

Air Force Institute of Technology

AFIT Scholar

Theses and Dissertations

Student Graduate Works

3-2006

Optimization of a Multi-Echelon Repair System via Generalized Pattern Search with Ranking and Selection: A Computational Study

Derek D. Tharaldson

Follow this and additional works at: <https://scholar.afit.edu/etd>



Part of the [Operational Research Commons](#), and the [Statistical Models Commons](#)

Recommended Citation

Tharaldson, Derek D., "Optimization of a Multi-Echelon Repair System via Generalized Pattern Search with Ranking and Selection: A Computational Study" (2006). *Theses and Dissertations*. 3446.
<https://scholar.afit.edu/etd/3446>

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact AFIT.ENWL.Repository@us.af.mil.



**OPTIMIZATION OF A MULTI-ECHELON REPAIR
SYSTEM VIA GENERALIZED PATTERN
SEARCH WITH RANKING AND SELECTION:
A COMPUTATIONAL STUDY**

THESIS

Derek Tharaldson, Captain, USAF

AFIT/GOR/ENS/06-18

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GOR/ENS/06-18

OPTIMIZATION OF A MULTI-ECHELON REPAIR SYSTEM VIA
GENERALIZED PATTERN SEARCH WITH RANKING AND SELECTION:
A COMPUTATIONAL STUDY

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Operations Research

Derek Tharaldson, B.S.

Captain, USAF

March 2006

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GOR/ENS/06-18

OPTIMIZATION OF A MULTI-ECHELON REPAIR SYSTEM VIA
GENERALIZED PATTERN SEARCH WITH RANKING AND SELECTION:
A COMPUTATIONAL STUDY

Derek Tharaldson, B.S.

Captain, USAF

Approved:

Dr. James W. Chrissis (Thesis Advisor)

date

Mark A. Abramson, Lt Col, USAF (Reader)

date

Abstract

With increasing developments in computer technology and available software, simulation is becoming a widely used tool to model, analyze, and improve a real world system or process. However, simulation in itself is not an optimization approach. Common optimization procedures require either an explicit mathematical formulation or numerous function evaluations at improving iterative points. Mathematical formulation is generally impossible for problems where simulation is relevant, which are characteristically the types of problems that arise in practical applications. Further complicating matters is the variability in the simulation response which can cause problems in iterative techniques using the simulation model as a function generator.

The mixed-variable generalized pattern search with ranking and selection (MGPS-RS) algorithm for stochastic response problems is applied to an external simulation model, by means of the NOMADm MATLAB software package. Numerical results are provided for several configurations of a simulation model representing a multi-echelon repairable problem containing discrete, continuous, and categorical variables. Computational experience results are presented.

Acknowledgments

I would like to thank my advisor, Professor Jim Chrissis, for his guidance and expertise. His patience and understanding helped me through when things were not cooperating. I would also like to thank Lt Col Mark Abramson, the author of the optimization code that enabled me to complete this research, who provided valuable insight and help throughout this effort.

I am especially thankful for my wife – she was a source of strength and inspiration throughout this process.

Derek Tharaldson

Table of Contents

	Page
Abstract	iv
Acknowledgments.....	v
List of Figures	viii
List of Tables	ix
I. Introduction	1-1
1.1 Background of the Problem.....	1-1
1.2 Purpose of the Research	1-3
1.3 Overview of the Document	1-3
II. Literature Review	2-1
2.1 Simulation-based Optimization	2-1
2.2 Pattern Search.....	2-3
2.2.1 Generalized Pattern Search	2-4
2.2.2 Bound and Linear Constraints.....	2-5
2.2.3 Nonlinear Constraints	2-5
2.2.4 Mixed-Variable Generalized Pattern Search.....	2-6
2.2.5 Generalized Pattern Search with Noisy Response.....	2-8
2.3 Ranking and Selection	2-9
2.4 Multi-Echelon Repairable Systems	2-12
2.4.1 Basic Problem.....	2-12
2.4.2 Optimization of Multi-Echelon Systems	2-12
2.5 Summary	2-13

	Page
III. Methodology	3-1
3.1 MGPS-RS	3-1
3.2 Multi-Echelon Repair Model Formulation.....	3-5
3.3 Simulation Model Construction and Integration	3-11
IV. Results.....	4-1
4.1 Discrete Variable Models.....	4-1
4.2 Mixed Variable Models.....	4-2
4.3 Indifference Zone	4-5
V. Conclusions and Recommendations.....	5-1
5.1 Conclusions.....	5-1
5.2 Future Algorithm Research	5-1
5.3 Future Integration Research.....	5-2
VI. Appendix	6-1
Bibliography	BIB-1
Vita	VITA-1

List of Figures

Figure		Page
3.1.	A Generic R&S Procedure	3-2
3.2.	MGPS-RS Algorithm for Stochastic Optimization	3-3
3.3.	Network for a two-echelon repairable system	3-6
3.4.	Penalty assessed for violation of availability constraint.	3-9
4.1.	Deterministic model using Formulation A, $y = 2$	4-4

List of Tables

Table		Page
2.1.	Summary of techniques proposed for simulation-based optimization.	2-3
4.1.	Model 1A/1B Results, M=5.	4-2
4.2.	Model 1A/1B Results, M=20.	4-2
4.3.	Model 2A Results.	4-3
4.4.	Model 2B Results.	4-5
4.5.	Model 3 Results.	4-5
4.6.	Model 2A results using different indifference zone parameters. . . .	4-6
4.7.	Model 2B results using different indifference zone values.	4-6

OPTIMIZATION OF A MULTI-ECHELON REPAIR SYSTEM VIA
GENERALIZED PATTERN SEARCH WITH RANKING AND SELECTION:
A COMPUTATIONAL STUDY

I. Introduction

1.1 Background of the Problem

With the increasing developments in computer technology and available software, simulation is becoming a widely used tool to model, analyze, and improve a real world system or process. A simulation model is often developed because the system under study is so complex that an analytical model either is difficult to develop or cannot be formulated. Each simulation model can be classified as one of two types depending on the presence of random, or stochastic, elements in the model. Those models without random elements are called deterministic models. Each time it is run with a particular set of input variables, a deterministic model yields a unique output, or response, with no variation. Simulation models that include some randomness, which means the response changes randomly for each run, are called stochastic models. This research addresses stochastic models with responses that have variation, or noise, as a result of randomness for each run of the model.

Often an analyst uses the simulation model as an *ad hoc* means to optimize the real system. Typically with this approach, the analyst sets the input variables, runs the simulation for one or more replications, and evaluates the response. The analyst updates the input variables and repeats the process, ultimately finding a “best” solution. The input variables yielding that solution are then implemented in the real system. However, the evaluation of the response with a given set of input variables is complicated due to the stochastic nature of the simulation. While the objective may be to optimize an overall average response, the response for a specific set of input variables may only be the result of a small number of simulation runs. One solution to this problem is to run the simulation at each set of input variables for a large number of replications. However, this is usually

not practical, particularly if the number of input variable combinations to be run is large or the simulation takes a long time to run. What is needed is a procedure that can handle the noisy response of the simulation to determine which input variables produce the “best” response.

The noisy response may be modeled as an unknown response function $F(x, \omega)$ which depends upon an n -dimensional vector of controllable design variables $x \in \mathbb{R}^n$, and the vector ω , which represents random effects inherent to the system. The objective function f of the optimization problem is the expected performance of the system, given by

$$f(x) = \mathbb{E}_P[F(x, \omega)] = \int_{\Omega} F(x, \omega) P(d\omega), \quad (1.1)$$

where $\omega \in \Omega$ can be considered an element of an underlying probability space (Ω, \mathcal{F}, P) with sample space Ω , sigma-field \mathcal{F} , and probability measure P (51). Because the responses come from a black-box simulation which cannot be represented analytically, the probability distribution that defines the response $F(x, \omega)$ is assumed to be unknown but can be sampled.

An optimal solution for either a deterministic or stochastic simulation model can be difficult to obtain. Since f is usually unknown and analytical derivatives are unavailable, classical optimization approaches generally do not apply. Also, simulation runs, necessary for the numerical evaluation of f , may be computationally expensive. The presence of noise only complicates matters because f cannot be evaluated precisely. Statistical tests to determine if one x is better than another, a requirement for many search methods, may require a large number of repeated samples. Additional complications arise when x contains non-continuous variables, either discrete-numeric (*e.g.* integer-valued) or categorical. Categorical variables are those that can only take on values from a predefined list that have no ordinal relationship to one another. For example, a company may have different types of materials used in the manufacture of their products. The variables may represent those materials (*i.e.* 1 = steel, 2 = aluminum, etc). The class of optimization problems that includes continuous, discrete-numeric and categorical variables is known as mixed variable programming (MVP) problems (10).

1.2 Purpose of the Research

Sriver (50) developed a general algorithm for optimization of a stochastic system. Using a combined generalized pattern search with ranking and selection approach, the algorithm handles the mixed variable case, where the input vector contains continuous and discrete/categorical variables, with bound and linear constraints to optimize a black-box simulation response. This thesis builds on the work of Sriver by developing a simulation representing a real world system to apply the Mixed-Variable Generalized Pattern Search with Ranking and Selection (MGPS-RS) algorithm for stochastic response functions. Dunlap (24) incorporated the MGPS-RS algorithm into NOMADm, an optimization tool written in MATLAB¹ programming language by Abramson (1) and the software used to optimize the simulation model. The simulation model developed represents a Multi-Echelon Repairable System that has wide use in military and industrial environments.

1.3 Overview of the Document

The next chapter contains a review of the literature for simulation-based optimization strategies, including a brief survey of pattern search and ranking and selection, followed by a description of real multi-echelon repairable systems and related solution methodologies. Chapter 3 presents the application of the algorithm to a simulation built to represent the multi-echelon repair problem. Chapter 4 presents computational results. Finally, Chapter 5 offers some concluding remarks and recommendations for further advancement of the optimization via simulation using the MGPS-RS stochastic algorithm.

¹MATLAB[®] is a registered trademark with MathWorks.

II. Literature Review

Prior to applying the algorithm for the optimization of the system, it is important to review the literature for competing methods and examine the procedures used in the algorithm. Section 2.1 provides an overview of techniques and methods for simulation-based optimization. The approach used in this thesis, due to Srivier, uses a generalized pattern search method modified to handle the stochastic nature of optimization primarily for its independence from the need for gradient information and for its convergence theory. Section 2.2 gives a history and recent advancements of the pattern search class of algorithms. Srivier's work focused on extending pattern search methods to stochastic problems with noisy responses. He accomplished this by augmenting pattern search with a ranking and selection strategy. Section 2.3 describes some general ranking and selection approaches. The last section discusses the multi-echelon repair problem and some of the recent solution methodologies, including approaches that use simulation-based optimization.

2.1 Simulation-based Optimization

There are several papers that discuss the foundations, theoretical developments, and applications of a variety of techniques for simulation optimization (32, 26, 9, 13, 53). Ranking and selection, described more thoroughly in Section 2.3, is a popular methodology, but it does not handle a large number of candidate solutions and is impractical in the case of continuous variables. When applying these procedures to problems with continuous variables, the variables must be discretized. The intervals are often user-defined and can combinatorically explode the search space when not appropriately specified. Multiple comparison procedures run a number of replications and make conclusions on a performance measure by constructing confidence intervals (26). Likewise, multiple comparison procedures work better when the entire decision space is completely discrete. Random search can deal with a large number of candidate solutions, as well as upper and lower bounds. However, because previous information is not used at each iteration, formal convergence proofs for random search methods are rare, especially with continuous variables and noisy response (49).

Response surface methodology (RSM) is a class of procedures characterized by fitting a series of regression models to the responses from a simulation evaluated at several specific design points, then optimizing the resulting regression function. RSM is a popular method because of its use of well-known statistical properties. In application to simulation-based optimization, much of the research in polynomial based RSM prior to 1990 is summarized in Jacobson and Schruben (32), in which several improvements are discussed such as screening for variable reduction, allowance for multiple objectives, constraint-handling via the methods of feasible directions and gradient projection, variance reduction via common and antithetic pseudorandom numbers, and the effects of alternative experimental designs. RSM does have drawbacks, notably its lack of convergence and inability to handle categorical variables (50). Gradient-based methods, such as finite difference, perturbation analysis, and likelihood ratio, that estimate gradients of the objective function are well-known and widely used, but are restricted to the continuous variable problem. Stochastic approximation is also a gradient-based method that recursively estimates the gradient. This method possesses some convergence theory and certain variants can be quite efficient, but like the other gradient-based methods, it is geared towards continuous variable problems. The preceding techniques can be classified into two types: those for use with discrete input parameters and those for use with continuous variables. None of these are able to deal with the mixed variable problem.

Much of the simulation software available today contains some sort of optimization package, usually in the form of a heuristic search (27). A search heuristic is a method used to solve a problem essentially by trial and error. The procedure described at the beginning in this thesis is also an example of a heuristic. While heuristics often have an intuitive justification and can yield good solutions, they do not necessarily produce an optimal solution (56). Examples of search heuristics used in simulation software include evolutionary algorithms (genetic algorithms, evolutionary strategies and evolutionary programming), scatter search, tabu search, and simulated annealing. Their relative ease of use and generality (they can easily be adapted to mixed-variable problems and require only black-box response samples) have made them popular choices for simulation-based optimization. However, their application to stochastic problems has been largely unmodified

Table 2.1 Summary of techniques proposed for simulation-based optimization.

	Continuous	Discrete		Convergence
		Numeric	Categorical	
Ranking & Selection		X	X	X
Multiple Comparison		X		
Random Search		X	X	x
RSM	X	X		
Gradient-Based	X			X
Heuristics	X	X	X	

from their original form, relying on inherent robustness to noise, rather than explicitly accounting for noise (27). Boesel *et al.* (15) provide a good framework for simulation-based optimization. Their work is similar to the algorithms used in this thesis, particularly their use of ranking and selection. However, they employ the genetic algorithm heuristic, which requires the continuous variables to be discretized. Moreover, the user must choose the discrete interval for each continuous variable.

Table 2.1 summarizes this section, listing the techniques with their appropriate use and convergence theory. A field containing an “X” denotes the ability of the particular method to handle a certain type of variable or shown to have convergence. The small “x” signifies that random search methods have some convergence theory, but not when continuous variables are part of the decision space.

2.2 Pattern Search

Pattern search methods are a class of direct search methods for nonlinear optimization. The term “direct” means the methods use minimal information about the objective function, making a direct comparison of objective function values without the need for exact derivatives or their approximation. Since an objective function from a simulation may not be easily computed, direct search methods apply well to simulation-based optimization. Also, because of their broad applicability, allowing for mixed variable input vectors, they can be applied to simulations with a variety of discrete and continuous parameters.

Pattern search methods can be traced back to the work of Box in industrial efficiency during the 1950s (18) as a way to create a “layman’s” method, not relying on the experimental designs and regression of response methodology. Additional direct search methods were developed in the 1960s, but these algorithms were not formally shown to have strong convergence theory (2). However, in the late 1990s, Torczon introduced pattern search as a generalization of several existing methods and established convergence theory for the entire class of algorithms (54). Torczon’s paper was significant in that it established a global first-order convergence theory without ever explicitly computing or approximating derivatives.

2.2.1 Generalized Pattern Search. Pattern search algorithms are defined through a finite set of directions used at each iteration. The direction set and a step length parameter are used to construct a conceptual mesh centered about the current iterate (the incumbent). Trial points are selected from this discrete mesh, evaluated, and compared to the incumbent in order to select the next iterate. If an improvement is found among the trial points, the iteration is declared successful and the mesh is retained or coarsened; otherwise, the mesh is refined and a new set of trial points is constructed. Torczon proved that, for a continuously differentiable function f , a subsequence of the iterates $\{x_k\}$ produced by the generalized class of methods converges to a stationary point of f by showing that the mesh size (step length) parameter becomes arbitrarily small.

The mesh is defined by a finite set of directions that must be sufficiently rich to ensure that at least one of them is a direction of descent, provided that the current iterate is not a stationary point. Lewis and Torczon (36) applied the theory of positive linear dependence (22) to establish criteria for such a set of directions. Specifically, the set of directions form a positive spanning set for \mathbb{R}^n , which is defined as a set for which any vector in \mathbb{R}^n can be expressed as a nonnegative linear combination of these directions. Typically this set forms a positive basis, which is the smallest proper subset of a positive spanning set that still positively spans \mathbb{R}^n . A positive basis contains between $n + 1$ (a minimal set) and $2n$ (a maximal set) elements (22). Therefore, the worst case number of trial points per iteration can be bounded to $n + 1$ points by an appropriately constructed direction set.

2.2.2 Bound and Linear Constraints. Lewis and Torczon extend the results of (54) to problems with bound constraints (37) and a finite number of linear constraints (38). In these situations, the set of positive spanning directions used in the algorithm must be chosen such that they conform to the geometry of the nearby constraint boundaries. Using this construct, at least one direction in the positive spanning set must be a feasible descent direction, unless the current iterate is already a stationary point.

Audet and Dennis (11) devise an alternative version of pattern search for bound and linearly constrained problems, along with a new convergence theory based on the nonsmooth calculus of Clarke (20) that generalizes previous results. Second-order behavior is described in (3). Audet and Dennis explicitly separate the evaluation of points into two distinct steps, an optional SEARCH and a local POLL step. The step allows the user to define a search strategy to seek an improved mesh point. The SEARCH step contributes nothing to the convergence theory, but allows the user to apply any finite heuristic to increase the efficiency and possibly affect the quality if a correct search is chosen (17). Examples for the use of this approach include randomly selecting a space-filling set of points using Latin hypercube design or applying a few iterations of a genetic algorithm (50). For problems with computationally expensive functions, the SEARCH step is often used to construct inexpensive surrogate functions and then optimize the resulting surrogate problem (*e.g.*, see (17)). Dunlap (24) studied the use of surrogates in pattern search methods, applied to mixed variable stochastic problems. The POLL step evaluates specific points on the mesh, referred to as the poll set, that are adjacent to the current iterate with respect to the current set of positive spanning directions.

2.2.3 Nonlinear Constraints. Audet and Dennis (11) extend their approach to nonlinear constraints by implementing a filter method (25), which accepts new iterates if the usual improvement in objective function is found, but also if an aggregate constraint violation function is reduced. Lewis and Torczon (39) give an alternate approach where a bounded subproblem, formed from an augmented Lagrangian function (21), is solved approximately using pattern search. Motivated by weaknesses in the convergence theory of the filter pattern search method (11), Audet and Dennis (12) introduced a new class

of algorithms, called Mesh Adaptive Direct Search (MADS), which generalizes GPS by allowing more flexibility in the selection of directions. In fact, MADS has been proved to converge to second-order stationary points, and even local minimizers of general set-constrained nonlinear optimization problems (5). MADS uses a barrier method, replacing the filter method, that assigns a value of $+\infty$ to infeasible iterates without ever evaluating their objective function.

2.2.4 Mixed-Variable Generalized Pattern Search. Audet and Dennis (10) provided a framework for mixed-variable problems with bound and linear constraints by including discrete neighborhood sets in the definition of the mesh. Their algorithm was applied successfully in (35) to the design of a thermal insulation system. In the mixed variable case, the POLL step is augmented with a search of points in a user-defined set of discrete neighbors. If the POLL step is unsuccessful in finding an improved solution, an EXTENDED POLL step is performed, in which a POLL is performed around any discrete neighbor that has an objective function value sufficiently close to that of the incumbent (*i.e.* within a tolerance $\xi > 0$). This algorithm allows extension of the convergence theory to the mixed variable domain but incurs a cost of more function evaluations, particularly if the user allows a large number of discrete neighbors. Abramson (2) extended the work done by Audet and Dennis by allowing nonlinear constraints in the mixed-variable case through the use of a filter. This work was applied successfully in (4) to the design of a load-bearing thermal insulation system.

The following description by Sriver (50) explains the definition of the mesh and poll set developed by Audet and Dennis (10):

“A set of positive spanning directions D^i is constructed for each unique combination $i = 1, 2, \dots, i_{\max}$, of values that the discrete variables may take, *i.e.*,

$$D^i = G_i Z_i, \tag{2.1}$$

where $G_i \in \mathbb{R}^{n^c \times n^c}$ is a nonsingular generating matrix and $Z_i \in \mathbb{Z}^{n^c \times |D^i|}$. The mild restrictions imposed by (2.1) are necessary for the convergence theory. The mesh is then formed as the direct product of Θ^d with the union of a finite

number of meshes in Θ^c , *i.e.*,

$$M_k(x_k) = \Theta^d \times \bigcup_{i=1}^{i_{\max}} \left\{ x_k^c + \Delta_k D^i z \in \Theta^c : z \in \mathbb{Z}_+^{|D^i|} \right\}. \quad (2.2)$$

At iteration k , let $D_k^i \subseteq D^i$ denote the set of poll directions corresponding to the i^{th} set of discrete variable values and define $D_k = \bigcup_{i=1}^{i_{\max}} D_k^i$. The poll set is defined with respect to the continuous variables centered at the incumbent while holding the discrete variables constant. Its form is

$$P_k(x_k) = \{x_k + \Delta_k(d, 0) \in \Theta : d \in D_k^i\} \quad (2.3)$$

for some $1 \leq i \leq i_{\max}$, where $(d, 0)$ denotes the partitioning into continuous and discrete variables; 0 means the discrete variables remain unchanged, *i.e.*, $x_k + \Delta_k(d, 0) = (x_k^c + \Delta_k d, x_k^d)$.

Within the GPS framework, mixed variables are incorporated through the use of discrete neighbors $\mathcal{N}(x_k)$ at each point x_k in the domain. The points in $\mathcal{N}(x_k)$ include the current point x_k and other points that differ in at least one of the discrete variables. For example, if the discrete variables are defined as integers, a neighborhood structure may be defined by holding the continuous variables constant and changing only one of the discrete variables by a single unit, *i.e.*,

$$\mathcal{N}(x_k) = \{y \in \Theta : y^c = x_k^c, \|y^d - x_k^d\|_1 \leq 1\}. \quad (2.4)$$

However, if the discrete variables are categorical, then this neighbor set may not be well-defined. For example, in a manufacturing process, a categorical variable might be material type, in which case, the norm function is not well-defined, since there is no measure of distance for this nonnumeric variable. In this case, changing the material type from one designated by a “1” to one designated by a “3” may just as valid a changing it to “2”. Thus for categorical variables, the discrete neighbor set must be defined by the user. It should also be noted that a change in a discrete variable may force an accompanying change to the continuous variables. As an example, if a continuous variable, such as thickness, were associated with each material, then a discrete neighbor would be of a different material type than the current point, but it might also have a different thickness.

2.2.5 *Generalized Pattern Search with Noisy Response.* The use of generalized pattern search applied to stochastic optimization is limited. Trosset (55) analyzed convergence in the unconstrained, continuous case by viewing the iterates as a sequence of binary ordering decisions. For $\Lambda_k = f(X_k) - f(Y)$, where X_k is the current iterate and Y is a trial point from the mesh, the statistical hypothesis test,

$$H_0 : \Lambda_k \leq 0 \tag{2.5}$$

$$H_1 : \Lambda_k > 0,$$

is conducted, in which Y is accepted as the new iterate if the null hypothesis H_0 is rejected. The test is subject to Type I and Type II errors. A Type I error is made if H_0 is rejected when it is actually true and occurs with probability α , the significance level of the test. A Type I error would select a new iterate incorrectly. A Type II error is made if H_0 is accepted when H_1 is true and occurs with probability β . A Type II error would not update the iterate with the new, better point. The number of Type I errors can be controlled, ensured (with probability equal to one) to be a finite number, by selecting a sequence of significance levels $\{\alpha_k\}$ such that $\sum_{k=1}^{\infty} \alpha_k < \infty$. In addition, let $\{\lambda_k\}$ be a sequence of alternatives satisfying $\lambda_k > 0$, $\lambda_k = o(\Delta_k)$, and $\lambda_k \rightarrow 0$ that require power $1 - \beta_k$ when conducting the test in (2.5). Choosing a sequence $\{\beta_k\}$ such that $\sum_{k=1}^{\infty} \beta_k < \infty$ ensures a finite number of Type II errors when $\Lambda_k \geq \lambda_k$. Trosset claims that a sequence of iterates from a GPS algorithm can be shown to converge almost surely to a stationary point of f but, in practice, would require a very large number of samples to guarantee convergence (55). He shows through a power analysis that the number of samples per iteration grows faster than the squared reciprocal of the mesh size parameter. A power analysis is a statistical technique to determine the required sample size to guarantee a probability $(1 - \beta)$ of rejecting H_0 when H_1 is true. Also, Ouali *et al.* (43) applied multiple repetitions of generalized pattern search directly to a stochastic simulation model to seek minimum cost maintenance policies where costs were estimated by the model. Srivier (50) was able to overcome the problem highlighted by Trosset with the use of an indifference zone in ranking and selection procedures.

2.3 Ranking and Selection

Ranking and selection (R&S) procedures are “statistical methods specifically developed to select the best system, or a subset of systems that includes the best system, from a collection of competing alternatives” (28). The following overview of R&S procedures details their use in iterative search routines applied to stochastic optimization via simulation.

Indifference zone and subset selection are the two main topics in R&S methods (26). Indifference-zone procedures guarantee a solution within $\delta > 0$ above the true best solution with user-specified probability $(1 - \alpha)$, where $\alpha \in [0, 1]$. The parameter δ , which represents a measure of tolerance known as the *indifference zone*, is called the indifference zone parameter. R&S procedures collect response samples from the alternatives using a single stage or multiple stages of sampling, check a certain stopping criteria, then either continue sampling or stop and select the alternative with the smallest response estimate in the final stage (49). The original procedure by Bechhofer (14) determines the number of samples required of each iteration beforehand, or *a priori*, according to a tabular value related to the user-defined values for δ and α . A potential drawback, especially in the area of simulation, is that Bechhofer’s method assumes that the variance in the response samples is known and equal across all alternatives.

Dudewicz and Dalal (23) and Rinott (46) extended the approach to apply to problems with unknown and unequal variances in response samples. They used a two-stage process, in which an initial stage of sampling is done to estimate the variances, which are then used to determine the number of samples needed in the second stage to ensure the probability of correct selection. Subset selection aims to select a subset of at most m points and guarantees that this subset contains at least the best solution, with probability at least $(1 - \alpha)$ (41). Subset selection is more useful when the number of candidates is large.

To define the requirements for a general indifference-zone R&S procedure, Srivier (50) gives the following formulation

“...consider a finite set $\{X_1, X_2, \dots, X_{n_C}\}$ of $n_C \geq 2$ candidate design points. For each $i = 1, 2, \dots, n_C$, let $f_i = f(X_i) = E[F(X_i, \omega)]$ denote the

true objective function value. The f_i values can be ordered from minimum to maximum as,

$$f_{[1]} \leq f_{[2]} \leq \dots \leq f_{[n_C]}. \quad (2.6)$$

The notation $X_{[i]}$ indicates the candidate with the i th best (lowest) true objective function value. If at least one candidate has a true mean within δ of the true best, *i.e.* $f_{[i]} - f_{[1]} < \delta$ for some $\delta > 0$ and $i \geq 2$, then the procedure is indifferent in choosing $X_{[1]}$ or $X_{[i]}$ as the best. The probability of correct selection (CS) is defined as

$$P\{CS\} = P\{\text{select } X_{[1]} \mid f_{[i]} - f_{[1]} \geq \delta, i = 2, \dots, n_C\} \geq 1 - \alpha, \quad (2.7)$$

where $\delta > 0$ and $\alpha \in (0, 1)$ are user specified. A random selection of the candidates guarantees at least $P\{CS\} = \frac{1}{n_C}$, so the significance level must satisfy $0 < \alpha < 1 - \frac{1}{n_C}$.

When a candidate number is large, traditional multi-stage indifference-zone procedures may prescribe too many simulation runs because they are based on the least favorable configuration assumption. That is, the best candidate has a true mean exactly δ better than all other candidates, which are all tied for the second best (53). As a result, the procedures can call for an unnecessarily high number of samples in the final stage to guarantee that (2.7) holds.

Two recent directions in R&S research reflect attempts to address this issue. The first approach has been to combine R&S with some type of search strategy to explore a large solution space. Ólafsson (42) and Pichitlamken and Nelson (45) each introduce an iterative technique that combines R&S with an adaptive sampling algorithm known as nested partitioning (NP), which is used to search the feasible space of (possibly large) combinatorial problems for a global optimum. The techniques use discrete time Markov chains to show almost sure convergence to a global optimum of the discrete variable space.

Heuristic search methods with incorporated R&S procedures have recently been developed for stochastic optimization. Ahmed and Alkhamis (8) describe and analyze a globally convergent algorithm for optimization over a discrete domain that uses R&S procedures within simulated annealing. Boesel *et al.* (16) and Hedlund and Mollaghasemi (30) apply R&S procedures to genetic algorithms.

The second approach for resolving the high sample number required for a large set of candidates combines the two topics of subset selection and indifference-zone selection.

These procedures identify and eliminate inferior solutions and then select the best from the remaining candidates. Nelson *et al.* (41) present a general theory and procedure that balances computational and statistical efficiency. This approach maintains a probability guarantee for selecting the best solution when using the combined technique. Kim and Nelson (33) and Goldsman *et al.* (28) present efficient fully sequential indifference-zone techniques that eliminate alternatives deemed inferior as sampling progresses.

Categorical and discrete variables are readily handled by modern R&S techniques since all design alternatives are determined *a priori* and corresponding variable values can be set accordingly. However, R&S procedures have difficulty with a large number of solutions. The existing provably convergent techniques (8, 42, 45) that combine R&S with adaptive search currently address entirely discrete domains. Continuous variables can be dealt with through discretization, but depending on the interval chosen, this can cause a combinatorial explosion of the search space and an increase in computational expense.

To improve on the implementation by Trosset and applicability to the general case, Sriver uses a ranking and selection procedure to identify a new iterate. Sriver (50) lists the specific advantages to include:

- It is amenable to parallelization techniques since several trial solutions can be considered simultaneously in the selection process rather than only two (incumbent and candidate).
- R&S procedures detect the relative order, rather than generate precise estimates, of the candidate solutions. This is generally easier to do (27) and provides computational advantages.
- Selection error is limited to Type II error only, *i.e.*, making an incorrect selection of the best candidate; Type I error is eliminated based on the assumption of a *best* system among the candidates.
- The use of an indifference zone parameter can be easily and efficiently adapted for algorithm termination.

Three such procedures were selected for use in MGPS-RS: Rinott's two-stage procedure (46), a screen-and-select (SAS) procedure of Nelson *et al.* (41), and the Sequential

Selection with Memory (SSM) procedure of Pichitlamken and Nelson (44). The SSM procedure is implemented by Dunlap (24) in the NOMADm software (1) and explained further in Chapter 3.

2.4 Multi-Echelon Repairable Systems

2.4.1 Basic Problem. Multi-echelon, or multilevel, problems exist in many arenas. One such area that has considerable interest is in the design and performance of maintenance systems for a repairable item. The general problem to be investigated is the determination of the optimal spare levels and repair channels in a maintenance system, in which a finite number of items is desired to be operational at any given time, and in which queuing can occur at the repair facilities if all channels are busy (6). The system usually consists of one or more bases at the lowest level (or first echelon), one or more depots at the highest level, and any number (to include zero) of intermediate levels in between. Multi-echelon systems can take on many forms to include number of levels, number of facilities at each level, number of machines in the system that depend on the use of the item, scheduling, resupply strategies (hierarchical and/or lateral), transportation delays, and cannibalization or condemnation. It is because of this complexity that multi-echelon models are often analyzed through simulation (34).

2.4.2 Optimization of Multi-Echelon Systems. Optimization of multi-echelon models have been approached both analytically and through simulation. Sherbrooke developed the METRIC model to minimize expected backorders, which is equivalent to maximizing availability when no cannibalization of parts is assumed (47). Using the assumption of an infinite number of repair channels, Sherbrooke made use of Palm's Theorem to calculate steady-state probability distributions for the number of units due in from repair. Gross *et al.* (29) relaxed that assumption and, using Markovian properties of the exponential distribution, formulated the expression for machine availability in terms of the decision variables, number of spares and number of repair channels at each facility. More recent applications in this area have been based on simulation-based optimization. Chris-sis and Gecan (19) use a direct search technique to iteratively seek an optimal solution to

the multi-echelon system. Ahmed *et al.* (7) present an integrated approach of simulated annealing and simulation to determine the design parameters of a multi-echelon repairable item inventory system. Köchel and Nielander (34) use a genetic algorithm to optimize a five-level simulation model. Alkhamis and Ahmed (6) use the results from the analytical solution by Gross *et al.* (29) to validate their evolutionary technique of particle swarm optimization. Those results are also used in the evaluation of the procedures used in this thesis.

2.5 Summary

This chapter has provided an overview of the literature pertaining to this study. Section 2.1 reviewed the various approaches to simulation-based optimization. Each method was reviewed to highlight the ability to handle different types of variables and convergence properties. The ideas of pattern search and ranking and selection used in the MGPS-RS algorithm were outlined in Sections 2.2 and 2.3, respectively. Finally, the multi-echelon repair problem was introduced in Section 2.4, and some previous attempts at optimizing this type of problem were discussed. The next chapter further details the elements of the MGPS-RS algorithm and its use in the NOMADm software. It also explains the specifics of the multi-echelon system models used in the application of the MGPS-RS algorithm.

III. Methodology

This chapter presents the methodology used in the optimization of the multi-echelon repair system. The first section explains the mixed variable generalized pattern search with ranking and selection algorithm used to optimize the system. Section 3.2 provides details of the model formulation for the multi-echelon repair problem. The last section discusses the simulation model and integration with the optimization code.

3.1 MGPS-RS

When using simulation as a black-box generator for objective function values, obviously the true values in (2.6) are not available. Therefore, it becomes necessary to use samples of the simulation response F to create estimates. Let n_c be the number of candidates in the candidate set C . For each $i = 1, 2, \dots, n_c$, let s_i be the total number of replications and let $\{F_{is}\}_{s=1}^{s_i} = \{F(Y_{is})\}_{s=1}^{s_i}$ be the set of responses obtained through simulation, where Y_{is} is the input vector for design i and replication s . Then for each $i = 1, 2, \dots, n_c$, the sample mean \bar{F}_i is computed as

$$\bar{F}_i = \frac{1}{s_i} \sum_{s=1}^{s_i} F_{is} \quad (3.1)$$

and is an estimator for f_i . These sample means may be ordered in the same manner as the true responses in (2.6). The ranking and selection procedure determines the ordering of the candidates with the i th best estimated response value denoted by $\hat{Y}_{[i]} \in C$. The candidate with the lowest mean response, $\hat{Y}_{[1]}$, is chosen as the best point. A general R&S algorithm, as given by Srivier (50), is shown in Figure 3.1, where the input parameters include a candidate set C , significance level α , and indifference zone δ . Procedure $RS(C, \alpha, \delta)$ returns the best candidate $\hat{Y}_{[1]}$.

Procedure RS(C, α, δ)

Inputs: A set $C = \{Y_1, Y_2, \dots, Y_{n_C}\}$ of candidate solutions, significance level α , and indifference zone parameter δ .

Step 1: For each candidate Y_q , use an appropriate technique to determine the number of samples s_i required to meet the probability of correct selection guarantee, as a function of α, δ and response variation of Y_q .

Step 2: Obtain sampled responses $F_{is}, i = 1, \dots, n_C$ and $s = 1, \dots, s_i$. Calculate the sample means \bar{F}_i based on the s_i replications according to (3.1). Select the candidate associated with the smallest estimated sample mean, $\hat{Y}_{[1]}$ as having the δ -near-best mean.

Return: $\hat{Y}_{[1]}$

Figure 3.1 MGPS-RS Algorithm for Stochastic Optimization

Sriver incorporates this R&S procedure into the MGPS algorithm for deterministic mixed variable optimization, due to Audet and Dennis (10). The MGPS-RS algorithm of Sriver (50) is presented in Figure 3.2. The binary comparison of the incumbent and trial points used in the deterministic case is replaced with Procedure RS(C, α, δ). The algorithm can use any specific R&S procedure, as long as it satisfies the probability of correct selection guarantee given in (2.7).

MGPS-RS Algorithm for Stochastic Responses

Initialization: Set the iteration counter k to 0. Set the R&S counter r to 0. Choose a feasible starting point, $X_0 \in \Theta$. Set $\Delta_0 > 0$, $\xi > 0$, $\alpha_0 \in (0, 1)$, and $\delta_0 > 0$.

1. SEARCH step (optional): Employ a finite strategy to select a subset of candidate solutions, $S_k \subset M_k(X_k)$ defined in (2.2) for evaluation. Use Procedure RS($S_k \cup \{X_k\}$, α_r , δ_r) to return the estimated best solution $\hat{Y}_{[1]} \in S_k \cup \{X_k\}$. Update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, and $r = r + 1$. If $\hat{Y}_{[1]} \neq X_k$, the step is successful, update $X_{k+1} = \hat{Y}_{[1]}$, $\Delta_{k+1} \geq \Delta_k^*$ and $k = k + 1$ and repeat Step 1. Otherwise, proceed to Step 2.
2. POLL step: Set extended poll trigger $\xi_k \geq \xi$. Use Procedure RS($P_k(X_k) \cup \mathcal{N}(X_k)$, α_r , δ_r) where $P_k(X_k)$ is defined in (2.3) to return the estimated best solution $\hat{Y}_{[1]} \in P_k(X_k) \cup \mathcal{N}(X_k)$. Update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, and $r = r + 1$. If $\hat{Y}_{[1]} \neq X_k$, the step is successful, update $X_{k+1} = \hat{Y}_{[1]}$, $\Delta_{k+1} \geq \Delta_k^*$ and $k = k + 1$ and return to Step 1. Otherwise, proceed to Step 3.
3. EXTENDED POLL step: For each discrete neighbor $Y \in \mathcal{N}(X_k)$ that satisfies the extended poll trigger condition $\bar{F}(Y) < \bar{F}(X_k) + \xi_k$, set $j = 1$ and $Y_k^j = Y$ and do the following.
 - (a) Use Procedure RS($P_k(Y_k^j)$, α_r , δ_r) to return the estimated best solution $\hat{Y}_{[1]} \in P_k(Y_k^j)$. Update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, and $r = r + 1$. If $\hat{Y}_{[1]} \neq Y_k^j$, set $Y_k^{j+1} = \hat{Y}_{[1]}$ and $j = j + 1$ and repeat Step 3a. Otherwise, set $Z_k = Y_k^j$ and proceed to Step 3b.
 - (b) Use Procedure RS($X_k \cup Z_k$, α_r , δ_r) to return the estimated best solution $\hat{Y}_{[1]} = X_k$ or $\hat{Y}_{[1]} = Z_k$. Update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, and $r = r + 1$. If $\hat{Y}_{[1]} = Z_k$, the step is successful, update $X_{k+1} = \hat{Y}_{[1]}$, $\Delta_{k+1} \geq \Delta_k^*$ and $k = k + 1$ and return to Step 1. Otherwise, repeat Step 3 for another discrete neighbor that satisfies the extended poll trigger condition. If no such discrete neighbors remain, set $X_{k+1} = X_k$, $\Delta_{k+1} < \Delta_k^*$ and $k = k + 1$ and return to Step 1.

*NOTE: The update rules for Δ_k in the algorithm can be found in (1) and have important implications for the convergence of the algorithm.

Figure 3.2 MGPS-RS Algorithm for Stochastic Optimization

The following assumptions are necessary for the convergence theory of the MGPS-RS algorithm to hold (50):

- A1 All iterates X_k produced by the MGPS-RS algorithm lie in a compact set.
- A2 The objective function f is continuously differentiable with respect to the continuous variables when the discrete variables are fixed.
- A3 For each set of discrete variables X^d , the corresponding set of directions $D^i = G_i Z_i$, as defined in (2.1), includes tangent cone generators for every point in Θ^c .
- A4 The rule for selecting directions D_k^i conforms to Θ^c for some $\varepsilon > 0$.
- A5 For each $q = 1, 2, \dots, n_C$, the responses $\{F_{qs}\}_{s=1}^{s_q}$ are independent, identically and normally distributed random variables with mean $f(X_q)$ and unknown variance $\sigma_q^2 < \infty$, where $\sigma_\ell^2 \neq \sigma_q^2$ whenever $\ell \neq q$.
- A6 The sequence of significance levels $\{\alpha_r\}$ satisfies $\sum_{r=0}^{\infty} \alpha_r < \infty$, and the sequence of indifference zone parameters $\{\delta_r\}$ satisfies $\lim_{r \rightarrow \infty} \delta_r = 0$.
- A7 For the r th R&S procedure considering candidate set $C = \{Y_1, Y_2, \dots, Y_{n_C}\}$, Procedure RS(C, α_r, δ_r) guarantees correctly selecting the best candidate $Y_{[1]} \in C$ with probability of at least $(1 - \alpha_r)$ whenever $f(Y_{[q]}) - f(Y_{[1]}) \geq \delta_r$ for any $q \in \{2, 3, \dots, n_C\}$.
- A8 For all but a finite number of MGPS-RS iterations and sub-iterations, the best solution $Y_{[1]} \in C$ is unique; *i.e.*, $f(Y_{[1]}) \neq f(Y_{[q]})$ for all $q \in \{2, 3, \dots, n_C\}$ where $C = \{Y_1, Y_2, \dots, Y_{n_C}\} \subset M(X_k)$ at iteration k .

Sliver includes a brief discussion on these assumptions. Assumptions **A1**, **A3**, and **A5** are noted since they have implications in the model formulation and presentation of results in this thesis. Under these assumptions, Sliver *et al.* (52) proved almost sure convergence (*i.e.*, with probability one) of a subsequence of iterates to a first-order stationary point. First-order stationarity in a mixed variable domain was first formally defined by Lucidi *et al.* (40) in the context of a more general framework for mixed variable optimization.

Since the NOMADm optimization code is used, it is important to highlight the specific R&S procedure implemented in the code. Based on the results of the computational results in (50), Dunlap (24) chose the SSM procedure of Pichitlamken and Nelson (44) for its superior performance. SSM is a fully sequential procedure designed for use with iterative search routines, such as the class of pattern search methods. The procedure collects one sample at a time from every candidate and eliminates clearly inferior alternatives as sampling progresses. To identify inferior candidates, SSM performs a pairwise statistical test at every iteration for each of the candidates. By removing the candidates that are statistically inferior to all the other members of the candidate set, SSM provides a computationally efficient procedure to select the best candidate. Dunlap states that another advantage of SSM is the ability to use previously stored sampling data (24). This mitigates some of the cost of obtaining additional simulation responses at each iteration of the optimization algorithm.

3.2 Multi-Echelon Repair Model Formulation

Since the multi-echelon repair problem has been solved analytically under certain assumptions, these solutions can be used as a baseline to compare algorithm performance on similar models. The specific model and analytical solution used as a baseline in this thesis is from Gross *et al.* (29), who gives an exact solution to the multi-echelon problem with exponential distributions for holding rates and arrivals according to a Poisson process; these results are used as a baseline comparison. Gross *et al.* devise a simple two-echelon model with a single base and single depot, where M machines generate item failures according to a Poisson process with mean $\lambda = \mu_U$. A failure is removed from the machine and replaced with a spare from the base inventory. If no spare is available, the machine waits for an item to become available, either through repair at the base or arrival from the depot. Failed items extracted from a machine are determined to be either base-serviceable or needing advanced repair that cannot be performed at the base. A certain percentage of failed items, α , are deemed to be base serviceable, with the remaining $(1 - \alpha)$ sent to the depot for repair.

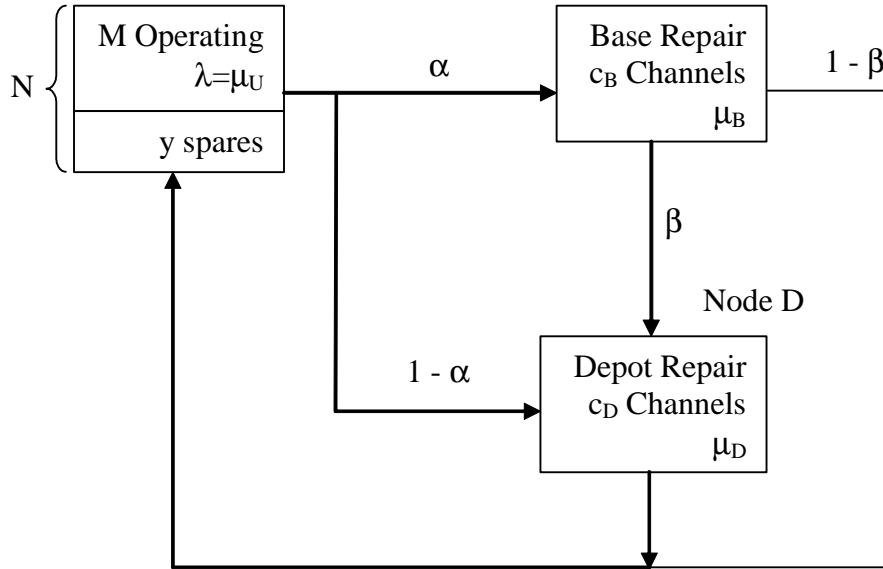


Figure 3.3 Network for a two-echelon repairable system

Both the base and depot repair items according to an exponential distribution with mean μ_B and μ_D , respectively. Each base has a specific number of repair channels. An item that arrives at a location with all of the channels busy waits in a queue. There is a single queue at each location, which follows a FIFO (first in, first out) rule. When a repair is complete, the item is returned to the spares inventory unless an inoperable machine is awaiting the item, in which case, the item is installed and the machine returns to service. However, a percentage of items, β , after undergoing repair at the base, need further repair and are sent to the depot.

Recall that Gross *et al* (29), with the assumption that all processes are Markovian, develop an expression for availability in terms of the decision variables. The decision variables are number of spares y to be inventoried at the base, and the number of channels at the base and depot, c_B and c_D , respectively. The number of spares at the depot is assumed to be limitless. A diagram of the flow model is shown in Figure 3.3. The objective function is defined to be a linear combination of the costs of the repair channels and the

base-inventoried spares. Gross *et al* (29) define the problem mathematically as

$$\begin{aligned} \min Z &= k_y y + k_B c_B + k_D c_D & (3.2) \\ \text{s.t. } \sum_{n=M}^{M+y} p_n &\geq A, \end{aligned}$$

where M is the number of machines and k_i is the cost per unit, $i = y, B, D$. The constraint is absolute availability A , with p_n equal to the steady state probability that n units are operational. Gross *et al* (29) use

$$M = 5, k_y = 20, k_B = 8, k_D = 10, A = 0.9, \alpha = 0.5, \beta = 0.5, \mu_U = 1, \mu_B = \mu_D = 5$$

as the parameters for this model. Unless noted otherwise, these parameter values are used in the remainder of the construction of the simulation and mathematical models.

In (3.2), availability is the stochastic component of the problem to be supplied by the simulation model. However, the availability level appears in the inequality constraint and not in the objective function. Since MGPS-RS is not currently able to handle stochastic nonlinear constraints, the problem formulation given in (3.2) must be modified. Two approaches for doing so are considered: 1) reformulate (3.2) as an unconstrained problem with a term added to the objective function to penalize infeasibility, or 2) swapping the constraint and objective function. The first approach becomes an unconstrained minimization problem with both deterministic and stochastic components. In the latter case, the problem becomes one of maximizing availability (or minimizing expected backorders, as in Sherbrooke (48), subject to a linear cost or budget constraint. This problem is essentially an integer knapsack problem, containing a linear objective function subject to a single linear constraint (56). The first formulation is

$$\min Z = 20y + 8c_B + 10c_D + P_A(0.9 - F), \quad (\text{Formulation A})$$

where P_A is the penalty and F is the absolute availability taken from the simulation response. Penalty terms are well-suited for problems in which some of the constraint functions require noisy response evaluations from the model, since it cannot be determined

prior to simulation if a design is feasible with respect to these constraints. However, as in the deterministic case, it is noted that penalty methods may suffer from computational difficulties due to ill-conditioning (50).

The second formulation is

$$\begin{aligned} \min Z &= 100 \left[1 - \left(\sum_{n=M}^{M+y} p_n \right) \right] && \text{(Formulation B)} \\ \text{s.t. } &20y + 8c_B + 10c_D \leq B \end{aligned}$$

where B is the budget constraint. The availability maximization problem is converted to a minimization problem by subtracting from 1, since MGPS-RS is presented as a minimization algorithm. However, the problem still behaves as a knapsack with an increase in the decision variables yielding a better result. The result is multiplied by the constant 100 simply to signify a percentage.

To apply Formulation A, certain parameters must be determined. The first is how to deal with values above the availability value A . In this formulation, there is no cost improvement for exceeding the availability threshold. Therefore, the penalty is only applied for values of availability less than 90% and all responses above are set to the value of 0.9. Also, the penalty coefficient for the new objective function must be determined.

Since availability was originally a constraint, it is important that the penalty coefficient in the objective function dominate the other terms. This approach attempts to force the solution to satisfy the constraint, and is consistent with the general penalty function philosophy. In this case, the coefficient was chosen to be ten times larger than the other terms. Also, the square root of the deviation from the constraint is taken to signify that even minor violations of the penalty get a harsh penalty. Figure 3.4 shows the effect of the additional penalty term on the objective function. The resulting model is

$$\begin{aligned} \min Z &= 20y + 8c_B + 10c_D + 2000\sqrt{(0.9 - F_{0.9})} && \text{(MODEL1A)} \\ &y, c_B, c_D \in \mathbb{Z}^+ \end{aligned}$$

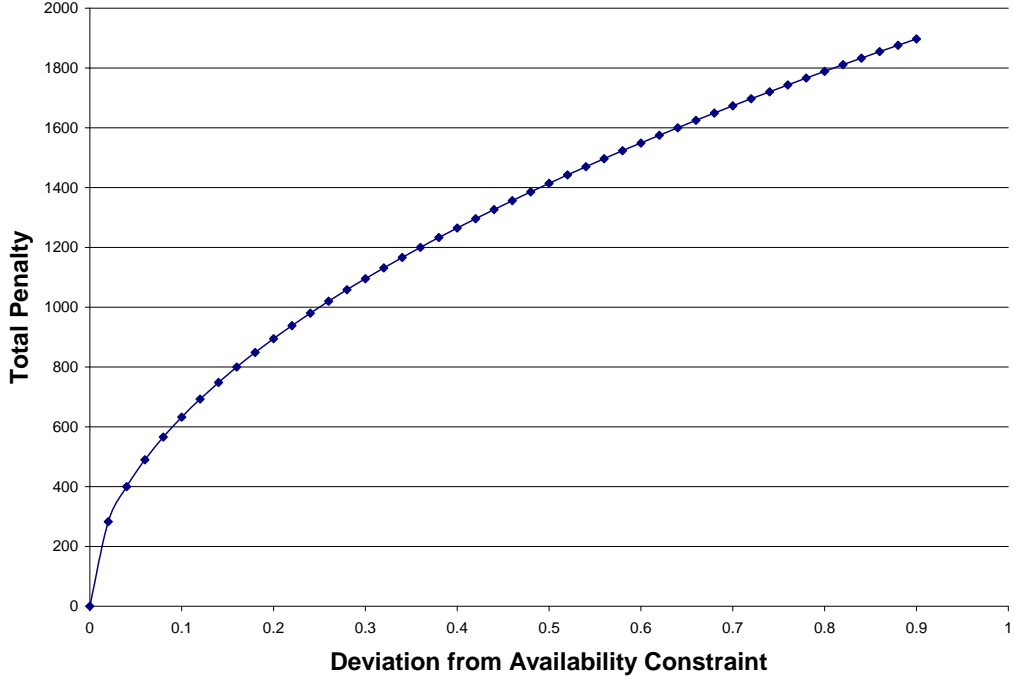


Figure 3.4 Penalty assessed for violation of availability constraint.

where

$$F_{0.9} = \begin{cases} F & \text{if } F \leq 0.9 \\ 0.9 & \text{if } F > 0.9. \end{cases}$$

The second approach only requires determination of the budget B . A budget of 96 monetary units was chosen to maintain a comparison with the analytical solution. The resulting mathematical model is

$$\begin{aligned} \min Z &= 1 - \left(\sum_{n=M}^{M+y} p_n \right) && \text{(MODEL1B)} \\ \text{s.t. } &20y + 8c_B + 10c_D \leq 96 \\ &y, c_B, c_D \in \mathbb{Z}^+. \end{aligned}$$

The decision variables in **MODEL1A** and **MODEL1B** are all discrete. This causes the decision space to be the set of user-defined neighbors because, with no continuous variables, the mesh is an empty set. The discrete model is still used for analysis of the algorithm and the results are reported in Chapter 4. To test a true mixed variable problem, the model is further adapted to provide for the inclusion of continuous variables. In the updated model, instead of the number of repair channels being decision variables, they are fixed at three each. To replace them, continuous variables representing the amount of workday the base and depot operate are introduced. All other variables and parameters remain the same. The continuous variables can be likened to shift work or overtime, for example, where 33% indicates a location is open eight hours each day. Changing the model to a mixed variable problem eliminates any potential for comparison to the analytical model presented in (29).

The mixed variable problem using Formulation A is

$$\begin{aligned} \min Z &= 20y + 80x_B + 100x_D + 2000\sqrt{(0.9 - F_{0.9})} && \text{(MODEL2A)} \\ x_B, x_D &\in [0.33, 1.00] \\ y &\in \mathbb{Z}^+ \text{ and } x_B, x_D \in \mathbb{R} \end{aligned}$$

where

$$F_{0.9} = \begin{cases} F & \text{if } F \leq 0.9 \\ 0.9 & \text{if } F > 0.9. \end{cases}$$

The coefficients for the base and depot variables increased to 80 and 100, respectively, to accommodate the change in scale. The mixed variable problem using Formulation B is

$$\begin{aligned}
\min Z &= 1 - \left(\sum_{n=M}^{M+y} p_n \right) && \text{(MODEL2B)} \\
s.t. & 20y + 80x_B + 100x_D \leq 200 \\
& x_B, x_D \in [0.33, 1.00] \\
& y \in \mathbb{Z}^+ \text{ and } x_B, x_D \in \mathbb{R}.
\end{aligned}$$

Finally, the simulation model was modified to incorporate the option to use contract support at the base. Contract support is a fixed cost, but if used, the facility can repair items quicker, and fewer items need to be sent to the depot for additional repair after attempting repair at the contractor location. The validity of this option can be seen in the area of contractor support or outsourcing. The use of contractor support is represented by a binary variable w , with $w = 1$ indicating its use. The simulation model was augmented with a module representing contract support that was enabled only when the binary variable was set to one. The model formulation that includes the contract option is

$$\begin{aligned}
\min Z &= 1 - \left(\sum_{n=M}^{M+y} p_n \right) && \text{(MODEL3)} \\
s.t. & 20y + 80x_B + 100x_D + 50w \leq 240 \\
& x_B, x_D \in [0.33, 1.00] \\
& y \in \mathbb{Z}^+; x_B, x_D \in \mathbb{R}; w \in \{0, 1\}.
\end{aligned}$$

The contracting option was only incorporated into the budget constrained problem.

3.3 Simulation Model Construction and Integration

The repair model was constructed using the widely available Arena¹ discrete-event simulation software. Arena is easy to use and verify, and it allows input variables that

¹Arena[®] simulation software is a registered trademark with Rockwell Software, Inc.

can be specified by the MGPS-RS algorithm. Also, Arena is packaged with statistical tools that can be used to provide the necessary availability response for the multi-echelon repairable system.

Steady-state results for the availability are desired. However, it is also important to have a degree of noise in the availability response to properly test the algorithm. With little to no variability, the problem becomes too much like the deterministic setting. To balance both of these criteria, each simulation replication must be run for an appropriate length. The approach to determine the simulation length is similar to the one given by Chrissis and Gecan (19). Ten replications were divided into two batches of five each. The five replications of the first batch were run for a specified length of time while the other five are run for a slightly longer length. Using the tools provided in Arena's Output Analyzer, a hypothesis test to compare means of the two batches was performed. If the result of the test indicated the means were statistically equal, then the estimator for availability was formed using the replications. If the hypothesis test result showed differing means, the replication lengths were increased and the process was repeated. However, this approach became too expensive computationally to perform for each function evaluation. The method was still used to determine an appropriate length, but it was done only once to determine a target replication length. The remaining simulation runs were conducted using that replication length. The time required to perform one replication of the actual Arena simulation turned out to be negligible, compared to the cost of integrating it with the optimization code and the setup cost involved.

Integration of the simulation and the optimization algorithm was particularly difficult because neither the optimization code nor simulation model are self-contained. The NOMADm optimization software runs inside the MATLAB computing environment and makes function calls, expecting a black box function to return a response. However, the simulation model runs inside the Arena simulation software, and it does not naturally lend itself to easy integration with a MATLAB code. To overcome this problem, an interface was written in Microsoft Visual Basic to integrate the simulation and optimization code. The MATLAB code outputs the input vector and waits for a signal from Visual Basic to resume. The simulation reads the input vector, runs a replication given those parameters,

outputs the availability response, and waits for the next input vector. The simulation is not stopped at this point because the random number streams would reset, meaning subsequent identical input vectors would yield the exact same response, resulting in no variability or noise in the response.

The NOMADm software website (1) gives specifics on how to properly define the initial iterate, function evaluation, discrete neighbors, and linear/bound constraint files in the software. Before the algorithm could be implemented, certain files had to be set up which define the problem to be solved. Among them, the set of discrete neighbors was constructed to be the commonly used set given in (2.4). However, as described in Chapter 2, neighbors for categorical variables may include any possible value for the categorical variable. Since the contracting variable was binary, this definition of the neighbor set was sufficient. The linear bound/linear constraints definition was important for two reasons. The first was to ensure assumption **A1** was satisfied that the iterates lie on a compact set. Since the continuous variables x_B and x_D have both lower and upper bounds, **A1** was satisfied. The second reason is that for proper definition of the linear constraints is that the set of directions must positively span the tangent cone of the nearby linear constraints.

The MGPS-RS algorithm within the NOMADm software was applied to the multi-echelon repair models explained in this chapter. The results for each of the models are presented in the next chapter.

IV. Results

This chapter presents the results for each of the models outlined in Chapter 3. For the models that have an associated deterministic or analytical solution, a discussion on accuracy is presented. Also given is a brief examination regarding the effect on convergence of the choice of indifference zone and alpha parameters, and the associated decay rates used in the R&S procedure.

4.1 Discrete Variable Models

In the case with all discrete variables, the POLL step of the MGPS algorithm simply consists of repeated evaluations of discrete neighbor points, and the EXTENDED POLL step is empty. In the discrete neighbor search, the algorithm acts similar to a one-factor-at-a-time approach; it methodically searches the neighbors looking at each of the variables for the best improvement by increasing each subsequently by one.

The algorithm was run using both models, **MODEL1A** and **MODEL1B**, at two different levels of machines, $M = 5$ and $M = 20$. For both models with $M = 5$, complete enumeration of the feasible design space occurred. This is primarily due to a small decision space and each of the variables having a large and nearly equal effect on the objective function. However, the optimal solution was obtained in both cases showing that the noise in the simulation response was overcome. In addition to the analytical solution, the discrete model results can also be compared to the proposed particle swarm optimization approach (PSO) by Alkhamis and Ahmed (6). It should be noted that the PSO algorithm attempts to solve the original formulation (3.2) and Alkhamis and Ahmed do not provide any measure of computational time. Table 4.1 summarizes the results for both models with $M = 5$ compared with the analytical solution from (29) and PSO.

In the case with $M = 20$, the results for the discrete models differed. The budget constrained availability model, **MODEL1B**, behaved similarly to the problem with fewer machines. However, the search method was able to eliminate some inferior solutions, so the solution space was not completely enumerated. Conversely, **MODEL1A** found no improvement from the initial iterate with $M = 20$ because the polling of the discrete

Table 4.1 Model 1A/1B Results, M=5.

	y	x_B	x_D	Evals
True (Analytical)	3	2	2	-
MODEL1A	3	2	2	60
MODEL1B	3	2	2	87
PSO	3	2	2	n/a

Table 4.2 Model 1A/1B Results, M=20.

	y	x_B	x_D	Evals
True (Analytical)	8	5	6	-
MODEL1A	1	1	1	16
MODEL1B	8	5	6	760
PSO	9	4	6	n/a

neighbors failed to yield an improved solution. This can be attributed to the cost of an increase in any of the decision variables outweighing each of their contributions to the improvement in availability. A higher penalty on the deviation from the desired 0.9 availability may have remedied this problem. For example, an initial point of $\{2, 2, 2\}$ when $M = 5$ would have likely eliminated reaching $\{1, 1, 1\}$, provided the variability in the model was not too large. Table 4.2 summarizes the results for the discrete variable models with $M = 20$ compared with the respective analytical solution and PSO.

It is important to note that a better initial design vector with the variables at higher starting levels would have likely prevented some of the enumeration, since the knapsack problem with negative coefficients yields a better solution relative to an increase in the values of the design variables.

4.2 Mixed-Variable Models

Results for **MODEL2A** were more interesting. The size of the problem enabled the continuous variables to be “discretized” in intervals of 0.025, creating a complete discrete decision space with a finite number of points. A deterministic model was constructed and

Table 4.3 Model 2A Results.

	y	x_B	x_D	Z
MODEL2A	5	0.833	0.9583	412.494
Deterministic	2	0.925	0.95	269

run using all possible decision points, completely enumerating the discrete space, to give an estimated value of the objective function without the random noise. The algorithm was unable to obtain the global optimum. The results for **MODEL2A** are also in Table 4.3; the deterministic model indicates an approximate “optimal” solution.

A problem with using the penalty approach is that f is not continuously differentiable, violating the assumptions, listed in Section 3.1, necessary to ensure convergence to a stationary point. Also, the algorithm has a significant problem at the 0.9 boundary point particularly because of the random noise. Since the solution is known to have a true response of 0.9041 (just above 0.9) from the deterministic function, noise can cause a significant penalty to be assessed. There is no noise in the objective function on the lower end because, if the availability is above 0.9, the objective function is only made up of deterministic values. Obviously, the error in the objective function at points that yield availability values close to 0.9 is not normally distributed. In fact, it is only one-sided. Figure 4.1 shows the behavior of the objective function for the deterministic model in terms of the continuous variables when the number of spares is two. Notice the surface sharply flattens in the area around the optimum, indicating the availability above 0.9. Surface plots for different number of spares are located in the appendix.

The algorithm had no problem handling **MODEL2B**. Since the true objective function is linear with a single linear constraint, the objective function is simply a series of planes that are “cut-off” by a plane representing the constraint. An approximate solution was found in less than 750 function evaluations. An estimated true optimal solution was also obtained for this model using the discretized deterministic model. Table 4.4 compares the results of the deterministic model with **MODEL2B**.

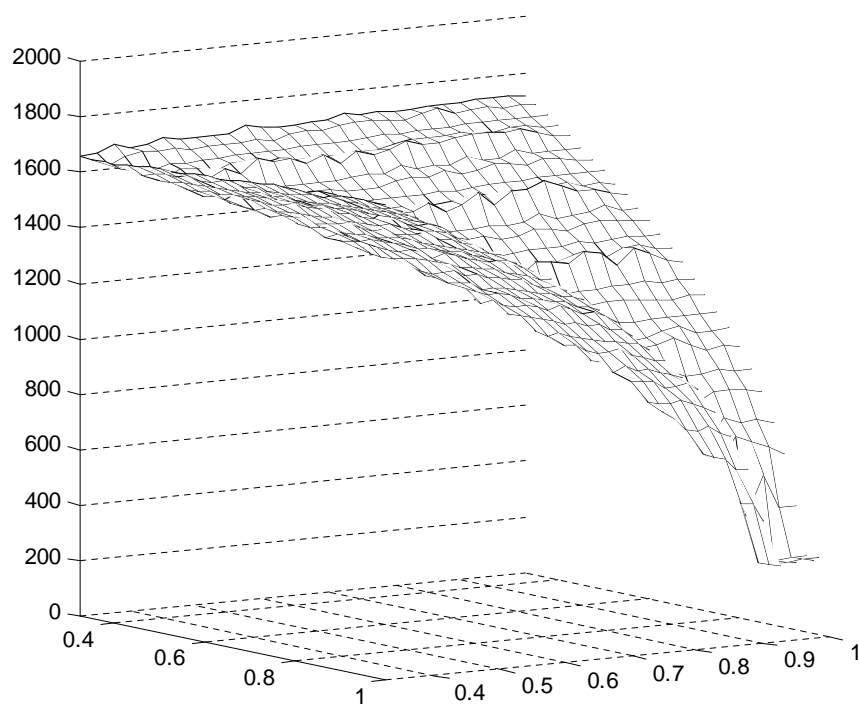


Figure 4.1 Deterministic model using Formulation A, $y = 2$.

Table 4.4 Model 2B Results.

	y	x_B	x_D	Z
MODEL2B	2	0.8864	0.8958	13.95
Deterministic	2	0.875	0.9	13.89

Table 4.5 Model 3 Results.

	y	w	x_B	x_D	Z
MODEL3	2	“Yes”	0.3333	0.8333	25.69

Results were similar for **MODEL3** since the true objective function in **MODEL3** is similar to **MODEL2B**. The algorithm seemed to handle the binary variable quite well. The indicated optimum found by the algorithm is shown in Table 4.5.

4.3 Indifference Zone

The R&S procedures used in this algorithm require the user to explicitly set the initial values for the indifference zone δ and significance level α , and the respective decay factors. Srivier (50) does not give guidance for the specific values to use, only that the values be “loose.” The reason δ must be initially started high is because this causes more samples to be required in the beginning of the algorithm when there is little information known regarding the variability of the simulation model. As additional samples are taken, the indifference zone may be reduced to identify inferior solutions. The results of changing the indifference zone value and decay rate verify this assertion. When the indifference zone or the decay rate were set very low, the algorithm yielded much poorer final solutions and required many more function evaluations. The default values used by Dunlap (24) in the NOMADm software performed well; however, this may be a coincidence with the scale of the problems used in this analysis. For **MODEL2A**, the indifference zone and the indifference zone decay were each varied. The results, shown in Table 4.6, show that a smaller initial indifference zone obtained the same solution, but took many more simulation runs. The smaller decay rate yielded worse results in the same number of simulation runs.

Table 4.6 Model 2A results using different indifference zone parameters.

MODEL2A	Indifference Zone	IZ Decay Rate	Z	Evaluations
Baseline	100	0.95	412.49	284
Smaller Indifference Zone	50	0.95	406.24	500*
Lower IZ Decay	100	0.75	1028.87	500*

*Note: The number of function evaluations was capped at 500.

Table 4.7 Model 2B results using different indifference zone values.

MODEL2B	Indifference Zone	Z	Evaluations
Baseline	100	25.73	215
Smaller Indifference Zone	10	26.53	353

A similar result was obtained for **MODEL2B** using a smaller indifference zone. Table 4.7 shows the results.

Finally, it should be noted that the computational limits in this study were not in the optimization code nor the simulation runs. The majority of the overhead came from the integration and transfer of data between the optimizer and the simulator. The advantages of using an efficient ranking and selection procedure are usually concerned with sampling cost, the number of simulations runs necessary to produce a “best” solution from a set of candidates. However, sampling cost can sometimes be outweighed by switching cost, or the cost of switching simulation configurations. Sriver included the switching cost in his analysis of R&S procedures. He showed that switching in the SSM algorithm can be quite significant, “requiring more switches than SAS by approximately two orders of magnitude on each of the test problems” in Sriver’s computational results (50).

The results presented in this chapter show reasonable results for the use of the MGPS-RS algorithm for simulation optimization. The next chapter provides some conclusions of this study, as well as recommendations for future research in this area.

V. Conclusions and Recommendations

This chapter summarizes the results given in Chapter 4 and gives recommendations for future research. Possible future work includes continued analysis of the algorithm and research on integration issues between simulation and optimization applications.

5.1 Conclusions

This thesis provided a framework for applying the MGPS-RS algorithm to a simulation model representing a real-world system. The results on the simple two-echelon model show promise that the algorithm applies well to simulation optimization. The algorithm was able to handle random noise in the response, moving towards improving solutions. Additional findings show the selection of indifference zone parameters may have a significant effect on the number of simulation runs needed to produce a solution and on the quality of that solution. The research also provided possible pitfalls to the application of the algorithm to an actual simulation environment. Simulations are not often written in the software that contain robust optimization algorithms, so the need for software integration becomes critical, particularly when the number of simulation runs is large. Finally, the NOMADm software, the only software that implements the MGPS-RS algorithm, was effectively integrated with an external simulation application.

5.2 Future Algorithm Research

The results of this study highlight some items for future research. The first is to develop a more formal design and analysis than used in this thesis. This thesis provides only an initial study regarding the use of this algorithm for optimization via simulation. Additional follow-on computational studies should have a formal experimental design structure. Also, a study on a larger more complex model may yield additional insight into the performance, limitations, and abilities of the algorithm. The results from the tests on varying indifference zone parameters warrants an examination of the user-defined R&S parameters, to include the choice of indifference zone and its decay rate. A way to balance sampling and switching cost could improve the performance of the algorithm, especially when inte-

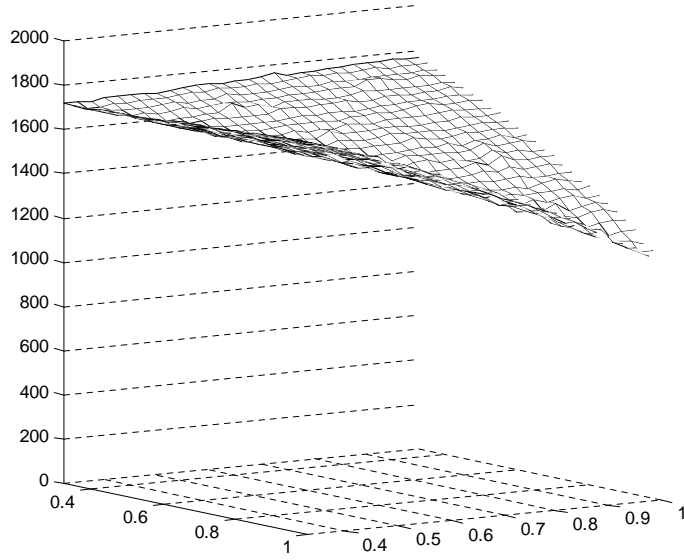
gration becomes a problem. As Sriver suggested (50), a simple modification may be to implement the minimum switching sequential procedure, developed by Hong and Nelson (31) as a sequential sampling technique that uses the same number of switches as two-stage procedures, and determine if its performance warrants inclusion in the algorithm. With less switching of the input parameters, the simulation could run batches of replications before returning to the optimization procedure.

5.3 *Future Integration Research*

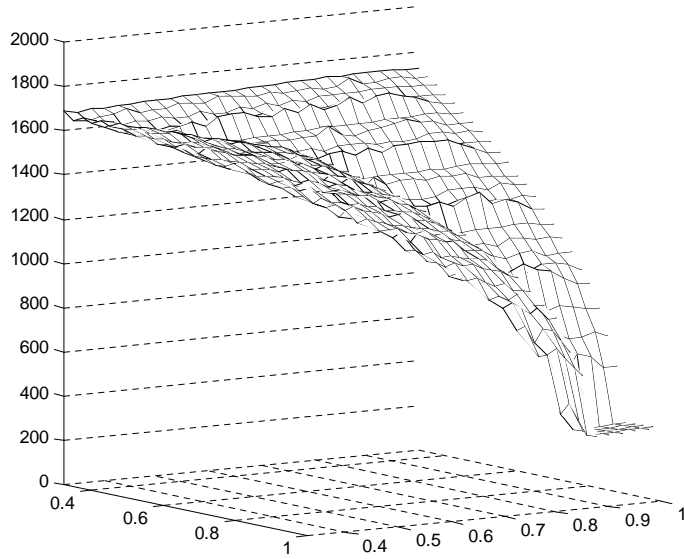
Since the integration of the optimization code and the simulation model provided the greatest cost in the simulation optimization procedure, future research should focus on mitigating this cost. Sriver discusses switching cost in his analysis of three procedures considered for implementation in MGPS-RS (50), where Rinott's procedure incurs no switching cost, SAS calls for a single switch for each candidate, and SSM requires a switch each time an additional sample is needed in the ranking and selection procedure. Because SSM performed well with GPS methods in controlling the number of required function evaluations while guaranteeing the proper level of confidence in the iterate selection, Dunlap (24) incorporated SSM into NOMADm. For the software to be effective in situations where the switching cost is more expensive than sampling cost, as in this thesis, an alternate procedure could be considered. Finally, consideration should be given to incorporating the optimization code and the simulation model into the same software application. For example, Boesel and Nelson (15) interface their genetic algorithm using ranking and selection technique with AweSim! simulation software for seamless operation. Since MGPS-RS is already encoded into the NOMADm MATLAB software, simulations using Simulink/SimEvents packages from MathWorks might be considered. However, since these MATLAB packages are not as widely available, the algorithm, if continued testing shows positive results, may be considered to replace heuristics in more popular simulation software.

VI. Appendix

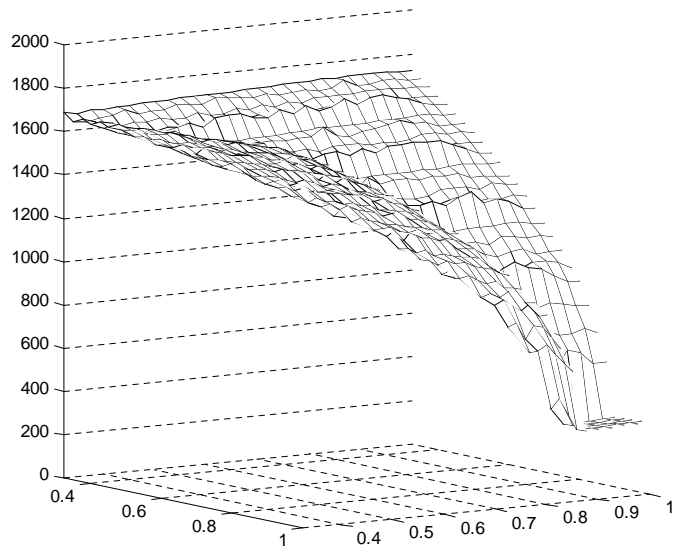
This appendix shows surface plots for the objective function behavior with no noise for **MODEL2A**.



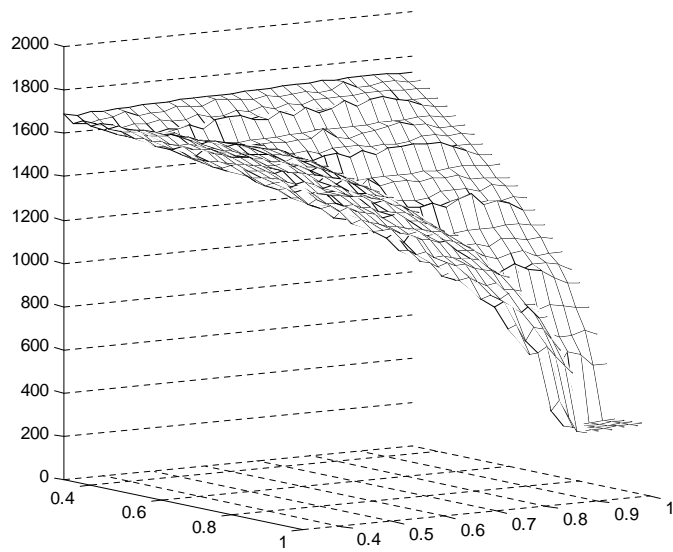
Deterministic Model2A, Number of Spares = 1



Deterministic Model2A, Number of Spares = 3



Deterministic Model2A, Number of Spares = 4



Deterministic Model2A, Number of Spares = 5

Bibliography

1. ABRAMSON, M. A. Nonlinear optimization with mixed variables and derivatives-matlab (NOMADm). Software. Available from Mark Abramson's NOMADm Page, <http://www.afit.edu/en/ENC/Faculty/MAbramson/nomadm.html>.
2. ABRAMSON, M. A. *Pattern Search Algorithms for Mixed Variable General Constrained Optimization Problems*. PhD thesis, Rice University, Department of Computational and Applied Mathematics, 2002. also appears as Tech. Rep. TR02-11.
3. ABRAMSON, M. A. Second order behavior of pattern search algorithms. *SIAM Journal on Optimization* 16, 2 (2005), 515-530.
4. ABRAMSON, MARK A. Mixed variable optimization of a load-bearing thermal insulation system using a filter pattern search algorithm. *Optimization and Engineering* 5, 2 (2004), 157-177.
5. ABRAMSON, M. A., AND AUDET, C. Convergence of mesh adaptive direct search to second-order stationary points. *SIAM Journal on Optimization* (to appear).
6. ALKHAMIS, T. M., AND AHMED, M. A. Simulation-based optimization for repairable systems using particle swarm algorithm. In *Proceedings of the 2005 Winter Simulation Conference* (Piscataway, New Jersey, 2005), M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, Eds., Institute of Electrical and Electronics Engineers, pp. 857-861.
7. AHMED, M. A., ALKHAMIS, T. M., AND HASAN, M. Optimizing discrete stochastic systems using simulated annealing and simulation. *Computers and Industrial Engineering* 4 (1997), 823-836.
8. AHMED, M. A., AND ALKHAMIS, T. M. Simulation-based optimization using simulated annealing with ranking and selection. *Computers and Operations Research* 29 (2002), 387-402.
9. ANDRADOTTIR, S. A review of simulation optimization techniques. In *Proceedings of the 1998 Winter Simulation Conference* (Piscataway, New Jersey, 1998), D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, Eds., Institute of Electrical and Electronics Engineers, pp. 151-158.
10. AUDET, C., AND DENNIS, JR., J. E. Pattern search algorithms for mixed variable programming. *SIAM Journal on Optimization* 11, 3 (2000), 573-594.
11. AUDET, C., AND DENNIS, JR., J. E. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization* 16 (2004), (to appear). also appears as Tech. Rep. TR04-02, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, January 2004.
12. AUDET, C., AND DENNIS, JR., J. E. A pattern search filter method for nonlinear programming without derivatives. *SIAM Journal on Optimization* 14, 4 (2004), 980-1010.

13. AZADIVAR, F. Simulation optimization methodologies. In *Proceedings of the 1999 Winter Simulation Conference* (Piscataway, New Jersey, 1999), P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, Eds., Institute of Electrical and Electronics Engineers, pp. 93-100.
14. BECHOFER, R. E. A single-sample multiple decision procedure for ranking means of normal populations with known variances. *Annals of Mathematical Statistics* 25 (1954), 16-39.
15. BOESEL, J., NELSON, B. L., AND ISHII, N. A framework for simulation-optimization software. *IIE Transactions* 35 (2003), 221-229.
16. BOESEL, J., NELSON, B. L., AND KIM, S. H. Using ranking and selection to 'cleanup' after simulation optimization. *Operations Research* 51 (2003), 814-825.
17. BOOKER, A. J., DENNIS, JR., J. E., FRANK, P. D., SERAFINI, D. B., TORCZON, V., AND TROSSET, M. W. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization* 17, 1 (1999), 1-13.
18. BOX, G. E. P. Evolutionary operation: A method for increasing industrial productivity. *Applied Statistics* 6 (1957), 81-101.
19. CHRISISSIS, JAMES W., AND GECAN, ANTON S. Multi-echelon system design via simulation. *Simulation* 47, 6 (1986), 240-243.
20. CLARKE, F. H. *Optimization and Nonsmooth Analysis*. SIAM Classics in Applied Mathematics 5. SIAM, Philadelphia, 1990.
21. CONN, A. R., GOULD, N. I. M., AND TOINT, P. L. A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis* 28, 2 (1991), 545-572.
22. DAVIS, C. Theory of positive linear dependence. *American Journal of Mathematics* 76, 4 (1954), 733-746.
23. DUDEWICZ, E. J., AND DALAL, S. R. Allocation of observations in ranking and selection. *The Indian Journal of Statistics* 37B, 1 (1975), 28-78.
24. DUNLAP, J. E. *On the Use of Surrogate Functions for Mixed Variable Optimization of Simulated Systems*. Masters thesis, Air Force Institute of Technology, Graduate School of Engineering and Management, Wright-Patterson AFB, Ohio, 2005.
25. FLETCHER, R., AND LEYFFER, S. Nonlinear programming without a penalty function. *Mathematical Programming* 91, 2 (2002), 239-269.
26. FU, M. C. Optimization via simulation: A review. *Annals of Operations Research* 53 (1994), 199-247.
27. FU, M. C. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing* 14, 3 (2002), 192-215.
28. GOLDSMAN, D., AND NELSON, B. L. Comparing systems via simulation (chap. 8). In *Handbook of Simulation* (New York, 1998), J. Banks, Ed., John Wiley and Sons, pp. 273-306.

29. GROSS, D., MILLER, D. R., AND SOLAND, R. M. A closed queueing network model for multi-echelon repairable item provisioning. *IIE Transactions* 15, 4 (1983), 344-352.
30. HEDLUND, H. E., AND MOLLAGHASEMI, M. A genetic algorithm and an indifference-zone ranking and selection framework for simulation optimization. In *Proceedings of the 2001 Winter Simulation Conference* (Piscataway, New Jersey, 2001), B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, Eds., Institute of Electrical and Electronics Engineers, pp. 417-421.
31. HONG, L. J., AND NELSON, B. L. An indifference-zone selection procedure with minimum switching and sequential sampling. In *Proceedings of the 2003 Winter Simulation Conference* (Piscataway, New Jersey, 2003), S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, Eds., Institute of Electrical and Electronics Engineers, pp. 474-480.
32. JACOBSON, S. H., AND SCHRUBEN, L. W. Techniques for simulation response optimization. *Operations Research Letters* 8 (1989), 1-9.
33. Kim, S. H., and Nelson, B. L. A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation* 11, 3 (2001), 251-273.
34. KOCHER, P., AND NIELANDER, U. Simulation-based optimisation of multi-echelon systems. *International Journal of Production Economics* 93-94 (2005), 505-513.
35. KOKKOLARAS, M., AUDET, C., AND DENNIS, JR., J. E. Mixed variable optimization of the number and composition of heat intercepts in a thermal insulation system, *Optimization and Engineering* 2, 1 (2001), pp. 5-29.
36. LEWIS, R. M., AND TORCZON, V. Rank ordering and positive bases in pattern search algorithms. Tech. Rep. ICASE 96-71, NASA Langley Research Center, 1996.
37. LEWIS, R. M., AND TORCZON, V. Pattern search algorithms for bound constrained minimization. *SIAM Journal on Optimization* 9, 4 (1999), 1082-1099.
38. LEWIS, R. M., AND TORCZON, V. Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization* 10, 3 (2000), 917-941.
39. LEWIS, R. M., AND TORCZON, V. A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Optimization* 12, 4 (2002), 1075-1089.
40. LUCIDI, S., PICCIALI, V. , AND SCIANDRONE, M. An Algorithm Model for Mixed Variable Programming. *SIAM Journal on Optimization* 15, 4 (2005), 1057-1084.
41. NELSON, B. L., SWANN, J., GOLDSMAN, D., AND SONG, W. Simple procedures for selecting the best simulated system when the number of alternatives is large. *Operations Research* 49, 6 (2001), 950-963.
42. OLAFSSON, S. Iterative ranking-and-selection for large-scale optimization. In *Proceedings of the 1999 Winter Simulation Conference* (1999), P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, Eds., pp. 479-485.

43. OUALI, M. S., AOUJIT, H., AND AUDET, C. Optimisation des stratégies de maintenance intégration à la production. *Journal Européen des Systèmes Automatisés* 37, 5 (2003), 587-605.
44. PICHITLAMKEN, J., AND NELSON, B. L. Selection-of-the-best procedures for optimization via simulation. In *Proceedings of the 2001 Winter Simulation Conference* (Piscataway, New Jersey, 2001), B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, Eds., Institute of Electrical and Electronics Engineers, pp. 401-407.
45. PICHITLAMKEN, J., AND NELSON, B. L. A combined procedure for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation* 13, 2 (2003), 155-179.
46. RINOTT, Y. On two-stage selection procedures and related probability-inequalities. *Communications in Statistics A7*, 8 (1978), 799-811.
47. SHERBROOKE, C. C. METRIC: A multi-echelon technique for recoverable item control. *Operations Research* 16 (1968), 122-141.
48. SHERBROOKE, C. C. *Optimal Inventory Modeling of Systems: Multi-Echelon Techniques*. John Wiley and Sons, New York, New York, 2003.
49. SPALL, J. C. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. John Wiley and Sons, Hoboken, New Jersey, 2003.
50. SRIVER, T. A. *Pattern Search Ranking and Selection Algorithms for Mixed-Variable Optimization of Stochastic Systems*. PhD thesis, Air Force Institute of Technology, Graduate School of Engineering and Management, Wright-Patterson AFB, Ohio, 2004.
51. SRIVER, T. A., AND CHRISSIS, J. W. Combined pattern search and ranking and selection for simulation optimization. In *Proceedings of the 2004 Winter Simulation Conference* (Piscataway NJ, 2004), R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, Eds., Institute of Electrical and Electronics Engineers, pp. 645-653.
52. SRIVER, T. A., CHRISSIS, J. W., AND ABRAMSON, M. A. Pattern search ranking and selection algorithms for mixed variable stochastic programming. *Mathematical Programming* (2005), (to appear).
53. SWISHER, J. R., JACOBSON, S. H., AND YUCESAN, E. Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey. *ACM Transactions on Modeling and Computer Simulation* 13, 2 (2003), 134-154.
54. TORCZON, V. On the convergence of pattern search algorithms. *SIAM Journal on Optimization* 7, 1 (1997), 1-25.
55. TROSSET, M. W. On the use of direct search methods for stochastic optimization. Tech. Rep. TR00-20, Department of Computational and Applied Mathematics, Rice University, Houston Texas, June 2000.
56. WINSTON, WAYNE L. *Operations Research: Algorithms and Applications*. Thomsen Publishing, New York, 2004.

Vita

Captain Derek Tharaldson was born in Edina, Minnesota and raised in Lakeville, Minnesota. He graduated from Lakeville Senior High School in 1994 and was selected to attend the United States Air Force Academy in Colorado Springs, Colorado. He graduated in 1998 with a Bachelor of Science degree in Operations Research and was commissioned in the United States Air Force on 27 May 1998.

In his first assignment, Captain Tharaldson served as an officer personnel policy analyst and Chief of Officer Promotions Analysis in the Air Force Personnel Operations Agency at the Pentagon. From February 2002 to August 2004, Captain Tharaldson served as Chief of Database Analysis and Operations for the 83d Fighter Weapons Squadron at Tyndall Air Force Base, Florida. In August 2004, he was reassigned to the Air Force Institute of Technology (AFIT) at Wright-Patterson Air Force Base, Ohio to work on a master of science degree in Operations Research. Upon graduation in March 2006, Captain Tharaldson will be reassigned to the Dean of Faculty, Department of Economics and Geography, at the United States Air Force Academy.

Permanent address: Lakeville, MN

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 23-03-2004		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From - To) Aug 2005 - Mar 2006	
4. TITLE AND SUBTITLE OPTIMIZATION OF A MULTI-ECHELON REPAIR SYSTEM VIA GENERALIZED PATTERN SEARCH WITH RANKING AND SELECTION: A COMPUTATIONAL STUDY				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Tharaldson, Derek D., Captain, USAF				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Street, Building 642 WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GOR/ENS/06-18	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <p>With increasing developments in computer technology and available software, simulation is becoming a widely used tool to model, analyze, and improve a real world system or process. However, simulation in itself is not an optimization approach. Common optimization procedures require either an explicit mathematical formulation or numerous function evaluations at improving iterative points. Mathematical formulation is generally impossible for problems where simulation is relevant, which are characteristically the types of problems that arise in practical applications. Further complicating matters is the variability in the simulation response which can cause problems in iterative techniques using the simulation model as a function generator.</p> <p>The mixed-variable generalized pattern search with ranking and selection (MGPS-RS) algorithm for stochastic response problems is applied to an external simulation model, by means of the NOMADm MATLAB[®] software package. Numerical results are provided for several configurations of a simulation model representing a multi-echelon repairable problem containing discrete, continuous, and categorical variables. Computational experience results are presented.</p>					
15. SUBJECT TERMS Simulation Optimization, Pattern Search, Ranking and Selection, Stochastic Optimization, Mixed Variable Programming					
16. SECURITY CLASSIFICATION OF:		17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT	b. ABSTRACT			c. THIS PAGE	Dr. Jim Chrissis (ENS)
U	U	U	UU	55	19b. TELEPHONE NUMBER (Include area code) (937) 255-3636, ext 4606; e-mail: james.chrissis@afit.edu