

2008

The behavior of simulated annealing in stochastic optimization

Xiaoqing Gracie Gu
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Industrial Engineering Commons](#)

Recommended Citation

Gu, Xiaoqing Gracie, "The behavior of simulated annealing in stochastic optimization" (2008). *Retrospective Theses and Dissertations*. 15383.

<https://lib.dr.iastate.edu/rtd/15383>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

The behavior of simulated annealing in stochastic optimization

by

Xiaoqing Gu (Gracie)

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Industrial Engineering

Program of Study Committee:
Douglas D. Gemmill, Major Professor
John K. Jackman
Karin Dorman

Iowa State University

Ames, Iowa

2008

UMI Number: 1454679

Copyright 2008 by
Gu, Xiaoqing (Gracie)

All rights reserved

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

UMI Microform 1454679
Copyright 2008 by ProQuest LLC
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

TABLE OF CONTENTS

LIST OF TABLES	iv
LIST OF FIGURES	v
ACKNOWLEDGEMENTS	vi
ABSTRACT	vii
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	1
1.2 Literature review	3
1.3 Thesis outline	7
1.4 Glossary.....	7
1.5 Notations	8
CHAPTER 2 SIMULATED ANNEALING ALGORITHM	10
2.1 The SA algorithm applied to deterministic optimization	10
2.2 The SA algorithm applied to stochastic optimization	13
2.3 Stochastic problem with varying sample size.....	14
CHAPTER 3 IMPLEMENTATION	16
3.1 Discussion of parameters chosen DETERM.....	17
3.2 Parameters chosen – VARYN.....	20
CHAPTER 4 RESULTS AND COMPARISON	23
4.1 Deterministic problem results.....	23

4.2 Stochastic problem results.....	28
4.3 Variable sample size problem results.....	36
4.4 Overall comparison of results.....	41
CHAPTER 5 HYBRID ALGORITHM.....	44
5.1 Hybrid algorithm introduction.....	44
5.2 Hybrid algorithm implementation.....	45
5.3 Hybrid algorithm results.....	47
CHAPTER 6 CONCLUSIONS AND FUTURE WORK	51
APPENDIX.....	53
Response surface figures corresponding to objective functions.....	53
BIBLIOGRAPHY	55

LIST OF TABLES

Table 1. Notations.....	8
Table 2. Parameters setting.....	18
Table 3. Average percent error of the best cost found in DETERM.....	25
Table 4. The average best value found for each objective function.....	30
Table 5. Percent error comparisons in STOCHA- noise / Uniform Distribution.....	33
Table 6. Percent error comparisons in STOCHA- noise /Normal Distribution.....	34
Table 7. The average percentage of 40 data for 5 functions.....	34
Table 8. Percent error comparisons in VARYN.....	37
Table 9. Comparisons of percent error between STOCHA and VARYN.....	37
Table 10. Percent error of the best cost- Actual Z.....	41
Table 11. Notations-Hybrid Algorithm.....	44
Table 12. Changing population size.....	48
Table 13. Changing the number of generations.....	49
Table 14. Changing the probability of mutation.....	49
Table 15. Changing the probability of crossover.....	50

LIST OF FIGURES

Figure 1. Changing K while keeping N as a constant value.....	22
Figure 2. Changing N while keeping K as a constant value.....	22
Figure 3. Representative plots of performance of SA applied in F1, F3, F4 and F5.....	26
Figure 4. The distributions of the best cost values in STOCHA.....	31
Figure 5. Comparisons between STOCHA and VARYN.....	39
Figure 6. Response surface corresponding to the objective function.....	47
Figure 7. Changing population size.....	48
Figure 8. Representative plot of performance of hybrid algorithm applied in this objective function.....	56

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratefulness to people who helped me on my research and thesis.

First of all, I want to thank Professor Gemmill for his great patience, inspiration and support throughout this research and writing of this thesis. With his wisdom, encouragement and guidance, everything becomes solvable. He has shown me what the correct way of doing research, to become professional and to prepare to be a future engineer is. He is always understanding in my graduate study.

I would give my gratitude to Professor Dorman for her help and the teaching of stochastic processes course, which help built a foundation for my research and graduate study.

I would also like to thank Professor Jackman for serving on my POS committee and being supportive.

Finally, I would like to express my great thanks to my parents, for whom I have great love and respect.

ABSTRACT

In this thesis we examine the performance of simulated annealing (SA) on various response surfaces. The main goals of the study are to evaluate the effectiveness of SA for stochastic optimization, develop modifications to SA in an attempt to improve its performance, and to evaluate whether artificially adding noise to a deterministic response surface might improve the performance of SA. SA is applied to several different response surfaces with different levels of complexity. We first experiment with two basic approaches of computing the performance measure for stochastic surfaces, constant sample size and variable sample size. We found that the constant sample size performed best. At the same time we also show that artificially adding noise may improve the performance of SA on more complex deterministic response surfaces. We develop a hybrid version of SA in which the genetic algorithm is embedded within SA. The effectiveness of the hybrid approach is not conclusive and needs further investigation. Finally, we conclude with a brief discussion on the strengths and weaknesses of the proposed method and an outline of future directions.

CHAPTER 1 INTRODUCTION

1.1 Introduction

The Simulated Annealing (SA) algorithm stands in contrast to other heuristic methods that base the acceptance of new solutions during the iterative search process solely on improvement or when a decrease in the cost function is achieved. Simulated Annealing will accept an increase in the cost function with some probability based on the annealing algorithm. The probability of accepting increases in the cost function slowly decreases to zero as the temperature declines. Simulated annealing is based on an analogy to a physical system which is first melted and then cooled or annealed into a low energy state. Although Simulated Annealing is intended for optimization of deterministic problems, it has also been applied to stochastic optimization problems. However, there is a lack of practical information about the performance of SA when applied to stochastic problems. In this thesis the results of a number of empirical studies regarding the performance of SA on stochastic optimization problems will be presented. The experiments allowed the authors to make some initial evaluation of the general performance of SA to this type of problem and also to examine the influence of tuning some of the main parameters of SA. The motivation for the study was the desire to learn more about the effect of varying amount of response surface noise on the performance of SA. Also, we were able to evaluate whether artificially adding noise to a deterministic response surface might improve the performance of SA for deterministic optimization problems.

The thesis addresses simulation optimization in the context of design problems where performance measures or objective functions are stochastic, and examines what makes it different from the corresponding deterministic problem. For the initial empirical studies in the thesis we chose to use a very straightforward approach. The basic SA algorithm is employed and the measure of performance of a candidate move in the neighborhood of a current solution is determined by simply taking a sample average. For each of the sample objective functions selected for testing, first the problem of optimizing the function over some range of feasible parameter values when the performance measure is deterministic is considered. Then, we analyze the behavior of the simulated annealing algorithm on the same functions when their performance measures are artificially changed to stochastic problems. We also discuss how the level of noise and sample size used affects the performance of simulated annealing. When dealing with problems with noise we are interested in collecting many alternatives that are nearly equally good. One of the contributions of the research is giving a clear statement about the effect of the sample size chosen or varying the sample size on convergence. In this thesis, we also propose a hybrid algorithm that combines simulated annealing and the genetic algorithm to achieve an improved performance. The hybrid algorithm relies on the generation of offspring by embedding parenting. The metropolis procedure is performed after crossover and mutation. In our proposed algorithm, the offspring chosen as candidates for the annealing process are better, thus convergence to a good solution may be improved.

1.2 Literature review

The simulated annealing (SA) optimization algorithm derives its name from the optimization process in the energy of a crystalline solid by removing defects in the atomic arrangement. Researchers have developed analytical and simulation models to gain insights into the connection between this thermodynamic behavior and local search algorithms. SA has been extensively applied to deterministic optimization problems and the theoretical basis of the algorithm for this application has been known for a number of years. Many instances of practical and difficult problems were successfully solved by simulated annealing. The effectiveness of SA is attributed to the nature that it can explore the design space by means of a neighborhood structure and escape from local minima by probabilistically allowing uphill moves controlled by a temperature parameter.

Kirkpatrick realized the similarity between the optimization of combinatorial optimization problems and the physical process of annealing (Kirkpatrick et al. 1983). Simulated Annealing (SA) became one of the more popular optimization algorithms. Most survey articles address the issue of the theoretical application of SA to deterministic optimization problems. The popularity of SA has inspired many questions regarding the values of parameters that control the algorithm such as how the choice of temperature, cooling schedule, and neighborhood affect the convergence of this hill-climbing algorithm. Sullivan and Jacobson (2001) studied generalized hill climbing algorithms and their performance. They extended necessary and sufficient convergence conditions for simulated annealing. Azizi and Zolfaghari (2004) addressed changes in temperature based on the number of

consecutive moves showing improvement by comparing two variations of the SA method in adaptive temperature control. Rosen and Harmonosky (2005) proposed a simulated annealing-based simulation optimization method, which is an asynchronous, team-type heuristic. It improved the performance of simulated annealing for discrete variable simulation optimization. With the conventional cooling schedule, the probability of transition decreases from the beginning of the search to the end. Ameer (2004) found a simple algorithm to compute the temperature in SA which is compatible with a given acceptance ratio of bad moves. He also provided a convex function for low temperatures and a concave function for high temperatures based on a geometric cooling schedule.

During the past two decades, the simulated annealing algorithm has gained wide attention in both theoretical and engineering fields, applied most frequently to deterministic problems. The original simulated annealing algorithm assumes that the objective function values can be evaluated exactly. However, in many practical problems, the evaluation of the objective function values may include some noise. In real life, the source of noise could come from measurement errors, or from the distinction between historic data versus the future data. For some cases the value of the objective functions are accurate only to a certain tolerance. The first theoretical analysis of simulated annealing applied to solve discrete stochastic optimization problems was given by Gelfand and Mitter (1989). They showed that if the noise in the estimated objective function values in iteration has the normal distribution with zero mean and positive variance, then their procedure converges in probability to the set of global optimal solutions provided that the sequence is chosen properly. In 1996, Gutjahr and

Pflug (1996) generalized a classical convergence result for the Simulated Annealing algorithm to the case where cost function observations are disturbed by random noise. They also showed that it converges in probability provided that the variance of normally distributed errors decreases at a rate that is significantly faster than the cooling schedule. Moreover, they extended their analysis to errors that are more peaked around zero than normally distributed errors.

Saul B. Gelfand and Sanjoy K. Mitter (1988) examined the effect of using noisy (i.e. with random error) or imprecise measurements (i.e. with deterministic error) of the energy differences on tracking the minimum energy state visited by the modified algorithms. When implementing the annealing algorithm, it keeps track of the minimum energy state visited by the annealing chain up to the current time. For the modified algorithms, noisy or imprecise measurements of the energy differences are used to select the successive states. They also showed that under suitable conditions on noise the modified SA still exhibits global convergence with probability. Charon and Hudry (1993) suggested adding noise to the SA algorithm, or the noising method. Their approach adds random noise initially and then gradually reduces the noise to zero in order to perturb the solution space. Gutjahr and Pflug (1996) demonstrated the classic convergence results for SA on the case where cost function observations are distributed by random noise. However, the computational effort of SA for a practical problem is often limited, that is, the convergence condition may not be satisfied. Aiming at the non-deterministic property of stochastic optimization problem, Mahmoud H. Alrefaei and Sigrin Andraddtir (1999) present a modified simulated annealing algorithm

designed for solving discrete stochastic optimization problems. The method is able to find global optimal solutions to discrete stochastic optimization problems with the hill climbing feature. The optimal solution is estimated by the different states in terms of the number of visits, or by the state with the best average estimated objective function value. However, the method differs from the other simulated annealing algorithms in that it uses a constant (rather than decreasing) temperature. They also discussed how both variants can be applied in discrete deterministic optimization problems when the objective function values are estimated using either transient or steady-state simulation.

Currently, the research on developing effective and robust algorithms applied on stochastic optimization problems has become a hot topic in international academic field. Since stochastic optimization problems include uncertainty and noise, and sometimes the objective function is not explicitly known, performance estimation for solutions can only be obtained by multiple evaluations or simulations. It is especially difficult to reach global optima when the search space is large with multiple local optima. In 2001 Charon and Hudry (2001) extended their noising method. The algorithm perturbs the solution space by adding random noise to the problem's objective function values. A stopping criterion is introduced in a precise way that gradually reduces the noise-rate. Prudius and Andradottir (2005) proposed two cooling schedule approaches for controlling the probability of moving to seemingly inferior points and used the state with the highest estimated objective function value obtained from all the previous observations. The difference between the two variants of the SA algorithm is the way of estimating the objective function values at the current and candidate

solutions in each iteration. Ling Wang and Liang Zhang (2005) proposed SA combined with hypothesis testing for stochastic discrete optimization problems, and demonstrated the effectiveness of the proposed approach by the simulation results based on stochastic numerical optimization problems. Meanwhile, the effects of simulated annealing and noise magnitude on the search performance have also been studied.

1.3 Thesis outline

The remainder of this thesis is organized as follows. In Chapter 2, we propose the simulated algorithms. Chapter 3 illustrates the procedure for finding the parameter set of the cooling schedule for SA and discusses some other properties of the proposed algorithms. Chapter 4 describes the performance of simulated annealing in deterministic problems, metropolis procedure with noise, and stochastic problem with varying sample size. Many numerical experiments are reported in this Chapter. Chapter 5 presents a theoretical complexity comparison included with comparison of hybrid algorithm. Finally, some concluding remarks and possible future work are given in Chapter 6.

1.4 Glossary

<i>Accept</i>	A move that changes the current state
<i>Annealing</i>	The thermal process for obtaining low-energy states of a solid
<i>Cost</i>	The value of the objective function
<i>Downhill</i>	In a minimization problem, when the change in cost of an accepted move is negative, the overall cost improves

<i>Equilibrium</i>	Indifference to previous states in a process, such that an independent sample can be drawn
<i>Hill-climbing</i>	Deteriorating moves and lateral moves are accepted with some probability
<i>Landscape</i>	The solution space, where the <i>height</i> is the overall cost at a single point or state
<i>Neighborhood</i>	All the possible states that could result after a single move
<i>State</i>	A single data point
<i>Temperature</i>	The control parameter in simulated annealing which determines the probability of accepting a bad move. When the value is larger the probability is greater
<i>Uphill</i>	As a minimization problem, when the change in cost of an accepted move is positive, the overall cost worsens

1.5 Notations

This subchapter introduces vocabulary and notation. The following terms provide a reference:

Table 1. Notations

SA	Optimization	Symbol
State / Configuration	Feasible solution	u_i
Cost value	Performance	z_i
State space / Configuration space	Domain of objective function	U
Energy	Objective function	$Z(x, y)$
Move class	Neighbor	$N(u_i)$

In addition to the table above, noise refers to fluctuations, perturbations or small errors. Energy is the cost function that has to be minimized. A state is a set of values for the decision variables that represent one feasible solution to an optimization problem. We indifferently use energy and cost to designate the same thing.

In this thesis, DETERM represents the SA algorithm applied to deterministic optimization; STOCHA represents the SA algorithm applied to stochastic optimization; VARYN represents stochastic problem with varying sample size.

CHAPTER 2 SIMULATED ANNEALING ALGORITHM

2.1 The SA algorithm applied to deterministic optimization (DETERM)

When using the simulated annealing algorithm, the first step is to define the solution space, or a range of feasible values for the decision variables which represent possible solutions for the optimization problem. Each solution has a specific cost value. The search starts from a randomly generated initial feasible solution. In order to choose potential moves to new solutions (new values for the decision variables) a neighborhood structure is defined in the vicinity of the initial or present solution. A potential move is then generated randomly from the neighborhood. To find a good solution we move from a feasible solution to one of its neighbors in accordance with a probabilistic criterion. The move is accepted if the value of the objective function decreases (for a minimization problem). Otherwise, the move is rejected only with some probability which depends on the difference of cost between the present and new solution and a controlling mechanism called the current temperature. It is this feature that prevents the method from becoming stuck in a local minimum. A cooling schedule is the method used to decrease the temperature which is one of the controlling parameters. As the temperature is decreased the probability of accepting a worse solution decreases. The performance of the SA algorithm strongly depends on the choice of the cooling schedule and the neighborhood structure. A small change in a combination of some variables can generate a neighboring solution with a different cost value. Ideas from ordinal optimization and statistical selection can be incorporated to help determine if a move should be made.

To describe the specific features of a simulated annealing algorithm, we need the following definitions and assumptions. This algorithm utilizes a cooling schedule with the constant decay parameter α that is chosen to suit the specific problem (where $0 < \alpha < 1.0$).

Suppose we wish to find a solution that minimizes the objective function Z . Solutions that satisfy the constraints are called feasible solutions. Let $u_i \in U$ where U is the solution space; z_i is the cost (performance) related to a configuration or solution u_i . A feasible solution with minimum cost value z^* is called an optimal solution u^* . i is simply an iteration counter. The number of iterations I needed to guarantee finding the global optimum is generally very large. T_k is the temperature at time or iteration k . A candidate solution u_{i+1} is generated by taking a random step in the neighborhood of u_i . SA then compares the cost z_i and z_{i+1} to determine whether or not to accept the move to u_{i+1} . A unique characteristic of SA is that it accepts not only better solutions but also worse solutions with some probability in order to escape local optimal points. The transition probability of accepting a worse configuration u_{i+1} is $e^{-(z_{i+1}-z_i)/T_k}$. In each iteration, the transition probability is compared with a uniform random number R on the range $[0, 1]$. If the transition probability value is greater than or equal to the random number R , then the transition to the worse solution is accepted. Otherwise it is rejected. If it is rejected, another solution in the neighborhood will be generated and evaluated. The duration of each temperature level determines the number of iterations at a certain temperature. The temperature decreases during the search according to

a function known as the cooling schedule. The procedure terminates when a specified lowest temperature has been reached.

The basic steps for SA when applied to a deterministic problem are as follows (assuming minimization):

Step 1: Select an initial feasible solution u_0 , Determine initial temperature $T_{initial}$.

$$T_k = T_{initial};$$

Step 2: Do while $T_k > T_{pause}$, where T_{pause} is the stopping temperature.

Set $i = 1$;

Step 3: For $i < I$ where I is the length of inside loop in which the temperature remains constant.

Step 4: Generate a new random solution u_{i+1} with cost z_{i+1} from the neighborhood of the current solution u_i with cost z_i .

Step 5: Calculate $\Delta z = z_{i+1} - z_i$

Step 6: If $\Delta z < 0$

Then goto Step 8

Else randomly generate $R = \text{Uniform}(0, 1)$

Step 7: If $R < e^{-(z_{i+1} - z_i)/T_k}$

Then goto Step 8

Else reject u_{i+1}

Set $i = i + 1$

Back to Step 3

Step 8: Accept u_{i+1} as present solution

Step 9: *Set $i = i + 1$ and goto Step 3*

Step 10: set $T_{k+1} = \alpha \cdot T_k$ and **goto Step 2**

2.2 The SA algorithm applied to stochastic optimization (STOCHA)

The above procedure describes the implementation of SA as applied to deterministic problems. It will now be extended to stochastic optimization. To achieve a good performance for stochastic optimization problems, 1) the evaluation for solutions should be reliable and not too time-consuming, 2) the SA cooling scheduling should be well designed, and 3) the number of repeated searches should be reduced to improve efficiency. Thus, we propose a modified SA algorithm for stochastic optimization. The strategy of choosing the initial temperature and neighborhood structure will be the same as for the deterministic approach, including the mechanism of accepting the candidate solutions and the transition probability. As before, each new configuration is randomly derived from the neighborhood of the old configuration. If the new configuration is better than the current configuration it is automatically accepted. Otherwise it is accepted with some probability using the same procedure as before.

To obtain the performance for a given configuration of the decision variables for a stochastic problem, Wang and Wu (1999) used an average value of the cost function generated from a

number sample problem instances. The mechanism implemented for stochastic SA in this thesis is very similar to what they addressed.

The modified algorithm is as follows:

Step 1 to Step 3 identical to “Step 1 to Step 3” of deterministic algorithm.

Steps 4: The cost z_{i+1} of the new configuration u_{i+1} is estimated by taking an average of n samples (similarly for the cost z_i of the current solution u_i), or

$$z_{i+1} = \frac{\sum_{j=1}^n z_{j,i+1}}{N}, \text{ where } z_{j,i+1} \text{ is the cost value of the sample } j, j = 0,1\dots N, \text{ and } N \text{ is a}$$

fixed sample size.

Step 5 to Step 10 identical to “Step 5 to Step 10” of deterministic algorithm.

Therefore, it is a very simple modification of the deterministic approach with the only change being the utilization of a number of samples to account for the noise in the response surface.

The sample size N remains constant throughout the SA process.

2.3 Stochastic problem with varying sample size (VARYN)

Despite the wide use of simulated annealing for solving deterministic optimization problems, until recently little effort has been spent on the use of SA for stochastic problems. Consequently little has been done regarding controlling the sample size to utilize for performance value evaluation in stochastic optimization problems. There are many ways to control the sample size. Andradottir (2005) advocated two simulated annealing algorithms

for noisy objective functions. In the first one, only observations obtained in the current iteration are used. The number of observations of the objective function values taken at the current and candidate solution is constant. The second method utilizes all observations obtained so far at these two points. The estimate of the objective function value is the average of all observations collected for the candidate move so far. In an attempt to improve the performance of SA a second method was developed in which the sample size utilized is varied. We decided to take a fairly simple approach where initially a single sample is taken, and then the sample size increases as the SA algorithm proceeds.

The modified algorithm is as follows:

Step 1: Identical to initial algorithm. $k = 0$;

Step 2: Do while $T_k > T_{pause}$, where T_{pause} is the stopping temperature.

$$k = k + 1;$$

$$\text{Sample Size } n(k) = 2 \cdot \left(\frac{k}{K I (N - 1)} + 1 \right); \quad \text{Where } K \text{ is the total number of}$$

iterations that the outside loop will be executed (the number of times the temperature is changed) during the SA process. The number of iterations for the inside loop I does not change, and N is a fixed sample size.

Step 3 to Step 10 unchanged.

CHAPTER 3 IMPLEMENTATION

To compare the performance of the above methods, all three methods have been coded in C++, and run on a PC with a 166 MHz Pentium CPU. Since we are interested in comparison of the three methods, we eliminated the effects of other factors by choosing the same neighborhood structure and cooling schedule for all methods. However, we will discuss parameter settings before we start to test three algorithms with five 2-dimensional functions named F1, F2, F3, F4 and F5. The functions were chosen from very simple response surfaces (F1, F3) to much more complex response surfaces (F4, F5). The performances of the algorithms are compared based on 200 independent replications. The figures representing the five response surfaces are provided in the Appendix.

We evaluated the computational performance under a variety of SA control parameter settings for the basic application of SA to deterministic response surfaces. The primary objective in our numerical studies was to study the effects of different combinations of the parameters (initial and final temperatures, cooling schedules, etc), and neighborhood structures to determine reasonable approaches. For application to stochastic response surfaces we also experimented with durations of the inside loop (the amount of time to remain at a given temperature). After determining the values for each of the SA parameters, we experimented with different levels of noise in the response surface in an attempt to measure the effect of this noise on the performance of the algorithm.

3.1 Discussion of parameters chosen- DETERM

When applied to a minimization problem, an optimal solution is a solution with the minimum possible cost for the feasible solution space of the problem. It is well known that SA is a good choice to efficiently find approximate solutions. Our interest lies in obtaining satisfactory answers to the questions such as how SA would perform on a given landscape with a given temperature decrement rule and a given neighborhood.

As previously stated, one of the most important properties of simulated annealing is its hill climbing feature, which is achieved by accepting some inferior moves. Consequently, the likelihood of accepting worse moves is very important in evaluating the ability of simulated annealing to escape from local minimum.

General problems associated with the implementation of SA include:

1. How to choose the initial/pause temperature?
2. What is the neighborhood structure to use?
3. Determining when the annealing process stops?
4. What are the interaction effects among parameters? How do they affect the performance of SA?

First of all, we will focus on the choice of the initial temperature and some other properties of the acceptance ratio. To allow simulated annealing to find good solutions, one has to

carefully select the initial temperature. A classical and intuitive method is described in Kirkpatrick et al. 1983. It consists of selecting a temperature such that the acceptance ratio is approximately equal to a given value χ_0 . For the objective functions chosen to be tested in this thesis, χ_0 is set to 0.9 at the Initial Temperature; χ_0 is set to 0.001 at the Pause Temperature. Using this type of rule, cycles are avoided and a good estimation of the temperature can be found for the corresponding objective function.

We chose $T_{initial}$ and T_{pause} such that the probability of accepting a “hill-climb” equal to one-tenth of the range of possible z values are equal 0.9 and 0.001 respectively. That is we solved

$P = e^{\frac{-\Phi}{T}}$ for T, where Φ is the range between the minimum and the maximum cost values and P is the probability of accepting a worse solution. (See Table2)

Table 2. Parameters setting

	F1	F2	F3	F4	F5
The actual value z	0.992405	3.90E-06	-6.40764	-0.375016	-1.49841
The range Φ	0.015195	50.3556	12.81206	0.758	2.92846
The bound of x,y	[-5,5]	[-5,5]	[-5,5]	[0,82]	[0,45]
Neighborhood factor	0.5	0.5	0.5	4	2.25
$T_{initial}$	0.0144	47.8	12.16	0.72	2.78
T_{pause}	0.00022	0.729	0.1854	0.011	0.0424
χ_0 Initial Probability	0.899855726	0.900012665	0.899998362	0.900074467	0.900018205
χ_0 Pause Probability	0.001000938	0.001000266	0.000997263	0.001016989	0.001001011

The study of different types of neighborhoods confirms that a good neighborhood is crucial for SA to find good solutions. The neighborhood of the local search is defined by a difference

of structures between the current solution and the solutions that can be reached by changing one or more elements in the current solution. The performance of SA substantially depends on the definition of the neighborhood and the move strategy. The neighborhood structure utilized in the thesis is:

$$u_{i+1} \sim \text{Uniform} [u_i - \sigma, u_i + \sigma] \quad \text{where } \sigma = \frac{1}{2} \cdot \text{Range} / 10.$$

Range is the difference between possible maximum and minimum values of the decision variables.

As proposed by Fleicher (1993), α controls the rate of temperature decay and the search time. The smaller the value of α , the faster the temperature declines. Choosing an appropriate value for α is linked to computation time, which also depends on the size and complexity of the function. Typically, the cooling schedule satisfies $T_{k+1} = \alpha \cdot T_k$ for all $k \in \mathbb{N}$; $0 < \alpha < 1$ (Rutenbar, 1989). Moreover, $\lim_{k \rightarrow \infty} T = 0$. When applying SA to the five objective functions, we utilized $\alpha = 0.95$ and $\alpha = 0.97$.

This acceptance probability strongly depends on the current temperature. When the temperature is relatively high, the acceptance probability is high. As the temperature decreases, so does the probability of accepting a neighbor that degrades the cost function. Thus, the SA search process when the temperature is low works similar to a simple local search. Finally, the algorithm terminates when some specified stopping criterion is met -- e.g., when the Temperature is reduced to T_{pause} .

During the application of SA, any time a move is not accepted, the previous point is restored. After a certain number of moves (inside loop), the temperature is decreased (outside loop) - thus decreasing the likelihood of accepting uphill moves. When no downhill (better) moves exist in the neighborhood, we have reached a local minimum and if at the same time the temperature is low, then we might get stuck in a valley from which we cannot escape. Hill-climbing, which provides the chance of moving to a worse point, gives simulated annealing the advantage over some other methods because of the possibility of escaping from a local valley in order to find global minima farther away. As the temperature decreases to T_{pause} , it is very unlikely that SA will allow a move to a worse solution therefore not unusual that you will become entrapped in a local minima u^* .

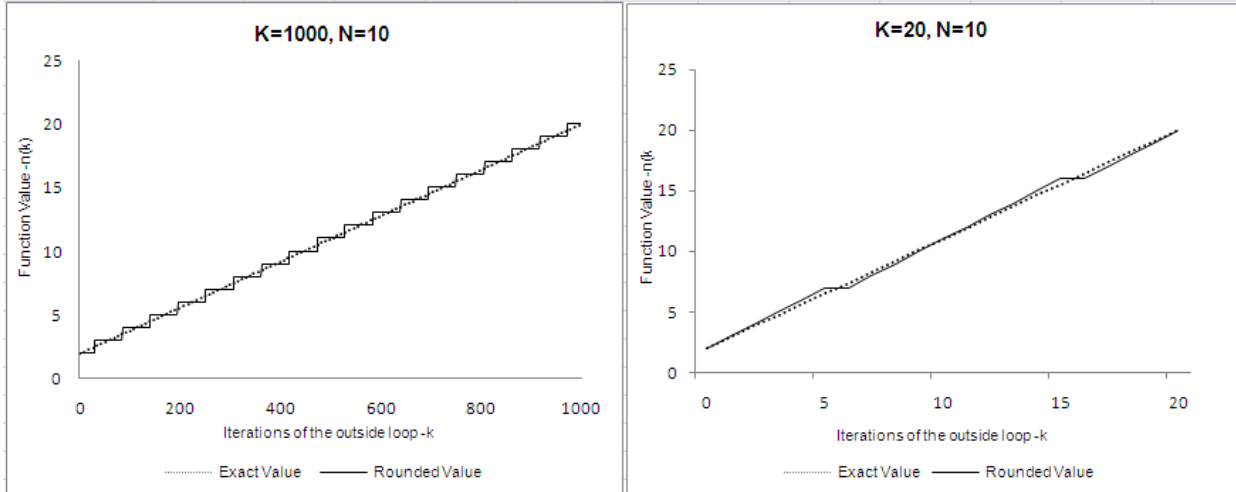
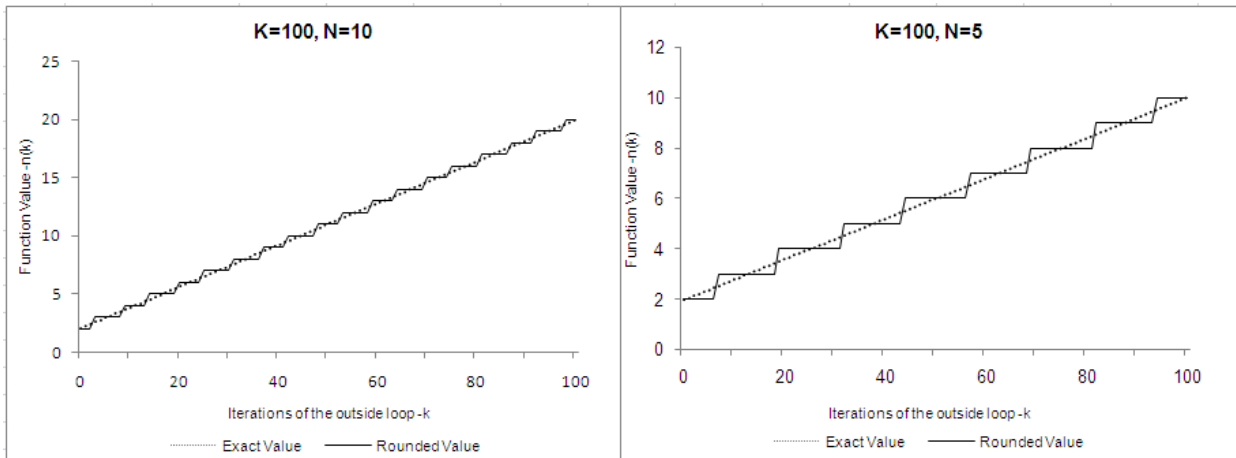
Generally speaking, the temperature is used to scale the differences in height of the landscape. Raising the temperature flattens a rugged landscape by providing a greater willingness to accept inferior points (climb a hill). However, small irregularities on a smooth landscape are accentuated by lowering the temperature and therefore most likely only accepting moves that improve the cost (downhill moves). We will elaborate on these ideas in the following chapter.

3.2 Parameters chosen - VARYN

When comparing a fixed sample size approach to our approach where a variable sample size mechanism is utilized, we designed our experiments so that the total number of samples generated during the process is basically a fixed amount. The number of times the outside

loop is executed (the temperature is changed) K and the fixed sample size N do affect the performance of VARYN Algorithm; therefore we examine the effects of changing K while keeping N as a constant value. (Figure1) We see that as K increases, the values of $n(k)$ have smaller increments. A larger K value causes the algorithm to divide the constant sample size into a proportionally greater number of parts. This trend is reflected in the graphs. Also, at higher values of K , the likelihood of having the same rounded n value for a two consecutive k values increases significantly.

$$\text{Sample Size } n(k) = 2 \cdot \left(\frac{k}{K/(N-1)} + 1 \right);$$

Figure 1. Changing K while keeping N as a constant valueFigure 2. Changing N while keeping K as a constant value

CHAPTER 4 RESULTS AND COMPARISON

4.1 Deterministic problem results (DETERM)

To illustrate the results from the application of the SA methods given in the previous sections, extensive numerical experiments are carried out. SA parameter settings utilized are held constant for all the test problems. The choice of the initial parameters, cooling schedule, and neighborhood structure can have a significant impact on the performance of SA. Experimental comparisons on five different functions or response surfaces are performed. The search begins with a high temperature allowing a greater chance of accepting a “hill-climb” and thus moving out of local minima. As we see, there is a higher probability of acceptance in the beginning, thus facilitating exploration of the search space; as the temperature decreases, the probability is reduced. The value chosen for α determines the rate at which the temperature is lowered. The choice for neighborhood size is also important. If the neighborhood is too small it may limit the ability of the process to sufficiently investigate the decision space U and/or decrease the probability of reaching the minimum value in a reasonable time. On the other hand, if the neighborhood is too large, then the process essentially performs a random search throughout U . In this thesis, we set the neighborhood size such that the half-length of a neighborhood is equal to one tenth of the *Range* corresponding to the feasible design space for each objective function.

The cooling schedule can have a significant impact on the quality of the solution; the

possibility of accepting a new solution depends on the relationship between Δz and T_K which is $P = e^{-\frac{\Delta z}{T_K}}$. It means that the annealing process is able to make transitions to higher energy state with positive probability so as to escape from local minima. As T_K decreases, the probability of the current energy state making a transition to an inferior state tends to zero. At colder temperatures, large uphill moves are unlikely to be accepted. As a lower temperature contributes to a lower transition probability, the end of the search has less chance of escaping local minima while in the beginning, the search is more likely to move out of local minimum solutions. Classically, the probability to accept bad moves (moves with increase in terms of cost) is high at the beginning to allow the algorithm to escape from local minimum. Intuitively, if the temperature is decreased to zero at a suitably slow rate then the annealing process converges in an appropriate probabilistic sense to the minimum state. After experimenting with different temperature combinations on the five functions it was found that a choice of initial and pause temperatures such that the probability (χ_0) of accepting a “hill-climb” equal to one-tenth of the range of possible z values are equal 0.90 and 0.001 respectively worked relatively well.

We use “percent error” to evaluate the performance of SA in DETERM. The percent errors in the table are calculated as

$$\text{Percent Error} = \frac{|z' - z^*|}{z^*}$$

Where z' is the actual best cost value found with SA and z^* is the true optimal cost value. A smaller percent error means the relative best cost value is closer to the true optimal cost value,

which means SA performs better in this objective function. Based on 40 independent runs, the average results found when applying SA to the four deterministic response surfaces can be found in Table 3. We haven't considered F2 here because the cost value of F2 is extremely small.

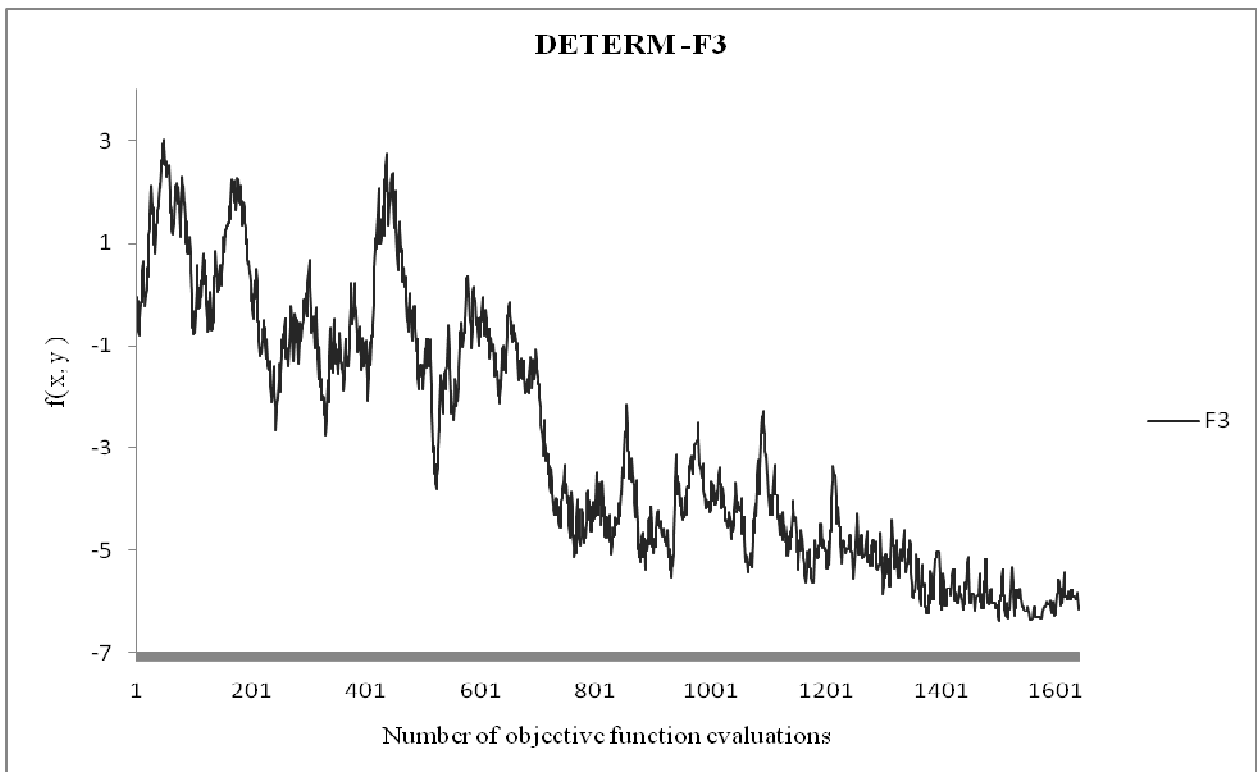
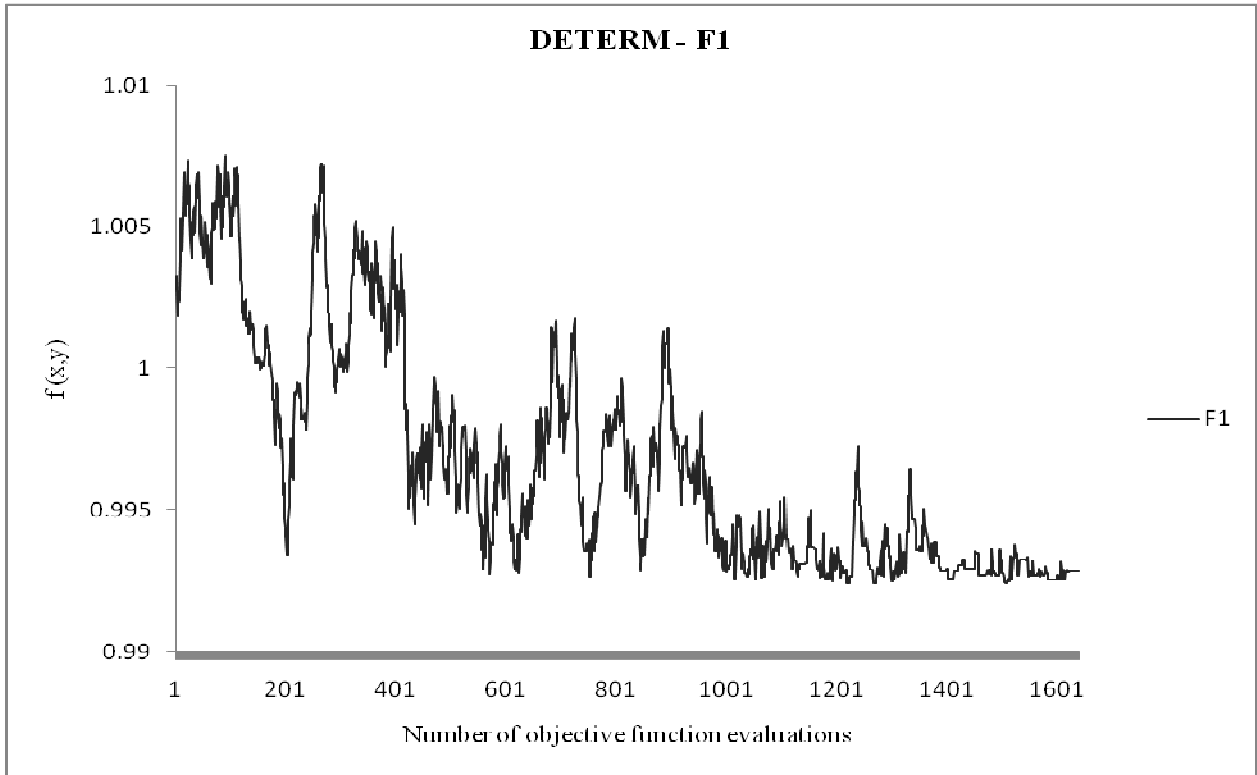
Table 3. Average percent error of the best cost found in DETERM

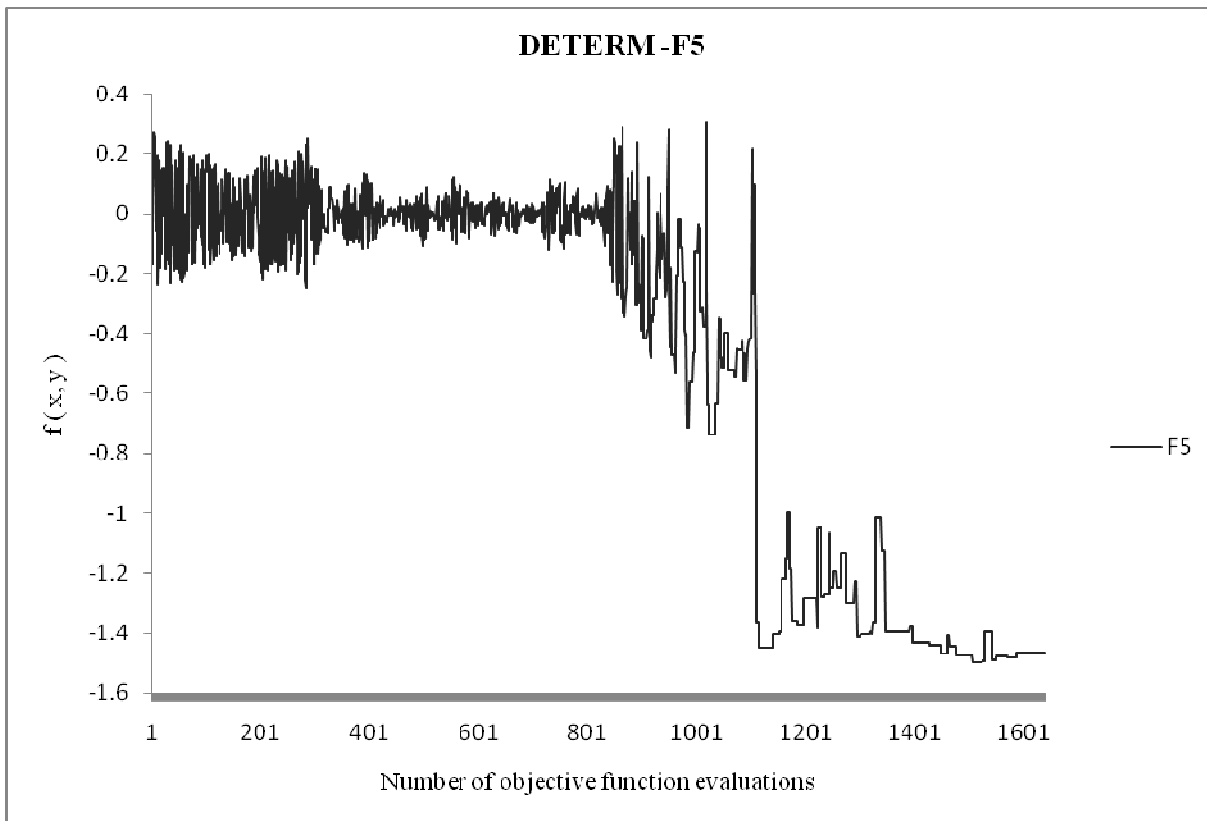
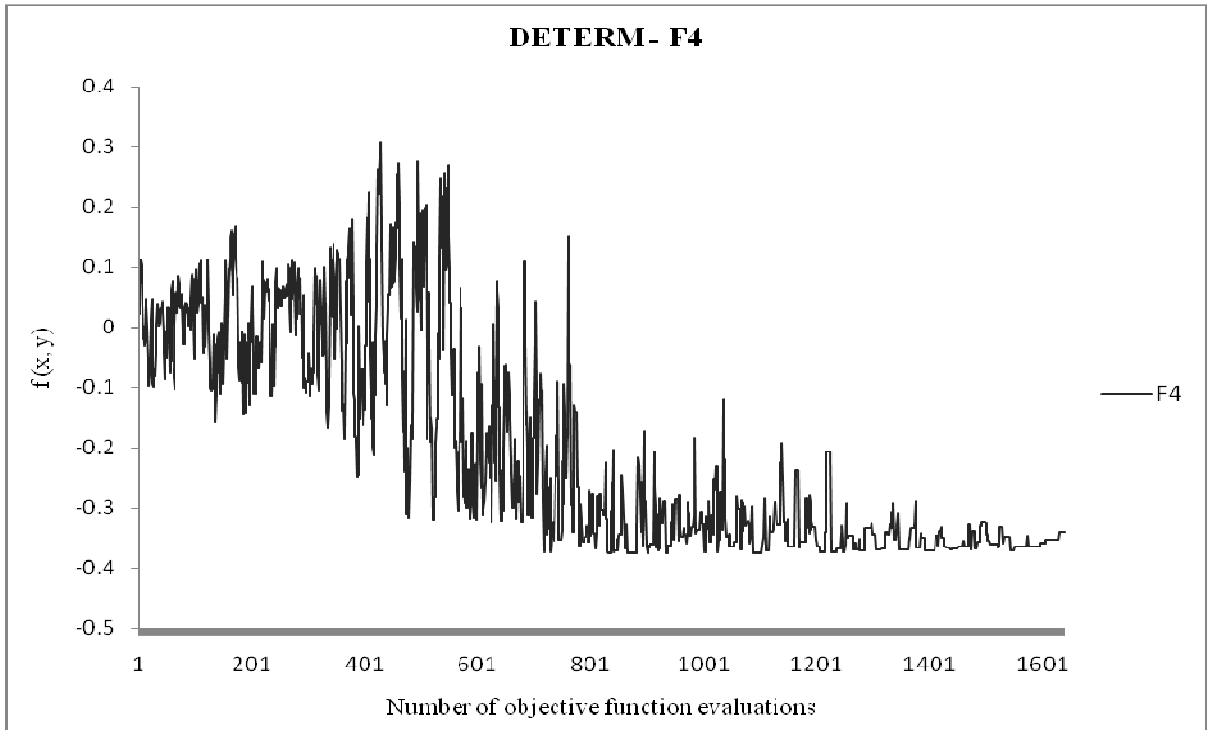
Objective Function (DETERM)	F1	F3	F4	F5
	0.0010%	0.4997%	7.2721%	16.7406%

As expected, the specific function itself has some influence on the performance of the annealing process. The response surface figures corresponding to objective functions can be found in the Appendix. For example, response surface F5 in which the response surface is very irregular resulted in an average percent error of 16.7406%. This is much higher than for the simpler functions or smoother response surfaces such as F1 and F3 where the percent errors are less than 1% respectively.

As would be expected, we found that the complexity of objective functions plays an important role in the performance of SA. The response surfaces of F4 and F5 are pretty rough, which means they have multiple local minima. Correspondently, the plot of the performance of SA for F4 fluctuates, although it converges in the end; the plot of the performance of SA for F5 jumps up and down in the beginning, then it goes down suddenly. On the contrary, SA performs very well in the objective functions with simple smooth response surfaces such as F1 and F3. (See Appendix: Response surface figures corresponding to objective functions).

Figure 3. Representative plots of performance of SA applied in F1, F3, F4, and F5





The following general conclusions can be made regarding the application to the deterministic response surfaces.

- The cooling schedule is important for efficient searches and good convergence characteristics.
- Proper selection of the neighborhood structure is important to the performance of SA.
- The mechanism of SA is based on a neighborhood search in which a probability function determines the transition from one solution to another. The magnitude of this probability depends in part on a temperature parameter that declines according to a cooling schedule.
- Parameters set depend not only on the particular problem being solved but also on the particular instance being solved.
- The temperature and the cost values should be of the same order of magnitude.
- The complexity of the function is an important factor affecting the performance of SA.

4.2 Stochastic problem results (STOCHA)

The main purposes of the thesis are to examine the performance of SA when applied to stochastic response surfaces and to evaluate whether artificially adding noise to a deterministic response surface might improve the performance of SA for deterministic optimization problems. We began with five deterministic response surfaces for which we could determine the true optimal performance. Random noise was then artificially added to these same surfaces and again the SA algorithm was applied to examine its performance given that the true optimal value is known.

The method used to add noise to the response surfaces is as follows. Let $z_{j,i}$ be the j^{th} random sample generated from solution u_i at the i^{th} iteration; where $j \in n$, n is the fixed sample size. The random samples are generated from the current configuration randomly according to the uniform distribution.

$z_{j,i} \sim \text{Uniform} [z_i - \mu \cdot \Phi, z_i + \mu \cdot \Phi]$, where μ is a noise factor $0 < \mu \leq 1$ and Φ is the range between the minimum and maximum cost values.

The performance z_{i+1} of u_{i+1} utilized in the metropolis procedure is simply

$$z_{i+1} = \frac{\sum_{j=1}^n z_{j,i+1}}{n}$$

Aimed at the uncertainty in stochastic optimization problems, multiple independent evaluations will be used to provide reasonable performance estimation for solutions. Noise factors μ of 0.025, 0.05, 0.10 and 0.15 were utilized combined with sample sizes n of 1, 5, 10 and 20 and randomly generated starting solutions. As expected, a simple examination of the plots of the results from making these various runs show that the performance of SA tends to improve as the noise decreases or the sample size increases. After making these preliminary runs we chose the following experimental design. We let the noise factor μ be 0.15 and 0.025; the inside loop $I = 50$ or 20; and α is fixed to be 0.97. So there are 4 groups of parameter combinations (noise, inside loop) utilized: (0.15, 50), (0.15, 20), (0.025, 50) and (0.025, 20). For each of the four functions 10 independent runs (different initial solutions) were made for

each possible combination or a total of 40 runs for each function. The result is illustrated in Table 4.

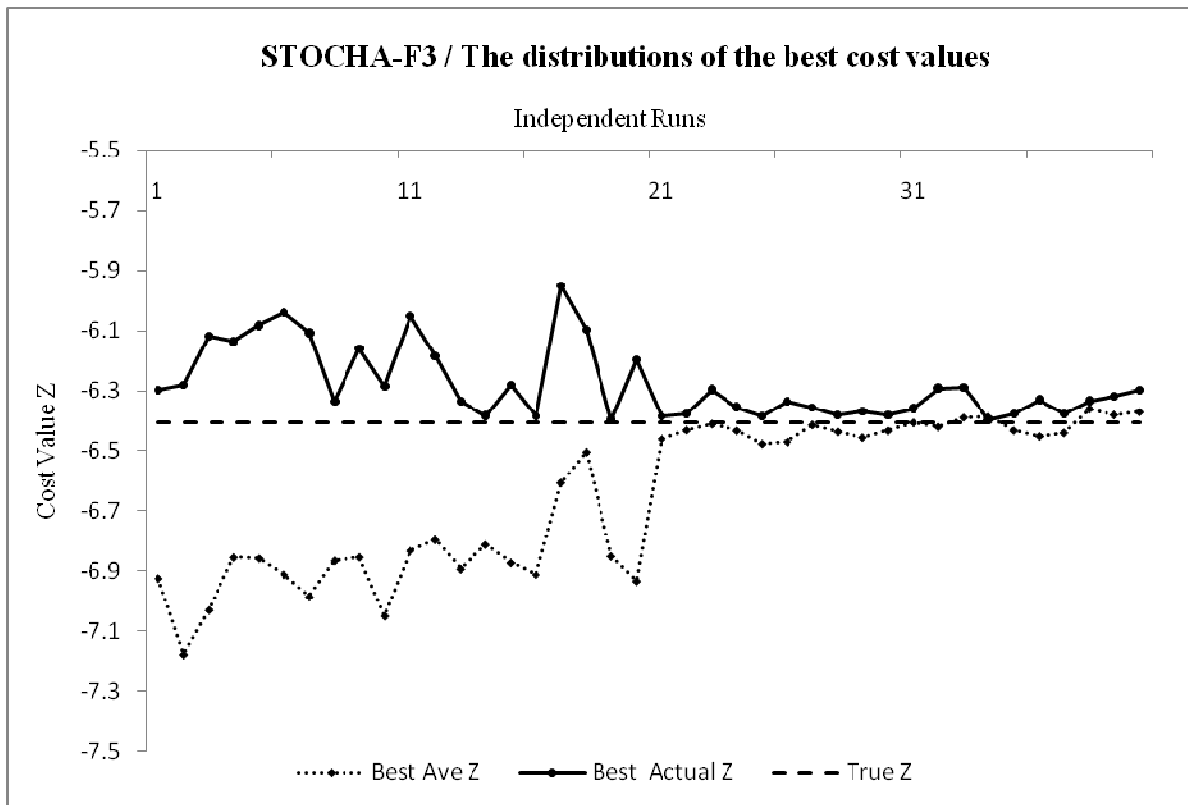
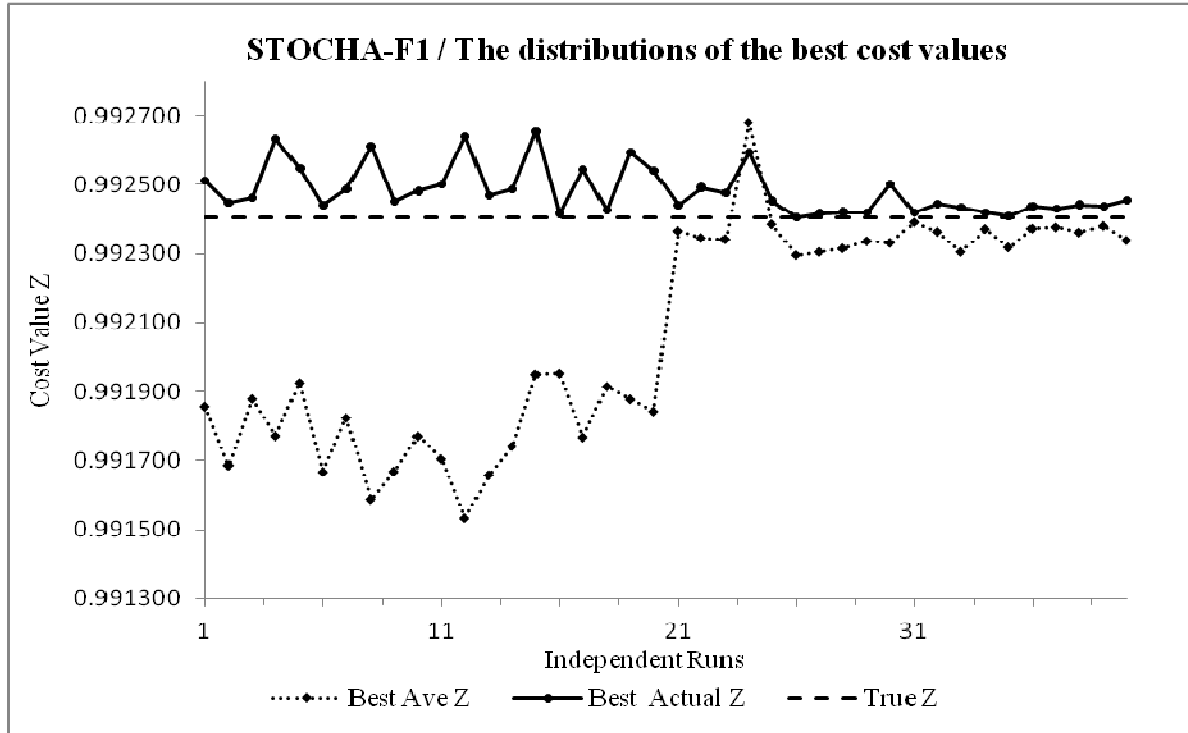
Table 4. The average best value found for each objective function over 40 independent runs.

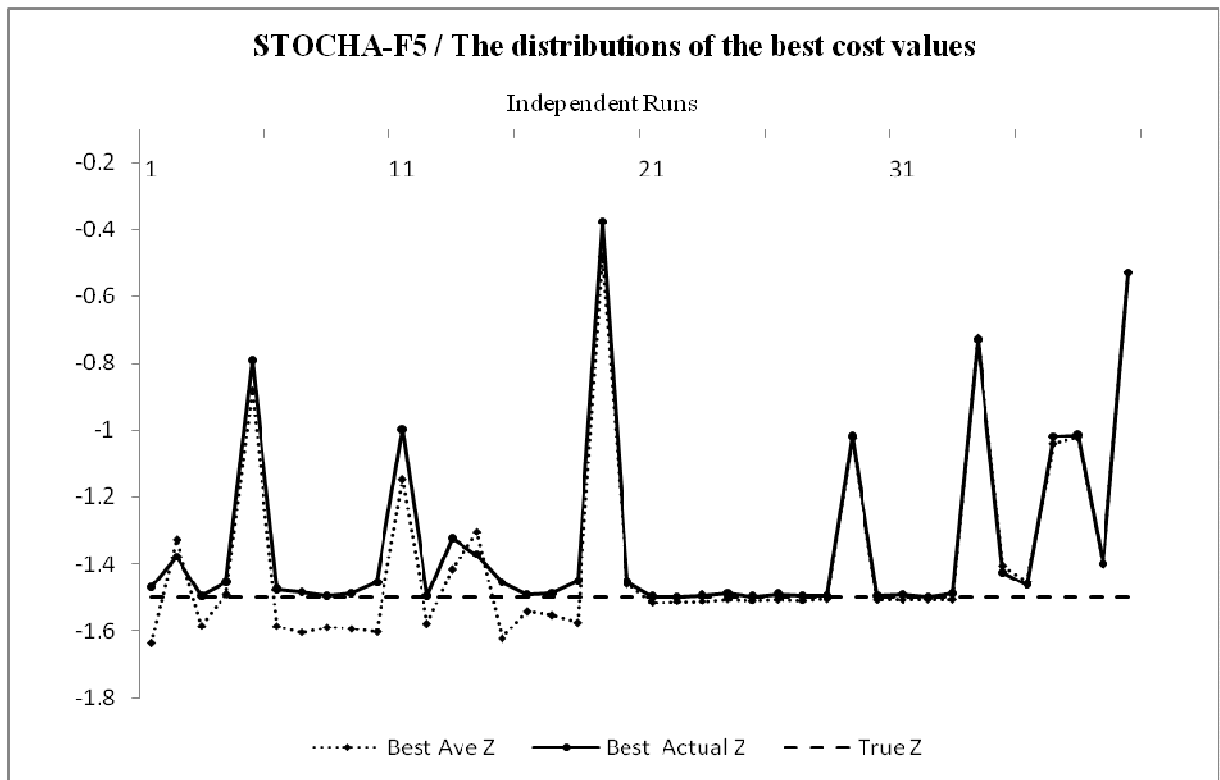
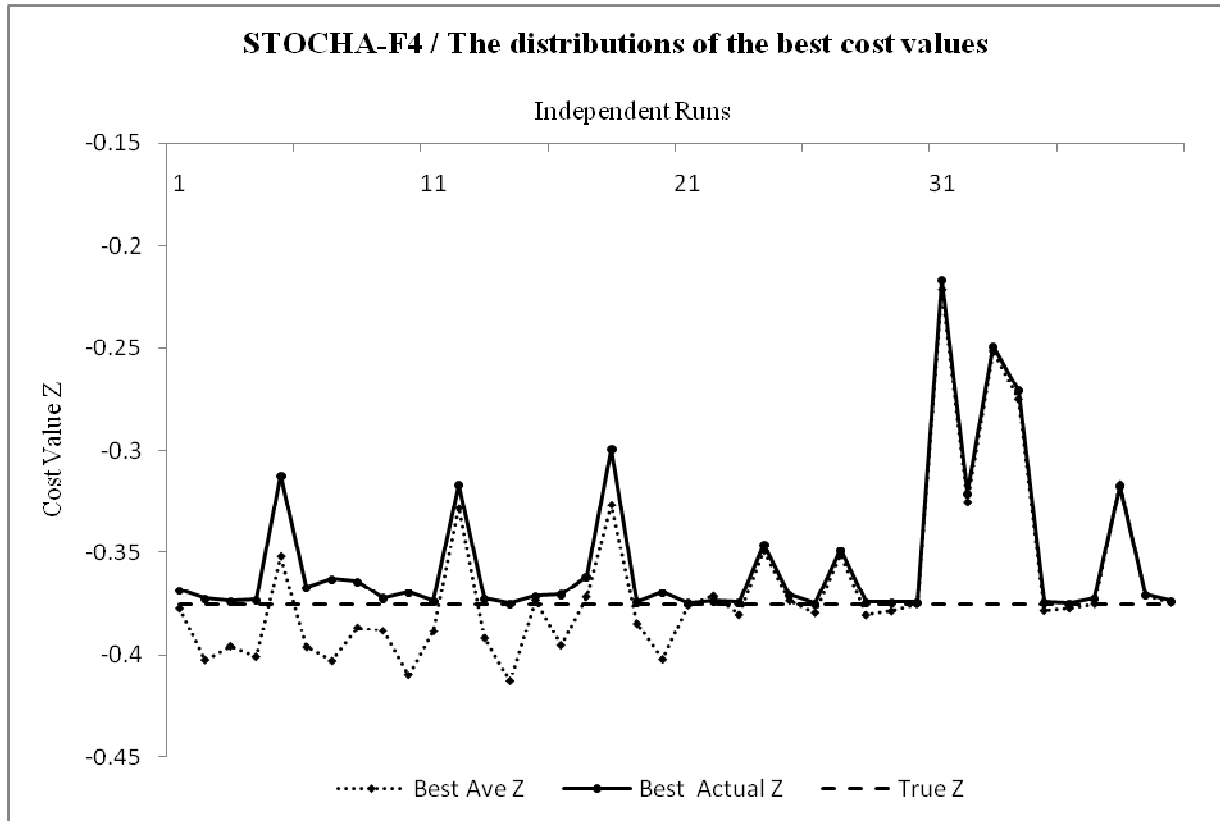
F1		F2		F3		F4		F5	
Ave Z	Actual Z	Ave Z	Actual Z	Ave Z	Actual Z	Ave Z	Actual Z	Ave Z	Actual Z
0.99207	0.992481	-1.57715	0.14799	6.649201	-6.2781	0.36684	-0.35376	-1.38096	-1.33546

Setting the parameters as Table 2 described in Chapter 3, we test the performance of SA in STOCHA with noise magnitude 0.025/0.15 and Inside Loop 20/50. Using 10 independent evaluations for performance estimation in each combination, the distributions of 40 independent runs in STOCHA is illustrated in Figure4.

Generally speaking, the plot of best average Z usually is under the true Z line; however, the plot of best actual Z is above the true Z line. (See Figure4) On the other hand, as the noise decreases, best average Z and best actual Z tend to be closer to the true Z line. The results for every run were plotted and it was found that an I of 50 seemed to have better performance characteristics than an I of 20 in the objective function with simple response surface such as F1 and F3. (I is the parameter of inside loop)

Figure 4. The distributions of the best cost values in STOCHA





Again we used percent error to evaluate the performance of SA. The average results in 40 independent runs found when applying SA to the response surfaces can be found in Table 5. The second column “The best Ave” represents the percent error corresponding to the average best cost value of the noisy samples; the third column “The best Act” represents the percent error corresponding to the actual best cost value; “The last Ave” represents the percent error corresponding to the average last cost value of noisy samples; “The last Act” represents the percent error corresponding to the actual last cost value.

Table 5. Percent error comparisons in STOCHA- noise / Uniform Distribution

Variance for	The best Ave	The best Act	The last Ave	The last Act
F1	0.0351%	0.0077%	0.0377%	0.0402%
F3	3.8887%	2.0217%	5.8659%	7.9913%
F4	6.7005%	5.6680%	27.7352%	29.5970%
F5	11.7828%	10.8746%	28.3280%	29.2526%

The response surfaces become more complex as you move from F1 to F5. As the response surface becomes more complex the percent error begins to increase dramatically, ranging from 0.0077% to 10.8746% in the best Actual Z.

To evaluate the influence of the noise, we modified the STOCHA algorithm by changing the noise generated to a Normal distribution instead of a Uniform distribution.

Let $z_{j,i}$ be the j^{th} random sample generated from solution u_i at the i^{th} iteration; where $j \in n$,

n is the fixed sample size. $z_{j,i} \sim N [z_i, (\frac{1}{10} \cdot \Phi)^2]$, where Φ is the range between the

minimum and maximum cost values.

The performance z_{i+1} of u_{i+1} utilized in the metropolis procedure is simply

$$z_{i+1} = \frac{\sum_{j=1}^n z_{j,i+1}}{n}$$

Percent error results are displayed as Table 6. Compared to Table 5, most of the percent error values in Table 6 are higher, particularly when STOCHA is applied in F5.

Table 6. Percent error comparisons in STOCHA- noise / Normal Distribution

Variance for	The best Ave	The best Act	The last Ave	The last Act
F1	0.0752%	0.0200%	0.0371%	0.0405%
F3	7.3798%	2.9917%	4.0435%	4.0217%
F4	7.1933%	8.4630%	16.5406%	23.1950%
F5	19.2203%	20.1303%	49.0645%	53.7168%

One final measure of performance is to examine at what point in the process SA located the best solution. For example, if the best solution found occurs at the 6900th accepted point out of a total of 7000 accepted points, the percentage is 98.6%. A larger percentage would seem to indicate that the SA algorithm is doing a better job of converging at the end to a near optimal solution. The results are summarized in Table 7.

Table 7. The average percentage of 40 data for 5 functions

	F1	F2	F3	F4	F5
Ave	77.8917%	70.9149%	83.2852%	38.5623%	69.2913%

The average percentage of F4 is the lowest followed by F5. This indicates that SA-STOCHA

performs worse for complex objective functions than for simple objective functions because of their irregular, rough response surface.

The following general conclusions can be made regarding the application of the constant sample size algorithm to stochastic optimization:

- SA performs well when applied to fairly simple stochastic functions but the chance of being trapped in the local minima increases as the function complexity increases.
- The length of time spent at a given temperature (inside loop) can have an important effect on the performance of SA.
- SA seemed to perform better on more complicated stochastic response surfaces than it does on more complicated deterministic response surfaces.
- The average point at which SA found the best solution seems to indicate that the algorithm and/or the SA parameters need further modification in order to improve the convergence characteristics.

In the stochastic problems, SA is not deterministic and will produce different samples each time they are run, even on the same problem. This is because of the probabilistic nature of accepting uphill moves and choosing moves. In particular, there is no guarantee of getting the same answer on multiple runs or even of getting precisely the optimum answer in any annealing algorithm.

4.3 Variable sample size problem results (VARYN)

The same runs were made as for the constant sample size approach. In order to compare the variable sample size approach to the constant sample size approach, we wanted to keep the total number of samples taken approximately the same. The total number of samples taken depends on the total number of iterations M , where $M = I \cdot K$, and I is the length of the inside loop, K is the length of the outside loop. By utilizing the calculation shown earlier for the sample size while performing the inside loop

$$\text{Sample Size } n(k) = 2 \cdot \left(\frac{k}{K/(N-1)} + 1 \right); \text{ (Round-off)}$$

The expected number of samples taken is

$$\text{Sum}[n(k)] = 2 \cdot \left(\frac{\sum_{k=0}^K k}{K/(N-1)} + 1 \right)$$

$$\text{Ave}[n(k)] = \frac{2 \cdot \left(\frac{\sum_{k=0}^K k}{K/(N-1)} + 1 \right)}{(K+1)} = \frac{(K+1)(N-1)+1}{(K+1)} = N$$

And therefore $\text{Ave}[n(k)]$ is N which will result in the same number of total samples.

The average percent errors can be found in Table 8. Similar to previous results, as the response surface becomes more complex the percent error begins to increase dramatically, ranging from 0.0132% to 13.2768%. Table 9 shows the percent error of both the best solution found and also the percent error of the last iteration of the SA algorithm. There is a significant difference in most cases indicating that SA is not converging to an optimal

solution at the end.

Table 8. Percent error comparisons in VARYN

Variance for	The best Ave	The best Act	The last Ave	The last Act
F1	0.0331%	0.0132%	0.0305%	0.0357%
F3	3.3962%	2.2794%	4.7116%	6.0004%
F4	8.7857%	12.1820%	34.2097%	35.9689%
F5	12.1788%	13.2768%	36.9082%	36.9473%

Table 9 shows the comparisons of average percent errors corresponding to the two approaches for stochastic optimization. We found that the percent error for variable sample size is very similar to the percent error for fixed sample size for the simpler response surfaces but the variable sample size performed somewhat worse as the complexity increased (F4 and F5). We believe that this is a result of less stable performance at the beginning of the search process (early iterations where the sample size is quite small) when using the variable sample size approach.

Table 9. Comparisons of percent error between STOCHA and VARYN

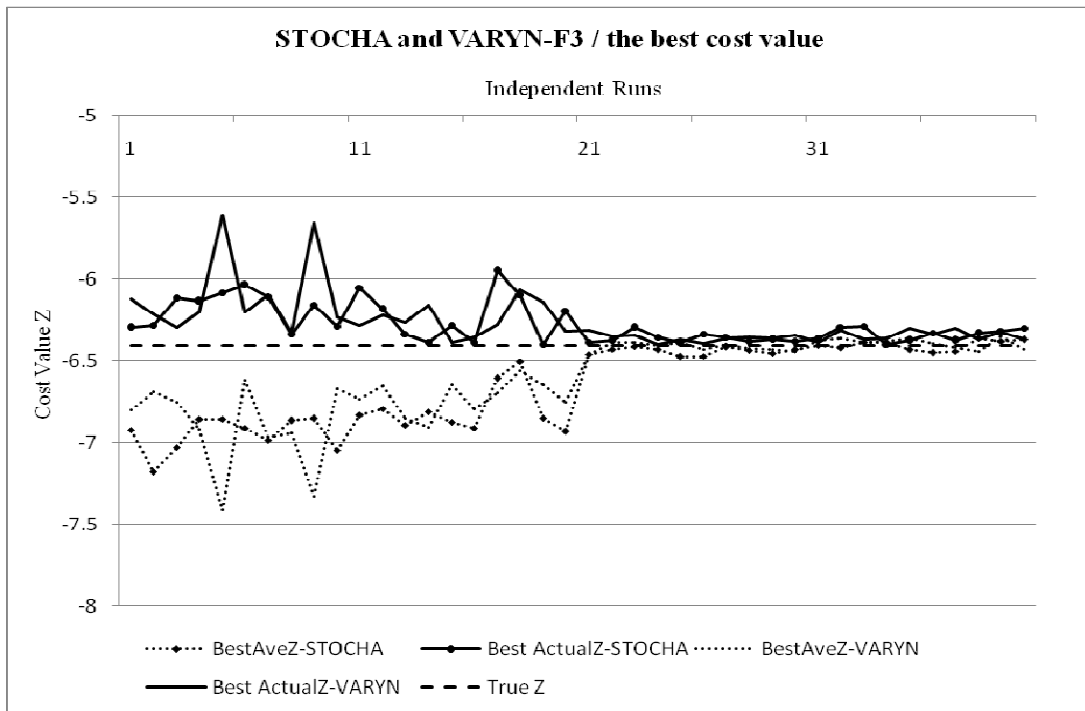
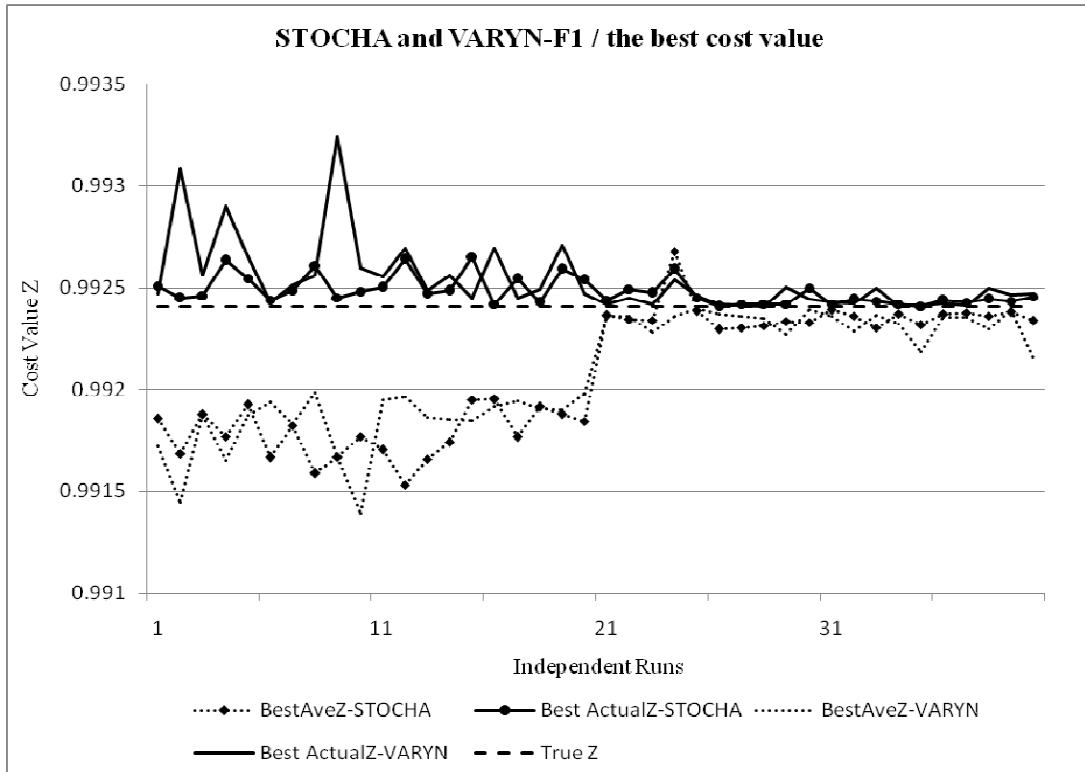
Variance for	The best Ave		The best Act	
	STOCHA	VARYN	STOCHA	VARYN
F1	0.0351%	0.0331%	0.0077%	0.0132%
F3	3.8887%	3.3962%	2.0217%	2.2794%
F4	6.7005%	8.7857%	5.6680%	12.1820%
F5	11.7828%	12.1788%	10.8746%	13.2768%

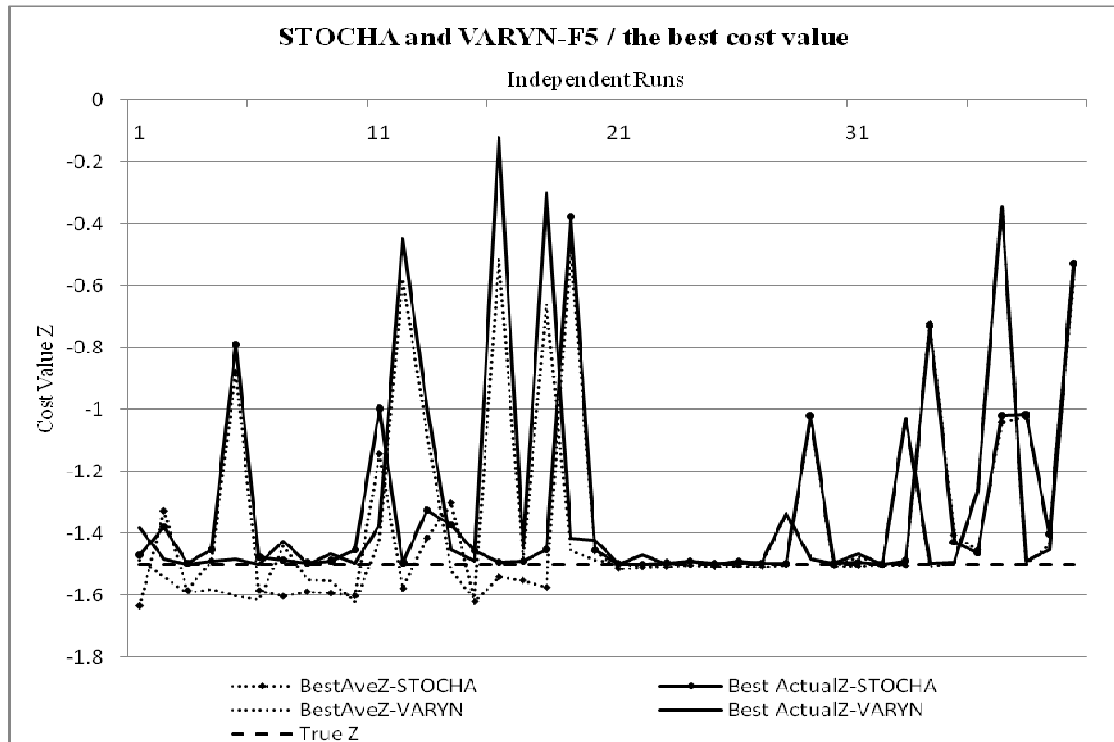
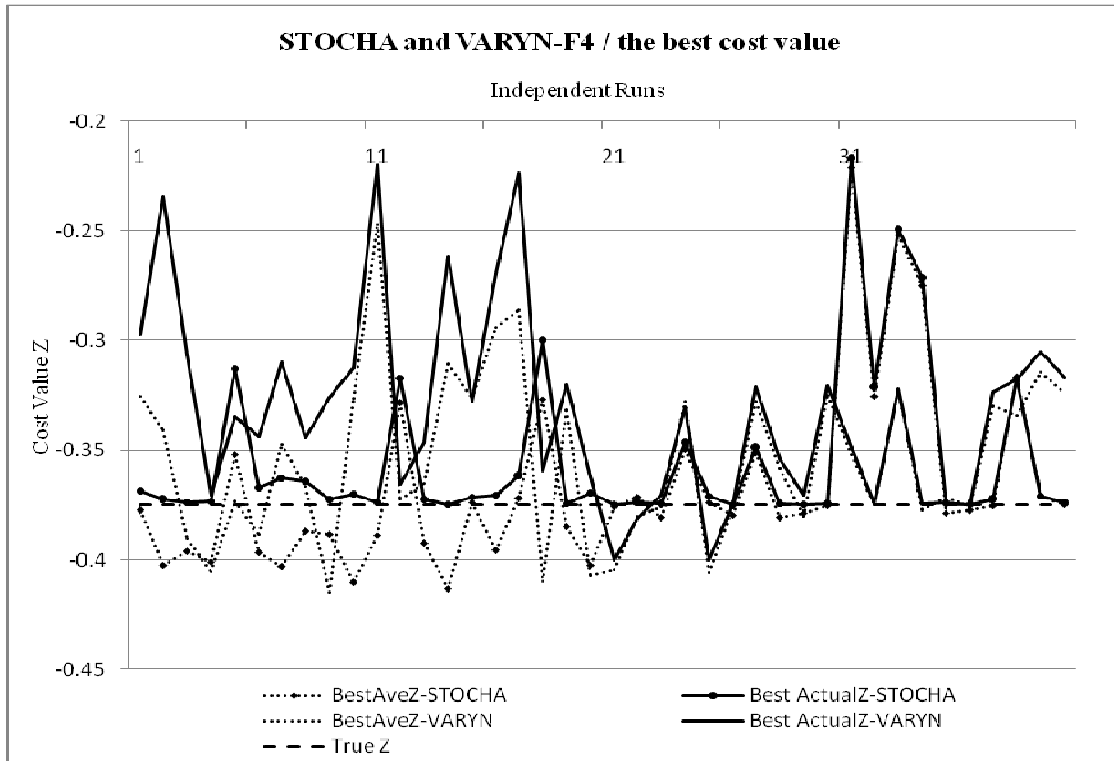
We use STOCHA algorithm and VARYN algorithm for solution performance estimation respectively. Using the same parameters, we test the performance of SA in VARYN with noise magnitude from 0.025 to 0.15 and Inside Loop is from 20 to 50. Using 10 independent

evaluations for performance estimation in each combination, the distributions of 40 independent runs both in STOCHA and VARYN with the above parameters are illustrated in Figure 5.

With regard to the objective functions with simple response surfaces, the distribution of the best cost values for VARYN is very similar to that for STOCHA. SA performs better with less noise or more inside iterations. Both BestAveZ-STOCHA line and BestAveZ-VARYN line are below the True Z line; Best ActualZ-STOCHA line and Best ActualZ-VARYN line are above the True Z line. Obviously, these four lines tend to converge to the True Z line as the noise decreases. The amplitude of fluctuation may differ in VARYN lines and STOCHA lines. However, objective functions with complex response surfaces don't match this pattern. Surrounding the central line True Z, both the BestAveZ line and the BestActualZ line jump up and down.

Figure 5. Comparisons between STOCHA and VARYN





The same general conclusions can be made regarding the application of the variable sample size approach as with the constant sample size approach with the following additions:

- The early smaller sample sizes (which creates more noise) appear to increase the chance of being captured in local minima early in the SA process. Smaller sample size means higher deviation of the noise, which makes the response surface highly noisy.
- The variable sample size algorithm becomes more conservative in terms of moving out of the local minimum due to the larger sample size taken at lower temperatures.

4.4 Overall comparison of results

It is interesting to note that for these particular functions SA did not perform as well on the stochastic response surfaces as it did for the deterministic response surfaces when the response surface was fairly simple, but actually performed better on the more complex response surfaces. This provides some evidence that artificially adding noise may improve the performance of SA on more complex deterministic response surfaces.

Table 10. Percent error of the best cost- Actual Z

	F1	F3	F4	F5
DETERM	0.0010%	0.4997%	7.2721%	16.7406%
STOCHA	0.0077%	2.0217%	5.6680%	10.8746%
VARYN	0.0132%	2.2794%	12.1820%	13.2768%

- SA performs better for simple functions than for complex functions regardless of the type of optimization being performed (Table 10).

- For simpler functions (F1 and F3), SA performs best in deterministic optimization, and worst for stochastic optimization with variable sample size.
- For those functions with more complex response surfaces (F4 and F5), SA performs the best in stochastic optimization with fixed sample size.
- Artificially adding noise may improve the performance of SA on more complex deterministic response surfaces.
- Since the landscape of a solution space may be hilly or smooth, how SA is cooled can have significant impact on the quality of the solution. If the cost function is really jagged and has really steep maxima or minima, the probability of SA finding them decreases significantly.

It is obvious from Table 10 that DETERM performs considerably better in those examples than STOCHA and VARYN. VARYN differs from STOCHA only in the way in which the noise samples of the solutions at the current and candidate solutions are obtained in each loop. In the varying sample size method, increase of sample size decreases the noise, while the latter method utilizes the constant sample size mechanism which tends to fix the size of the noise. A numerical example has been provided that documents the performance of our approaches. The reason why VARYN performs worse than STOCHA in the beginning of the search is that the smaller sample size results in bigger variance and hence, the objective functions are initially very noisy. But as the search progresses, VARYN becomes more conservative in terms of moving to worse points because better candidates become available and the temperature values are much lower.

Generally speaking, when SA is applied to some complicated function there is a possibility that the solution will not be globally optimal. However, the solution will usually be better than the standard local optimization algorithm. It should also be noted that SA might not always or, in some cases never, find the optimal solution to a given problem because SA does have a random property.

CHAPTER 5 HYBRID ALGORITHM

5.1 Hybrid algorithm introduction

Metaheuristics have gained considerable attention and have experienced remarkable growth over the past decade. Simulated annealing (SA) and genetic algorithms (GA) are among the most popular metaheuristics, and each of them has advantages and improvements. SA is a simple algorithm which takes a long time to reach the desired solution because of the annealing process. GA is based on natural genetic and evolutionary mechanisms that replicate the principles of population genetics, selection and inheritance. GA is more complicated but it takes less time than SA. The performance of these heuristics can be further improved if used in combination with an application-specific procedure. The work presented in this chapter has evolved out of the results of the experiments carried out employing SA and GA, which was aimed at improving the design of SA algorithms.

Table 11. Notations-Hybrid Algorithm

Hybrid Algorithm	Symbol
Current Population	U_k
A group of chromosomes in the k^{th} population	$u_{j,k}$
The fitness corresponding to one chromosome	$Fitness_{j,k}$
Initial Temperature	$T_{initial}$
Pause Temperature	T_{pause}
The length of inside loop	I

5.2 Hybrid algorithm implementation

In this section, we propose a hybrid method that appends the Genetic Algorithm to SA, state its performance and discuss the importance of this result. The proposed GA-SA method has the benefits of both GA and SA. This hybrid technique utilizes the advantages of SA to escape local minima and at the same time it improves the performance of the search by parenting, and performing crossover and mutation. Our approach consists of three phases. Initialization is the first phase, where the parameters of the algorithms are specified. By reducing the number of solutions in the search space, it starts from a relatively good candidate solution. Similar to GA, each solution is represented by binary and integer strings. Second, there is the actual search phase, that is, the usual iteration of the genetic algorithm selection, crossover, and mutation operators. In particular, we use a selection procedure to first quickly filter out inferior solutions and then determine the best solution by carrying out additional runs for the remaining solutions. The alternatives generated by the GA search are evaluated using ranking and selection. Finally, the search terminates and returns to the inside/outside iteration of SA. A pseudo-code for the Hybrid algorithm is presented below.

Step 1: Generate a random population U_k and a group of chromosomes $u_{j,k}$ with $Fitness_{j,k}$,

$$u_{j,k} \in U_k \quad (j \in J, J \text{ is the number of chromosomes})$$

Set $U_k = U_0$ where U_0 is an initial population

Step 2: Determine initial temperature $T_{initial}$.

Step 3: Do while $T_k > T_{pause}$, where T_{pause} is the stopping temperature.

Set $i = 1$;

Step 4: For $i < I$ where I is the length of inside loop in which the temperature remains constant.

Step 5: Generate children $u_{j,k+1}$ with $Fitness_{j,k+1}$ from parents $u_{j,k}$ by using three operators.

Step 5.1 Reproduction

Step 5.2 Crossover

Step 5.3 Mutation

Step 6: Compare children's fitness $Fitness_{j,k+1}$ to parents' fitness $Fitness_{j,k}$

$$\varepsilon = Fitness_{j,k+1} - Fitness_{j,k}$$

Step7: If ($\varepsilon \leq 0$) Or $\text{Random}(0,1) \leq e^{\frac{-\varepsilon}{T_k}}$

Then the new population U_{k+1} replace U_k

Else reject U_{k+1} ;

Step 8: Set $i = i + 1$ and goto Step 4

Step 9: set $T_{k+1} = \alpha \cdot T_k$ and goto Step 3

An excellent explanation of how reproduction, crossover and mutation are performed in genetic algorithms can be found in Goldberg (1989). We used the same approach in our implementation.

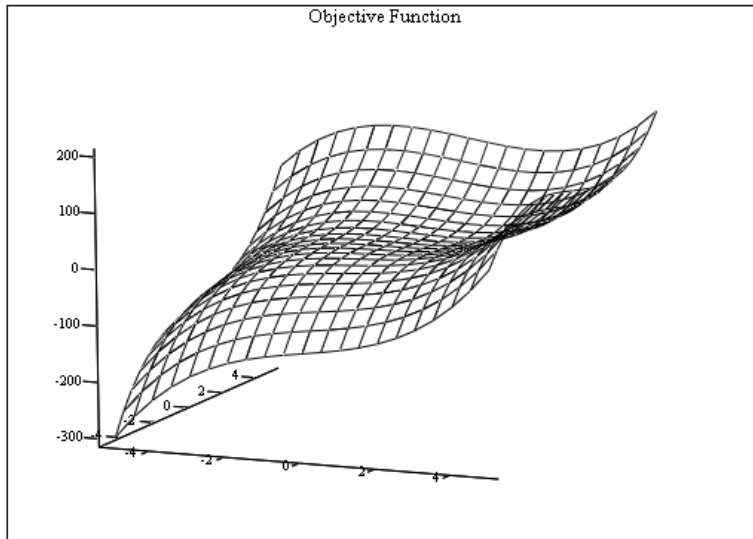
The main differences between the hybrid algorithm and regular SA algorithm are the following:

- Instead of the original neighborhood strategy in SA, the next solution is generated from the current population by implementing reproduction, parenting and mutation in hybrid algorithm.
- The acceptance of new chromosomes is determined by the annealing process and becomes a part of the next population.

5.3 Hybrid algorithm results

In this chapter, we use the genetic algorithm together with simulated annealing to develop a system for simulation optimization. The algorithm has been tested on a simple objective function with 2 variables. (See Figure 6)

Figure 6. Response surface corresponding to the objective function



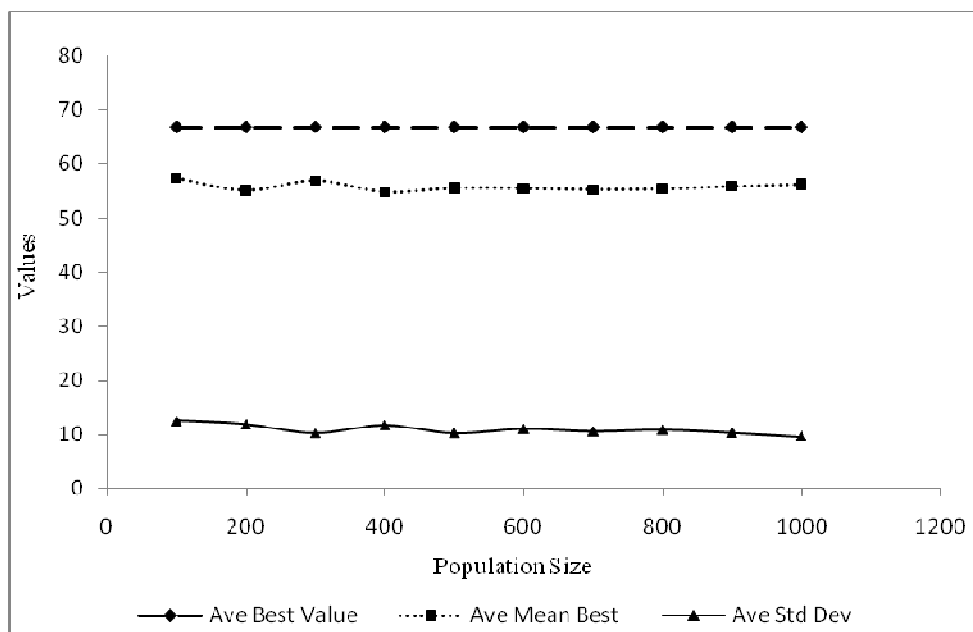
The results are averaged over 50 runs, where ‘Ave Mean Best’ stands for the average mean best function value obtained in the last generation and ‘Ave Std Ave’ indicates average standard deviation. First, we changed the population size while keeping other parameters as

fixed values. We found that Average standard deviation tends to decrease as Population Size goes up. (Table 12, Figure 7)

Table 12. Changing population size

Population Size	Ave Best Value	Ave Mean Best	Ave Std Dev
100	66.836	57.456	12.517
200	66.836	55.205	11.873
300	66.836	57.018	10.284
400	66.836	54.915	11.688
500	66.836	55.642	10.263
600	66.836	55.614	11.026
700	66.836	55.343	10.581
800	66.836	55.483	10.863
900	66.836	55.942	10.335
1000	66.836	56.314	9.625

Figure 7. Changing population size



Then, we changed the number of generations while keeping the population size at 100. The average mean tends to decrease as the number of generations increases. So as you increase the number of generations, the hybrid algorithm performs better, though the average best value doesn't change. (Table 13)

Table 13. Changing the number of generations

Generations	Ave Best Value	Ave Mean Best	Ave Std Dev
1000	66.836	57.456	12.517
2000	66.836	55.815	12.179
3000	66.836	55.252	9.563
4000	66.836	56.409	13.300
5000	66.836	56.261	11.242

Table 14 demonstrates that Ave Mean Best decreases as the probability of mutation increases from 0.1 to 0.25. The most interesting thing is Ave Std Dev is only 6.705 when the probability of mutation is 0.1, but it tends to increase to around 12 as the probability of mutation is greater than 0.15. This shows that the probability of mutation does affect the performance of the hybrid algorithm. Table 15 shows the results while changing the probability of crossover. The probability of crossover has less impact on the performance of the hybrid algorithm.

Table 14. Changing the probability of mutation

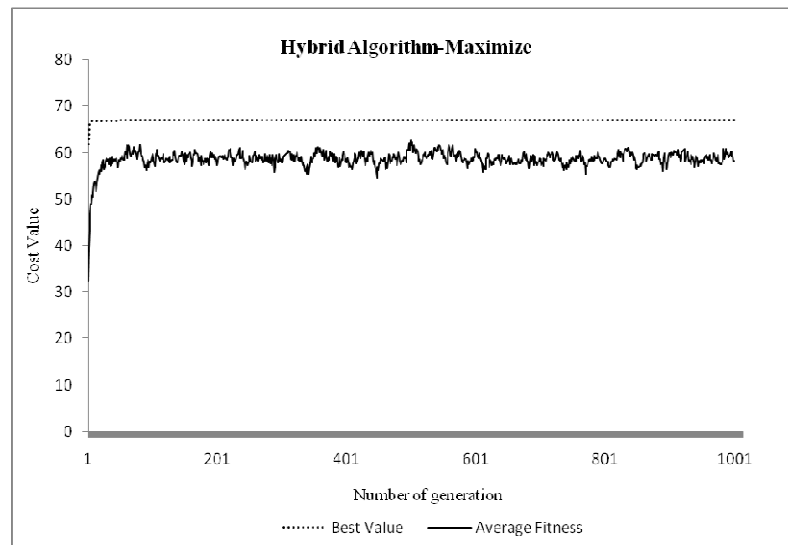
Probability of Mutation	Ave Best Value	Ave Mean Best	Ave Std Dev
0.1	66.836	57.978	6.705
0.15	66.836	55.815	12.179
0.2	66.836	54.332	11.756
0.25	66.836	53.477	12.198

Table 15. Changing the probability of crossover

Probability of Crossover	Ave Best Value	Ave Mean Best	Ave Std Dev
0.7	66.836	58.657	5.846
0.8	66.836	57.978	6.705
0.9	66.836	59.869	9.612
0.95	66.836	56.970	8.533

Compared to the performance of SA in deterministic problems, the hybrid algorithm reaches the optimum value more rapidly. However, the hybrid algorithm needs more memory to store the data, which means hybrid algorithm is more complicated than SA. The computational results give some indication of the effectiveness of the hybrid approach. However, the work presented here is still in its infancy and ongoing. Future work includes searching the effects of SA parameters on the solution quality to improve the solution and running time of the algorithm.

Figure 8. Representative plot of performance of hybrid algorithm applied in this objective function



CHAPTER 6 CONCLUSIONS AND FUTURE WORK

In this study the application of simulated annealing to stochastic optimization was examined. We proposed two adaptations to the standard simulated annealing in conjunction with a noise mechanism. In addition to a straightforward application in which a constant sample size was utilized, we also examined a modification in which the sample size varied. Sets of five different response surfaces of varying complexity were used to evaluate the performance of SA. The comparisons of the proposed methods and the conventional SA clearly demonstrate that the complexity of the cost function features does affect the performance of SA to a certain extent. The conclusion is supported by our statistical analysis. As would be expected the performance declined as the complexity of the response surface increased. In general the constant sample size approach outperformed the variable sample size approach. A somewhat surprising observation was that as the response surface complexity increased SA seemed to work better on stochastic response surfaces than on deterministic response surfaces. Therefore it appears that artificially adding noise may improve the performance of SA on more complex deterministic response surfaces.

Due to the fact that there is no general SA algorithm that works well for complex functions, there is still considerable room for further research. Most results that point in that direction are based on numerical evidence and further research direction may consist in introducing some modifications and studying the convergence of the algorithm in stochastic optimization problems. It will focus on how to best search for the optimal solution in the presence of noise.

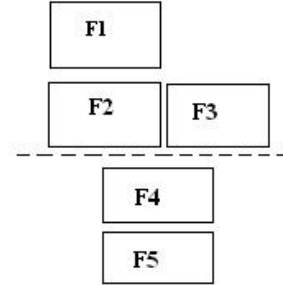
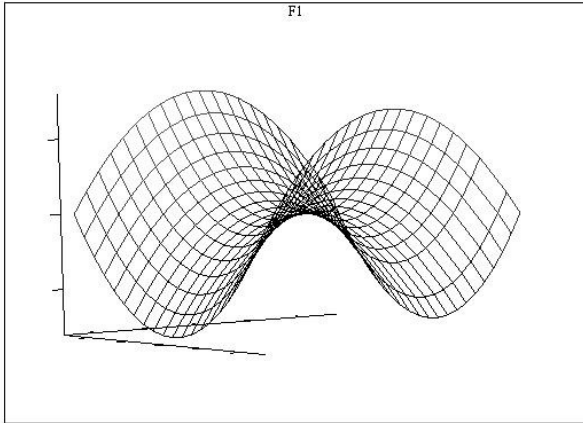
The algorithm allowing the selection of the initial temperature that is compatible with a given acceptance probability of all transitions is now under study. Investigating this balance is an important future research topic.

In addition to performing more tests on similar response surfaces, we aim at testing our algorithm with some other classes of problems such as n -dimensional cost functions ($n > 3$). Further work also should be devoted to additional modifications to the SA algorithm that might improve its convergence characteristics when applied to this type of optimization. As this research is carried out, it is important to take into account the concerns of computational efficiency. In order to demonstrate the performance of the hybrid algorithm we have considered only deterministic problems as a test bed. Another direction of potential interest may be the application of the hybrid algorithm to various problems. For example, extending the hybrid algorithm to a noisy environment is another research goal.

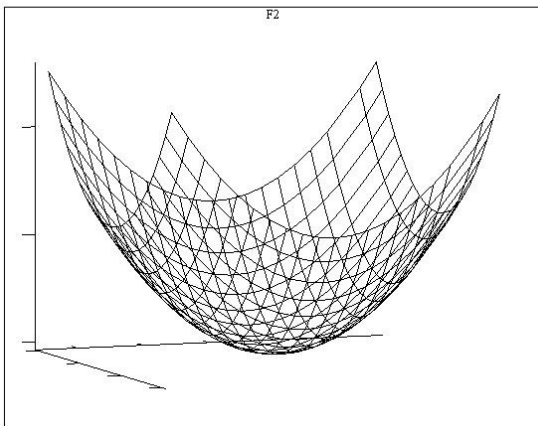
APPENDIX

Response surface figures corresponding to objective functions

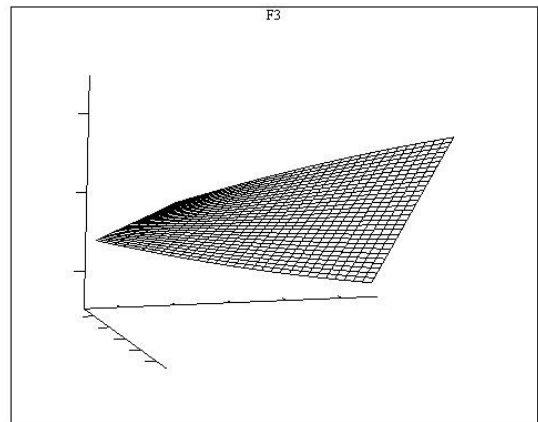
$$F(x, y) := \sin(\text{deg } x)^2 + \cos(\text{deg } y)^2$$



$$F(x, y) := \sin(\text{deg } x \cdot y) + x^2 + y^2$$



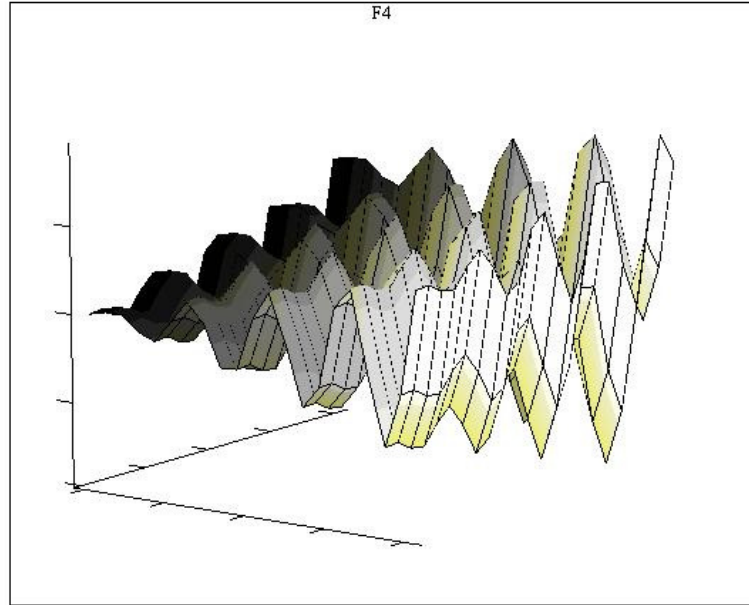
$$F(x, y) := \sin(\text{deg } 4x) \cdot y + \cos(\text{deg } 4y) \cdot x$$



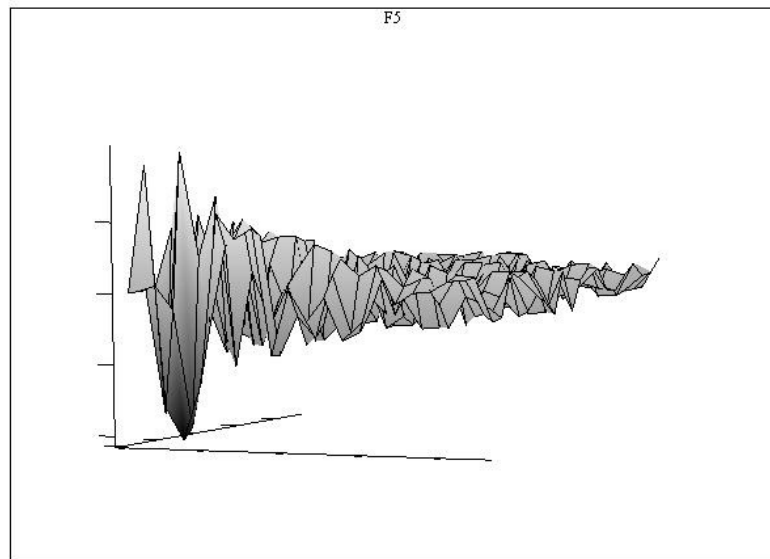
$$i := 0..82$$

$$j := 0..82$$

$$f(i,j) := \pi \frac{0.3}{360} j \cos\left(\frac{j}{\pi}\right) + \pi \frac{0.3}{360} i \sin\left(\frac{i}{\pi}\right)$$



$$f(i,j) := i \pi 0.5 \frac{\cos\left(\frac{2}{\pi} + 2 \frac{j}{\pi}\right)}{\left(1 + \frac{j}{2} + 2 \frac{i}{\pi}\right)} \quad \begin{array}{l} i := 0..45 \\ j := 0..45 \end{array}$$



BIBLIOGRAPHY

- [1] Kirkpatrick, S. Gellatt, Jr., C.D. and Vecchi, M.P. (1983), "Optimization by simulated annealing", Science, 220, p.671-680
- [2] Sullivan, K.A., Jacobson, S.H. (2001), "A Convergence analysis of generalized hill climbing algorithms", IEEE Transactions on Automatic Control, 46(8), p.1288-1293
- [3] Nader Azizi, Saeed Zofaghari (2004), "Adaptive temperature control for simulated annealing: a comparative study", Computers and Operations Research, Volume 31, Issue 14, p. 2439 - 2451
- [4] Charon, I and Hudry, O. (1993), "The noising method - a new method for combinatorial optimization", Operations Research Letters, 14, p.133-137
- [5] Charon, I and Hudry, O. (2001), "The noising methods - a generalization of some metaheuristics", European Journal of Operational Research, 135, p.86-101
- [6] Andrei A. Prudius, Sigrun Andradottir. (2005), "Two simulated annealing algorithms for noisy objective functions", Proceedings of the 2005 winter simulation conference
- [7] Fleicher M. (1993), "Assessing the performance of the simulated annealing algorithm using information theory", Doctoral Dissertation, Department of operations research, Case Western Reserve University, Cleveland, Ohio
- [8] Rutenbar, R.A. (1989), "Simulated annealing algorithms: an overview", Circuits and Devices Magazine, IEEE, Volume 5, Issue 1, p.19-26
- [9] Scott L. Rosen, Catherine M. Harmonosky (2005), "An improved simulated annealing simulation optimization method for discrete parameter stochastic systems", Computers and

Operations Research, v.32 n.2, p.343-358

[10] Walid Ben-Ameur (2004), "Computing the initial temperature of simulated annealing", Computational Optimization and Applications, Volume 29, Issue 3, p. 369 - 385

[11] Tai-Yue Wang, Kuei-Bin Wu (1999), "A parameter set design procedure for the simulated annealing algorithm under the computational time constraint", Computers & Operations Research 26 p.665-678

[12] Ling Wang, Liang Zhang (2005), "Stochastic optimization using simulated annealing with hypothesis test", Applied Mathematics and Computation Volume 174, Issue 2, 15 March 2006, Pages 1329-1342

[13] Mahmoud H. Alrefaei, "Sigriin Andraddttir (1999), A Simulated Annealing Algorithm Constant Temperature for Discrete Stochastic Optimization", Management Science, Vol. 45, No. 5, May 1999

[14] Hans-Georg Beyer(1999), "Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice", Computer Methods in Applied Mechanics and Engineering, 186 (2000) 239±267

[15] Hajek, B. (1988), "Cooling schedules for optimal annealing", Mathematics of Operations Research, 13 311-329

[16] G. Sasaki. (1989), "Simulated annealing-to cool or not", Systems & Control letters 12 443-447

[17] Gelfand and Mitter (1989), Gelfand, S. B., S. K, Mitter. 1989, "Simulated annealing with noisy or imprecise energy measurements", Journal of Optimization Theory and Application, 62 49-6

- [18] Gutjahr W.J and Pflug G.C (1996), “Simulated annealing for noisy cost functions”, Journal of Global Optimization, 8, 1-13
- [19] Andrei A. Prudius (2007), “Adaptive random search methods for simulation optimization”, Georgia Institute of Technology, Ph.D Thesis, August 2007
- [20] David E. Goldberg (1989), “Genetic algorithms in search”, Optimization and Machine Learning, Addison-Wesley, Reading Massachusetts, 1989