metamodels and design of experiments," *Proceedings of the 1996 Winter Simulation Conference*, pp.1433-1439, 1996.

[19] L'Ecuyer, P., N.Giroux, and P.W.Glynn., "Stochastic optimization by simulation: numerical experiments with the M/M/1 queue," *Management Science*, Vol. 40, pp.1245-1261, 1994.

[20] L'Ecuyer, P., "An overview of derivative estimation," *Proceedings of the 1991 Winter Simulation Conference*, pp.207-217, 1991.

[21] Law, Averill M. and K.W.David., *Simulation Modeling and Analysis*, McGraw Hill, Singapore, 1991.

[22] Meyyappan, M., *Computational Modeling in Semiconductor Processing*, Artech House, Norwood, MA., 1995.

[23] Nguyen, Manh-Quan T., "Improving cluster tool performance by finding the optimal sequence and cyclic sequence of wafer handler moves," MS Thesis, Mechanical Engineering Department, University of Maryland, College Park, 2000.

[24] Panico, Joseph A., *Queueing theory: a study of waiting lines for Business, Economics and Science*, Prentice-Hall Inc., Englewood Cliffs, N.J., 1969.

[25] Pflug, Georg Ch., *Optimization of Stochastic Models: The Interface between Simulation and Optimization*, Kluwer Academic Publishers, Norwell, MA, 1996.

[26] Robinson S.,"Analysis of sample-path optimization," *Mathematics of Operations Research*, Vol.21, pp.513-528, 1996.

[27] Rubinstein. R.Y., and A.Shapiro, *Discrete event systems: Sensitivity analysis and Stochastic optimization by the Score function method*, Wiley, Chicester, West Sussex, England, 1993.

[28] Shantikumar J.G., and D.D.Yao, “On server allocation in multiple center manufacturing systems,” *Operations Research*, Vol. 36, No. 2, pp. 333-342, 1988.

[29] Spall, J.C., “Implementation of the simultaneous perturbation algorithm for stochastic optimization,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 3, pp.817-823, 1998.

[30] Spall, J.C., “Multivariate stochastic approximation using a simultaneous perturbation gradient approximation,” *IEEE Transactions on Automatic Control*, Vol. 37, pp. 332-341, 1992.

[31] Srinivasan R.S., “Modeling and performance analysis of cluster tools using petri nets,” *IEEE Transactions on Semiconductor Manufacturing*, Vol. 11, No. 3, pp.394-403, 1998.

[32] Suri, Rajan., “Perturbation analysis: The state of art and research issues explained via the GI/G/1 queue,” *Proceedings of the IEEE*, Vol. 77, No. 1, pp.114-137, IEEE, 1989.

[33] Uzsoy, R., C.Lee and L.A.Martin-Vega, “A review of production planning and scheduling in the semiconductor industry Part I: System characteristics, performance evaluation and production planning,” *IIE Transactions*, Vol. 24, No. 4, 1992.

[34] Wood S.C., “Simple performance models for integrated processing tools,” *IEEE Transactions on Semiconductor Manufacturing*, Vol. 9, No. 3, pp.320-328, 1996.

[35] Yan D. and H. Mukai, “Stochastic discrete optimization,” *SIAM Journal on Control and Optimization*, Vol. 30, pp.594-612, 1992.