Nonparametric Metamodeling for Simulation Optimization

by

Anthony C. Keys

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Management Science

APPROVED:

Loren Paul Rees, Chairman

Daniel G. Conway

Cliff T. Ragsdale

Jerry R. Rakes

Robert T. Sumichrast

April 19, 1995
Blacksburg, Virginia

Keywords: Global Search, Kernel Regression, Spline Regression, Expert System, Stochastic Search
Nonparametric Metamodeling for Simulation Optimization
by
Anthony C. Keys
Loren Paul Rees, Chairman
Management Science
(ABSTRACT)

Optimization of simulation model performance requires finding the values of the model's controllable inputs that optimize a chosen model response. Responses are usually stochastic in nature, and the cost of simulation model runs is high. The literature suggests the use of metamodels to synthesize the response surface using sample data. In particular, nonparametric regression is proposed as a useful tool in the global optimization of a response surface. As the general simulation optimization problem is very difficult and requires expertise from a number of fields, there is a growing consensus in the literature that a knowledge-based approach to solving simulation optimization problems is required. This dissertation examines the relative performance of the principal nonparametric techniques, spline and kernel smoothing, and subsequently addresses the issues involved in implementing the techniques in a knowledge-based simulation optimization system.

The dissertation consists of two parts. In the first part, a full factorial experiment is carried out to compare the performance of kernel and spline smoothing on a number of measures when modeling a varied set of surfaces using a range of small sample sizes. In the second part, nonparametric metamodeling techniques are placed in a taxonomy of stochastic search procedures for simulation optimization and a method for their implementation in a knowledge-based system is presented. A sequential design procedure is developed that allows spline smoothing to be used as a search technique. Throughout the dissertation, a two-input, single-response model is considered.

Results from the experiment show that spline smoothing is superior to constant-bandwidth kernel smoothing in fitting the response. Kernel smoothing is shown to be more accurate in placing optima in X-space for sample sizes up to 36. Inventory model examples are used to illustrate the results. The taxonomy implies that search procedures can be chosen initially using the parameters of the problem. A process that allows for selection of a search technique and its subsequent evaluation for further use or for substitution of another search technique is given. The success of a sequential design method for spline smooths in finding a global optimum is demonstrated using a bimodal response surface.
ACKNOWLEDGMENTS

I would like to thank the chairman of my committee, Professor Loren Paul Rees, for teaching me a great deal about the research and publishing process within a collegiate and congenial atmosphere.

I would like to thank Dr. Daniel G. Conway, Dr. Terry R. Rakes, Dr. Cliff T. Ragsdale and Dr. Robert T. Sumichrast for serving on my committee and for their advice.

I would like to thank Professor Bernard W. Taylor III, Head of the Department of Management Science, for the financial support and facilities provided during my graduate studies.
# TABLE OF CONTENTS

Chapter 1:

<table>
<thead>
<tr>
<th>INTRODUCTION</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation</td>
<td>1</td>
</tr>
<tr>
<td>Simulation optimization</td>
<td>2</td>
</tr>
<tr>
<td>Approaches to the solution of the simulation optimization problem</td>
<td>3</td>
</tr>
<tr>
<td>Nonparametric-regression based approaches to simulation optimization</td>
<td>6</td>
</tr>
<tr>
<td>Knowledge-based approaches to simulation optimization</td>
<td>9</td>
</tr>
<tr>
<td>Purpose of the research</td>
<td>10</td>
</tr>
<tr>
<td>Scope and limitations</td>
<td>11</td>
</tr>
<tr>
<td>Plan of presentation</td>
<td>11</td>
</tr>
</tbody>
</table>

Chapter 2:

<table>
<thead>
<tr>
<th>LITERATURE REVIEW</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonparametric regression</td>
<td>12</td>
</tr>
<tr>
<td>Kernel smoothing</td>
<td>12</td>
</tr>
<tr>
<td>Spline smoothing</td>
<td>16</td>
</tr>
<tr>
<td>Simulation optimization</td>
<td>19</td>
</tr>
<tr>
<td>Knowledge-based approaches to simulation optimization</td>
<td>23</td>
</tr>
</tbody>
</table>
Chapter 3:

A PERFORMANCE BASED STUDY OF NONPARAMETRIC METAMODELING TECHNIQUES FOR THE CLASSIFICATION OF RESPONSE SURFACES IN SIMULATION OPTIMIZATION ........................................ 27

Introduction ........................................................................................................... 27
The simulation optimization problem ...................................................................... 27
Approaches to the simulation optimization problem .............................................. 28
Review of metamodeling techniques ...................................................................... 29
Metamodeling and the Crouch, Greenwood and Rees Classifier ................................ 33
Research questions .............................................................................................. 35
Methodology ........................................................................................................ 35
Design of the experiment: factors ........................................................................ 36
Design of the experiment: ANOVA model ............................................................ 44
Design of the experiment: metamodel performance measures ............................... 45
Results ................................................................................................................ 50
Computation ........................................................................................................ 51
Statistical analysis .............................................................................................. 51
Interpretation of the results ................................................................................ 56
Implications of the results for a knowledge-based approach to simulation optimization ........................................................................................................ 59
Examples using simulation models ....................................................................... 59
Example 1 ........................................................................................................... 59
Example 2 ........................................................................................................... 61
Conclusions ......................................................................................................... 65
Future Research ................................................................................................. 70
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPLEMENTING NONPARAMETRIC METAMODELING TECHNIQUES</td>
<td>71</td>
</tr>
<tr>
<td>IN A CLASSIFIER KNOWLEDGE-BASED SIMULATION OPTIMIZATION SYSTEM ..........</td>
<td>71</td>
</tr>
<tr>
<td>Introduction</td>
<td>71</td>
</tr>
<tr>
<td>Simulation optimization and the classifier knowledge-based approach</td>
<td>71</td>
</tr>
<tr>
<td>Nonparametric methods</td>
<td>72</td>
</tr>
<tr>
<td>The Classifier</td>
<td>73</td>
</tr>
<tr>
<td>Nonparametric techniques</td>
<td>75</td>
</tr>
<tr>
<td>Kernel smoothing</td>
<td>76</td>
</tr>
<tr>
<td>Spline smoothing</td>
<td>76</td>
</tr>
<tr>
<td>Research questions</td>
<td>77</td>
</tr>
<tr>
<td>Methodology</td>
<td>78</td>
</tr>
<tr>
<td>A Classify-Then-Search architecture to incorporate nonparametric</td>
<td>78</td>
</tr>
<tr>
<td>metamodeling</td>
<td></td>
</tr>
<tr>
<td>Knowledge concerning nonparametric metamodels</td>
<td>82</td>
</tr>
<tr>
<td>Experimental design for nonparametric metamodels</td>
<td>83</td>
</tr>
<tr>
<td>The accuracy and precision of nonparametric metamodels</td>
<td>83</td>
</tr>
<tr>
<td>Implementing spline smoothing in the Classify-Then-Search process</td>
<td>84</td>
</tr>
<tr>
<td>Search selection</td>
<td>84</td>
</tr>
<tr>
<td>Initialization of the search</td>
<td>84</td>
</tr>
<tr>
<td>An iteration of the search process</td>
<td>85</td>
</tr>
<tr>
<td>Updating the characteristics of the problem</td>
<td>87</td>
</tr>
<tr>
<td>Example</td>
<td>88</td>
</tr>
<tr>
<td>Conclusions</td>
<td>96</td>
</tr>
<tr>
<td>Future research</td>
<td>97</td>
</tr>
</tbody>
</table>
Chapter 5:

CONCLUSIONS AND FUTURE WORK ................................................................. 99
  Conclusions ................................................................................................. 99
  Future research ............................................................................................ 102

BIBLIOGRAPHY ............................................................................................... 104

VITA .................................................................................................................... 109
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>Example of response surface for two input, single response model</td>
<td>4</td>
</tr>
<tr>
<td>Figure 1.2</td>
<td>Example of weight function for estimating response at X = 0.2 with a bandwidth of 0.4</td>
<td>7</td>
</tr>
<tr>
<td>Figure 1.3</td>
<td>Example of kernel weight function showing assignment of weights to responses covered by a bandwidth of 0.4 at X = 0.2</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2.1</td>
<td>Changes in Rice boundary weight functions as x approaches the right boundary: Epanechnikov kernel, bandwidth = 4</td>
<td>15</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Architecture of an expert system (Rolston, 1988)</td>
<td>24</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>The Classify-Then-Search process (Crouch, Greenwood and Rees, 1993)</td>
<td>30</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Deterministic function, unimodal surface</td>
<td>39</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Deterministic function, bimodal surface</td>
<td>40</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>Deterministic function, tetramodal surface</td>
<td>41</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>Geometric illustration of the components involved in performance measure M</td>
<td>46</td>
</tr>
<tr>
<td>Figure 3.6</td>
<td>Metamodel comparison</td>
<td>48</td>
</tr>
<tr>
<td>Figure 3.7</td>
<td>Scatter plot of raw data from example 1</td>
<td>62</td>
</tr>
<tr>
<td>Figure 3.8</td>
<td>Spline smooth metamodel of data from example 1</td>
<td>63</td>
</tr>
<tr>
<td>Figure 3.9</td>
<td>Kernel smooth metamodel of data from example 1</td>
<td>64</td>
</tr>
<tr>
<td>Figure 3.10</td>
<td>Scatter plot of raw data from example 2</td>
<td>66</td>
</tr>
<tr>
<td>Figure 3.11</td>
<td>Kernel smooth metamodel of data from example 2</td>
<td>67</td>
</tr>
<tr>
<td>Figure 3.12</td>
<td>Spline smooth metamodel of data from example 2</td>
<td>68</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>The Classify-Then-Search process (Crouch, Greenwood and Rees, 1993)</td>
<td>74</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>The taxonomy of stochastic search techniques</td>
<td>79</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Classification and search process</td>
<td>80</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Contour plot of function with no added error</td>
<td>90</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Spline smooth metamodel using initial design of 16 runs</td>
<td>91</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>Spline smooth metamodel using a total of 24 runs</td>
<td>92</td>
</tr>
<tr>
<td>Figure 4.7</td>
<td>Spline smooth metamodel using a total of 25 runs</td>
<td>93</td>
</tr>
<tr>
<td>Figure 4.8</td>
<td>Spline smooth metamodel at termination of the sequential design process, 33 runs used</td>
<td>94</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 3.1:  Factors and levels for the experimental design .......................................................... 37
Table 3.2:  Variance models ........................................................................................................ 42
Table 3.3:  Tables showing absolute performance results
            for spline and kernel metamodeling techniques .............................................................. 53
Table 3.4:  Tables showing relative performance results
            for spline and kernel metamodeling techniques .............................................................. 54
Table 3.4 cont'd. ......................................................................................................................... 55
CHAPTER 1

INTRODUCTION

SIMULATION

Many simple systems can be described or modeled in terms of mathematical equations. To permit this modeling to take place, the mathematical description usually imposes some restrictive assumptions on the system which make the model limited in its application. For example, the period of a simple pendulum is given by

$$T = 2\pi \sqrt{\frac{s}{g}}$$

where $s$ is the length of the string from the point of suspension to the center of mass of the pendulum bob, and $g$ is the gravitational acceleration. However, this model assumes that the bob oscillates only a few degrees either side of its resting position, the pendulum resides in a vacuum and the string is inextensible. In the presence of air, it must be assumed that the mass of the bob is concentrated at a point and that the string is infinitesimally thin to avoid the effects of air resistance. The model is a reasonably good predictor of the period of the pendulum as long as the assumptions are met. In business, systems are orders of magnitude more complicated than the pendulum. The behavior of business systems is often probabilistic, not deterministic, and any assumptions made carry no guarantee that they will be honored. Attempting to model business systems with equation-based models often results in an intractable problem, especially in systems where feedback exists and there are many nonlinear relationships. In many cases, simulation is the only tool that can be applied to study a business system.

A simulation model is most often a computer program written to run on a digital computer. The program contains an appropriate level of detail of the business process so that the variables of interest can be studied. As certain parts of the business process may have variability in them, the computer program uses random number generators to model the probabilistic nature of those parts. Usually, a particular set of system parameter values is of interest. The parameters that relate to the parts of the business process that the manager has the ability to change are termed the controllable variables. Other parameters that are beyond the ability or control of the manager are termed the uncontrollable variables. The probabilistic behavior of certain processes is also beyond the control of the manager: this behavior gives rise to uncertainty in the
values of the system responses the manager may be interested in. A run of a simulation model involves setting the controllable parameters, running the simulation model and observing the output of interest when the run terminates. Outputs are often averages or time-weighted averages of some measure of performance of the business process.

The simulation model may be used to study a business process in several different ways. First, it may be used to determine the values of an output for a given setting of the controllable variables. Runs may be repeated at the same setting of the controllable variables to obtain information about the variability of the response. Second, the simulation model may be used to compare the performance of the process at two or more settings of the controllable variables. Additionally, the simulation model may be used in an experiment to determine the settings that maximize or minimize a response or responses - this activity is termed simulation optimization.

SIMULATION OPTIMIZATION

The problem of determining the set of controllable values for a simulation model that optimizes a response is difficult. To see this, consider probable courses of action that one might take to solve the simulation optimization problem. One course of action might be to run the simulation model at settings of the controllable variables that cover the region over which the model is valid as densely as possible then record the responses at each setting and compare them to find the optimum. This naive approach neglects the possibility that the response may be stochastic, so that a simple comparison of response values will not necessarily result in the best settings of the inputs being chosen. Considering carrying out replications at each setting to obtain a mean response and a variance estimate neglects to take into account the time/expense of running simulation models. This approach may be justifiable if the time or cost of making a simulation run is very small, but the costs would be prohibitive if simulation runs were expensive and/or time consuming. It can be concluded that the simulation optimization problem will usually require the use of parsimonious techniques for its solution. To enable a choice of an efficient solution technique, it is necessary to carefully define the problem.

Let \( X = (X_1, X_2, \ldots, X_n) \) be a vector of \( n \) controllable variables that are used to produce \( Y = (Y_1, Y_2, \ldots, Y_m) \), a vector of \( m \) responses. Let the constraints on the values of \( X \) define a feasible region \( S \). Then the simulation optimization problem is defined as the following constrained optimization problem:
Optimize: \[ E(Y) \text{ over the region } S \subseteq \mathbb{R}^n, \]

where the domain of S may be either continuous, discrete or mixed and
\[ X = (X_1, X_2, \ldots, X_n) \in S \]

Subject to: \[ g_i(X) \leq 0 \]

where \( i = (1, 2, \ldots, n, n+1, \ldots, n+q) \) and
\[ g_i(X) = (\gamma_i < X_i < \theta_i), i = 1, 2, \ldots, n \]
\[ g_i(X) = (\gamma_i < f(X) < \theta_i), i = n+1, n+2, \ldots, n+q \]

where there are \( q \) constraints involving more than one variable

Given that the characteristics of the problem are so varied and that each set of characteristics may demand a separate solution approach, this dissertation will investigate only those problems where the controllable variables and the response are continuous. This can include problems where the discrete nature of the variables is such that their behavior can be approximated by representing them as continuous variables, for example, if X represents the number of units produced and ranges between 5,000 and 10,000.

**APPROACHES TO THE SOLUTION OF THE SIMULATION OPTIMIZATION PROBLEM**

Generally, solution approaches to the problem are selected according to the nature and state of knowledge of the objective function (Zhigljavsky, 1991). Unfortunately, little is often known about the nature of the response, making the choice of a solution method problematic. A key concept in the solution procedures is the depiction of the response function as a response surface. Momentarily consider the case of a problem with two inputs and a single response. Each run of a simulation model provides a response value at the two values of the controllable variables. These values can be used as co-ordinates and plotted on a set of orthogonal axes. The controllable variables \( X_1 \) and \( X_2 \) define a point on a plane, with the response value then defining a height above that plane. An infinite set of runs of the simulation model would give a response surface, comprising the locus of the expected values of the plotted points. Figure 1.1 depicts an example of such a response surface. Response surfaces are not so easily visualized in higher dimensions, but the concept is still very important for discussing solution procedures, their advantages and their disadvantages. Throughout this section, the two input, single response model will be used to illustrate and discuss various approaches. The discussion of solution procedures is restricted to those applicable to problems with continuous, real-valued variables.
Figure 1.1: Example of response surface for two input, single response model
Methods of optimizing simulations can be pragmatically divided into those methods which require modification of the code used in the simulation model (invasive techniques) and those that regard the simulation model as a black-box system (non-invasive). Meketon (1987) discusses and compares invasive techniques such as Stochastic Approximation and Perturbation Analysis. The attraction of these techniques is their ability to obtain information for optimization during a single long run of the simulation model. However, the techniques impose conditions on the behavior of the simulation model such as:

1. The simulation response must constitute a regenerative process.
2. The sequence of events in the simulation model must be unchanged for some small change in one of the parameters.

The techniques also impose a heavier burden on the simulation model development process as the developers must be expert with both simulation model development and the optimization algorithm (Azadivar, 1992).

Jacobsen and Schruben (1989) categorize simulation optimization approaches by search method into path, pattern, random, and integral techniques. Path techniques use gradient information and include sophisticated methodologies such as Response Surface Methodology (RSM), while pattern searches do not use gradient information. Random methods use random assignment of values to variables in the planning of simulation runs, and integral methods attempt global optimization through space covering and elimination of least likely regions. Jacobsen and Schruben conclude that no approach is generally applicable, as the path, pattern and random approaches are local optimizers and the integral methods suffer from slow convergence and difficulties in establishing required constants. Barton (1992, 1994) reviews the non-invasive techniques that construct metamodels. The idea behind metamodeling is to model the response surface by estimating a response function based on a small sample of simulation runs. The metamodeling techniques can be classified into traditional and non-traditional approaches. RSM dominates the traditional approaches, while spline, kernel, radial basis function, spatial correlation and frequency domain basis functions comprise the non-traditional approaches.

RSM is described in great detail in a number of texts including Box and Draper (1987), Myers (1976), and Khuri and Cornell (1987). RSM combines experimental design and gradient-based search to tackle the problem of stochastic response. Initially, a first-order experimental design allows a local linear model to be built which can give a good estimate of the direction of the maximum gradient. The search then steps out, evaluating simulation runs in the direction of maximum ascent until the response value drops. The first-order design plus search procedure is repeated until the first-order design does not adequately fit the data. At places of large local curvature (such as an optimum), a second-order model is fitted and the
direction of ascent determined or an optimum found. Because of its dependence on local modeling of the response surface and gradient search, RSM is a local search technique whose success in finding a global optimum depends on the response surface being unimodal. The statistical analyses associated with the experimental designs also assume normality and homogeneity of variance. In simulation experiments, the assumptions of normality, homogeneity of variance and unimodality are often not met (Greenwood, Siochi, and Rees, 1995).

Barton (1992, 1994) concludes that the non-traditional metamodeling techniques offer the most promise because of their robust nature and ability to fit global models. Frequency domain basis function and spatial correlation metamodel approaches have potential but are still in the early stages of development. Problems in the solution procedure for radial basis functions have only recently been addressed (Barton, 1994). Spline and kernel metamodels have the most mature theoretical background and are easy to code or have codes available. The spline techniques have several implementations, the best known in a theoretical sense being the smoothing splines. Spline and kernel smoothing are known in the statistical literature as nonparametric regression techniques (Härdle, 1990).

NONPARAMETRIC-REGRESSION BASED APPROACHES TO SIMULATION OPTIMIZATION

Silverman (1984) shows that spline smooths have a kernel smooth representation, so that a single discussion suffices to describe the behavior of both of these nonparametric regressors. Nonparametric regressors do not have a global functional representation in the same way that polynomial-based regressors do. They are represented by a functional form which applies to any local region of the data. The global response function is the overall result of applying the local function to the data, giving it the ability to respond to local variations in the data (see figure 1.2 for a univariate example). The response at a point is estimated by a weighted sum of the responses in the neighborhood of the point. An illustration of the assignment of weights to the responses for a univariate kernel smooth is given in figure 1.3. Terms such as bandwidth and window are often used instead of neighborhood in the literature. The fit of the nonparametric regressor is critically dependent on the value chosen for the bandwidth. It is preferable to have an objective method of choosing the value of the parameter, though some leading researchers in the field recommend using a visual fit (Härdle, 1990). A theoretically justified (Härdle, 1990) and generally accepted method for choosing the bandwidth is cross-validation, which chooses the bandwidth that minimizes the cross-validation sum of squares.
Figure 1.2: Example of weight function for estimating response at $X = 0.2$ with a bandwidth of 0.4
Figure 1.3: Example of kernel weight function showing assignment of weights to responses covered by a bandwidth of 0.4 at X = 0.2

Chapter 1: Introduction
H"ardle (1990) gives many examples of the application of nonparametric regressors to empirical data sets and of the superior insights obtained compared to the use of polynomial-based regression. Unfortunately, most of the examples use very large data sets, whereas in simulation optimization small samples are of interest because of the high computational cost of simulation runs.

Spline smooths behave like a kernel with a bandwidth that changes with position. This attribute allows splines to follow the local data more closely than kernel smooths with less bias at sharp peaks. However, it also gives the possibility that the spline smooth could be more responsive to outliers in the data, so that the model would incorporate too many features due to noise. Kernel smooths are not affected by outliers to the same extent when a constant bandwidth is used for all positions. Because of their averaging behavior, kernel smooth estimates of sharp peaks have a heavy negative bias. At the edges of the response surface, the techniques give estimates that have larger variance because fewer points fall into the bandwidth. The amount of bias at the boundary of the response surface depends on the activity of the surface, the bias being high if the gradient of the surface is high at the boundary and low if the gradient of the surface is near zero. For kernel smooths, an adjustment to the weight function at the boundary to reduce bias is given by Rice (1984).

According to Barton (1992, 1994), spline smoothing and kernel smoothing appear to be, at present, the most promising non-invasive procedures for solving the simulation optimization problem. What is not clear is how these techniques perform relative to one another, especially at low sample sizes.

**KNOWLEDGE-BASED APPROACHES TO SIMULATION OPTIMIZATION**

The wide variety of approaches described in the previous sections and the realization that some techniques do better than others for particular problems (Jacobsen and Schruben, 1989) has given rise to the idea that all the above techniques could be implemented by a knowledge-based expert system. This approach has the benefit of hiding the complexity of experimentation and analysis from the simulation model user by embedding the required knowledge in the form of If-Then rules in a knowledge base. Shannon, Mayer and Adelsberger (1985) and Fox, Husain, McRoberts and Reddy (1989) explore the potential for expert systems and simulation and conclude that their marriage would facilitate model development and simulation analysis. O'Keefe (1986) gives a taxonomy of expert system/simulation model combinations. For the purposes of simulation optimization, the cooperative model of O'Keefe seems of be most appropriate. The cooperative model views the simulation model and knowledge base as being accessible to the user and that
the knowledge base and simulation model share data and information. Crouch, Greenwood and Rees (1993) build an architecture for a knowledge-based simulation optimization system that follows the cooperative model. Their system classifies the response surface in terms of characteristics important for the choice of a search technique and then uses a knowledge base to select the search technique. Extensions are proposed that enable machine learning, allowing the knowledge base to update its rules after an analysis of the system's performance on past examples of optimizing simulation models.

Crouch, Greenwood and Rees (1993) are to our knowledge the only group to have presented a detailed architecture for the implementation of an expert system to guide a simulation model for simulation optimization. In their work the surface classification is carried out on a metamodel synthesized by a backpropagation neural network. As nonparametric techniques are considered excellent candidates for metamodelling tools in the context of simulation optimization, it is of interest to discover how they would be implemented in a cooperative model architecture and what the impact of their implementation would be on the system architecture.

PURPOSE OF THE RESEARCH

Although the theoretical properties of spline and kernel smoothing are well known in the form of asymptotic statements, the behavior of the techniques in the case of small samples is not well known. In particular, there is little reason given in the literature (Eubank, 1988; Härdle, 1990) to prefer one technique over the other for any given set of circumstances. The implementation of kernel and spline smoothing for metamodelling in simulation optimization has not been reported in the literature, nor have these techniques been considered in the context of a knowledge-based approach to search. Considering the consensus in the literature on the promise of the techniques for metamodelling and on the necessity for knowledge-based support for activities such as simulation optimization, research in these areas is sorely lacking.

This research improves the state of knowledge in these areas through an empirical examination of the relative performance of the nonparametric techniques over a wide range of typical simulation response conditions using small samples. Furthermore, the implications for the Crouch, Greenwood and Rees (1993) knowledge-based simulation optimization system of including nonparametric techniques as metamodelling agents will be investigated. A new architecture will be proposed and the implementation of a nonparametric technique will be demonstrated through an example.

Chapter 1: Introduction
SCOPE AND LIMITATIONS

Throughout this research the simulation optimization problem is restricted to problems in which there are two real valued continuous inputs and a single, real-valued response. Of necessity, the number of problems, represented by deterministic functions plus error, will be limited to three carefully chosen examples to keep the size of the experiment computationally feasible. For the implementation of the nonparametric technique in the knowledge-based system, only the parts of the system pertinent to the successful incorporation of the technique will be described in any detail.

PLAN OF PRESENTATION

The next chapter surveys the literature, tracing the development of spline and kernel smooths and the recognition of such techniques by simulation optimization researchers. Knowledge-based approaches in simulation are described. Chapter three investigates the relative performance of kernel and spline smoothing in the context of simulation optimization, drawing conclusions about the problem characteristics that define the usage of each technique. Chapter four describes a new architecture for the implementation of nonparametric techniques in a knowledge-based simulation optimization system and illustrates the use of nonparametric techniques as a solution procedure through an example. Chapter five summarizes the research contributions and presents plans for further research.
CHAPTER 2
LITERATURE REVIEW

NONPARAMETRIC REGRESSION

The term nonparametric regression covers a wide variety of techniques for the estimation of noisy functions using a function with a local basis. The two techniques described here represent the most widely known and well developed methods in nonparametric regression. These methods were very favorably evaluated in terms of their usefulness for metamodeling in the simulation optimization domain by Barton (1992, 1994). Because of the great depth and mathematical complexity of the literature, the review here will be an abridgment. Kernel methods are treated first, as most nonparametric regressors have a kernel representation.

Kernel smoothing

Kernel smoothing is a form of moving average procedure. A moving average requires just one parameter to be set: the number of responses to include in the average. Each response in the average can be considered to have a weight of one. The division of each weighted response by the sum of the weights ensures that the weights sum to one. It is easy to see the analogs to the moving average in the univariate Nadaraya-Watson estimator (Nadaraya, 1964; Watson, 1964) which produces a kernel smooth \( \hat{k}_h(x) \):

\[
\hat{k}_h(x) = \frac{n^{-1} \sum_{i=1}^{n} K_h(x-X_i)Y_i}{n^{-1} \sum_{i=1}^{n} K_h(x-X_i)} \tag{2.1}
\]

where

\[
K_h(x-X_i) = h^{-1}K\left(\frac{x-X_i}{h}\right) = h^{-1}K(u), \tag{2.2}
\]

\[
\int K(u)du = 1, \tag{2.3}
\]

\( K_h(u) \) is real, continuous and symmetric,

and \( \{X_1, X_2, \ldots, X_n\} \) is a sample of data of size \( n \), \( \{Y_1, Y_2, \ldots, Y_n\} \) are the corresponding responses, \( x \) is the point at which an estimate is required and \( h \) is the bandwidth for the sample.

There are two choices to make in specifying the kernel estimator: the kernel function \( K_h \) and the bandwidth \( h \). Härdle (1990) gives results to show that in terms of asymptotic mean integrated square error...
performance, there is little difference in the performance of various kernels. He suggests that other criteria, such as the ease of computation of the kernel function, could be more important than the choice of the function itself. A kernel function which is easy to compute and has some optimal properties is the Epanechnikov kernel (Epanechnikov, 1969):

\[ K(u) = 0.75(1 - u^2)I(|u| \leq 1) \]  \hspace{1cm} (2.4)

where the indicator function \( I(g) \) equals 1 if \( g \) is true, 0 if \( g \) is false.

Choosing the correct value of \( h \) is critical to the performance of the kernel smooth. The method of choosing \( h \) has to have the following properties:

1. The chosen \( h \) must provide a good fit to the data in terms of mean squared error
2. The bandwidth should have the property of converging to the asymptotically correct bandwidth.

Härdle (1990) compares the performance of several potential methods for bandwidth selection through simulation and concludes that methods based on the method of cross-validation not only have the desired properties but possess robustness to the character of the underlying function. Cross-validation in the context of kernel smoothing was first proposed by Clark (1975). The technique was enhanced by Craven and Wahba (1979) and termed generalized cross-validation. In cross-validation the jth data point is estimated using the data set with the jth point removed for a given value of the bandwidth, i.e.

\[ \hat{k}_{h,j}(X_j) = \frac{n^{-1} \sum_{i \neq j} \frac{K_h(X_j - X_i)}{n^{-1} \sum_{i \neq j} K_h(X_j - X_i)}}{} \hspace{1cm} (2.5) \]

These estimates are used to create the cross-validation function, CV(h),

\[ \text{CV}(h) = n^{-1} \sum_{j=1}^{n} \left[ Y_j - \hat{k}_{h,j}(X_j) \right]^2 w(X_j) \hspace{1cm} (2.6) \]

where \( w(X_j) \) is a weight function that restricts the evaluation of \( \text{CV}(h) \) to the central part of the data.

\( \text{CV}(h) \) is computed for a wide range of values of \( h \) and the value of \( h \) that minimizes \( \text{CV}(h) \) is selected as the value of the bandwidth to be used in computing the estimated response. Golden-section search may be used to find the minimum value of \( \text{CV}(h) \).

Chapter 2 : Literature Review 13
Kernel methods encounter problems with estimation near the edges of the data because points there have fewer points on one side than on the other. This gives more variance and/or bias to estimates in the boundary region. It is for this reason that the evaluation of CV(h) is restricted to the central part of the data. However, Härädle and Marron (1985) have shown that the value of h is not significantly affected by the amount of data at the boundary cut off by the weight function in equation 2.6 until the cutoffs at each end equal 10% of the data. In dealing with small samples the bandwidth can easily cover 60% of the data range, making little data available for cross-validation. Rice (1984) provides a method to reduce bias encountered at the boundary. He defines a boundary kernel that is a linear combination of two kernel smooths of bandwidth h and αh, where x is taken as the distance from the edge of the data, as follows:

\[ K_p^B(u) = (1 - R)K(u) - \left( \frac{R}{2} \right) K\left( \frac{u}{2} \right) \]  
(2.7)

where \[ R = -\frac{v_K(1, \rho)/v_K(0, \rho)}{v_K(1, \rho/\alpha)/v_K(0, \rho/\alpha) - v_K(1, \rho)/v_K(0, \rho)} \]  
(2.8)

and \[ v_K(0, \rho) = \int_{-1}^{0} K(u)du, \]  
(2.9)  
\[ v_K(1, \rho) = \int_{-1}^{0} u^2 K(u)du, \]  
(2.10)  
\[ \rho = \frac{x}{h}, \quad \alpha = 2 - \rho. \]  
(2.11)

At a distance h from the boundary of the data, the boundary kernel is equivalent to the interior kernel, but as the boundary is approached the shape of the boundary kernel departs increasingly from that of the interior kernel (see figure 2.1). Using this correction, complex though it may appear, gives greater confidence in determining the bandwidth by performing cross-validation on all the data.

Another kernel estimator that is used in the literature is that of Gasser and Müller (1979). The performance of this estimator is comparable to that of the Nadaraya-Watson estimator (Hall and Wehrly, 1991). Both estimators can be used with kernels of a higher order of differentiability to estimate derivatives of the underlying function. The estimators can be extended to higher dimensions.

For the Nadaraya-Watson estimator, the multivariate case is represented by the following kernel function (Härdle, 1990):
Figure 2.1: Changes in Rice boundary weight functions as $x$ approaches the right boundary: Epanechnikov kernel, bandwidth = 4
\[ \hat{k}(x) = \frac{n^{-1} \prod_{j=1}^{d} \kappa_{h}(x_j - x_0) y_i}{n^{-1} \prod_{j=1}^{d} \kappa_{h}(x_j - x_0)} \] (2.12)

Note that the same smoothing parameter is used for each dimension. Unless a large amount of data is available, problems in high dimensions have a sparse distribution of data. For example, to obtain the same data density as a 10 response, univariate problem requires 100 responses for two dimensions. The kernel methods will fail if there is insufficient data in an area of space. K-nearest neighbor (K-NN) methods surmount this problem by using a weighted average of the k-nearest neighbors of a point, effectively using a variable bandwidth (Härdle, 1990). However, the performance of K-NN methods is inferior to that of kernels when the data are uniformly spaced.

Elaborate methods have been constructed to give kernels a variable bandwidth capability, wherein the kernel bandwidth varies with the curvature of the estimate of the underlying function (Müller and Stadtmüller, 1987). This is achieved by spline smooths in a natural way (Härdle, 1990).

Little work has been carried out in the area of experimental design for kernel smooths. An important paper by Müller (1984) determined that for kernels with a constant bandwidth across the region a uniform design was optimal. Kernels using a variable bandwidth should have design points placed so as to have the design point density proportional to the density of the square of the second derivative of the estimate. Faraway (1990) utilized variable bandwidth kernels of the Gasser-Müller type in an attempt to deliver a sequential design technique using Müller's results. His procedure needs "sufficient data" to initialize itself. Simulation results showed that the procedure performed substantially better than equispaced designs in a mean square error sense. Because of the global modeling capabilities of kernel smooths, problems such as aliasing of the true function through inadequate sampling are important. This issue has not been addressed in the literature in the context of nonparametric regressors.

**Spline smoothing**

The idea behind spline smoothing is to model a global function through local polynomial functions (splines) that join smoothly at points called knots. The order of the polynomials used depends on the amount of smoothness required in the global function. For computational simplicity, the positions of the knots are usually chosen to coincide with data points, as this gives a linear problem versus a nonlinear problem. Cubic polynomials are often chosen as they provide a sufficient degree of smoothness and flexibility of form coupled with a low computational burden due to their low number of coefficients (Dierckx, 1993).
Splines have long been used to interpolate data. When the data are deterministic, this results in a smooth function pleasing to the eye. When the data are noisy, the interpolating spline produces a fit to the data that is too "wiggly" (Härdle, 1990). Whittaker (1923) was probably the first to suggest relaxing the interpolation condition and suggested a series of adjustments to the ordinates of the data to minimize the following:

$$\min_{m, \lambda} \lambda \sum_{i=1}^{n-m} (\Delta^m y_i^*)^2 + \sum_{i=1}^{n} (y_i^* - y_i)^2.$$  (2.13)

where $\Delta^m$ is the mth divided difference, $1 \leq m \leq n$, $\lambda$ is the smoothing parameter and m and $\lambda$ have to be chosen.

The modern formulation of problem 2.13 was given by Schoenberg (1964), who showed that a spline function was the solution to the problem of finding a function $f(x)$ with square integrable mth derivative to solve:

$$\min_{f(x)} \lambda \frac{1}{m} \int_1 [f^{(m)}(x)]^2 dx + \sum_{i=1}^{n} [f(x_i) - y_i]^2,$$  (2.14)

where $m < n$, $\lambda > 0$ and $I = [a, b]$.

The objective function of this problem has two components: the first term is a roughness penalty function that dictates the smoothness of the resulting solution through the smoothing parameter $\lambda$, the second term is a least squares measure of fit to the data. A value of zero for $\lambda$ results in an interpolating spline solution, while a value of infinity for $\lambda$ gives a linear regression solution. Reinsch (1967) formulates the problem as a constrained optimization problem and uses a Lagrangian method to show the solution is a cubic spline. The natural way in which the spline smoothing problem portrays itself as an optimization problem makes it difficult to envision what the process is doing with the data. A line of research developed to try and provide a more intuitive representation of a spline smooth. Notable results include those of Silverman (1984), in which the smoothing spline was shown to be approximately equivalent to a kernel method using a design point density-dependent bandwidth, and those of Jennen-Steinmetz and Gasser (1988), where spline smoothing is shown to be one case of a general class of kernel estimators with design-adaptive bandwidths, with the cases being defined by the value of a single parameter.

Wahba and Wold (1975) introduced the idea of using cross-validation to choose the value of $\lambda$ for the smoothing spline. The technique of generalized cross-validation (GCV) was developed by Craven and
Wahba (1979) for the general case in which data were unequally spaced. GCV involves rotation of the co-ordinate system to achieve a circulant prediction matrix which is then followed by cross-validation on the transformed data. Craven and Wahba also showed that the technique produced a value of the smoothing parameter that converged asymptotically to the value of the true smoothing parameter. Uteras (1987) proved the asymptotic optimality of GCV when used with a multivariate smoothing spline, the thin plate smoothing spline. Kohn, Ansley and Tharm (1991) carried out an extensive simulation study in which they demonstrated that GCV gave excellent results for practical sample sizes.

The convergence rate of the smoothing spline is studied by Rice and Rosenblatt (1981). They find that the convergence rate of the spline is very dependent on the derivatives of the true function at the boundaries, being fastest for periodic functions where the boundary gradient is well defined. They also comment on the implications of their findings for cross-validation, in that boundary bias could affect the GCV statistic and become a significant problem in higher dimensions. Various methods (Gu, 1987; Gehlert, 1992) have been proposed to reduce boundary bias for splines. Pragmatically, the implication may be that design points are needed at the boundary to help reduce the uncertainty there.

Multivariate smoothing splines may be put into one of two groups (Barton, 1994):

1. Smoothing splines; where knots occur at the data points and the smoothing parameter is chosen by GCV.

2. Regression splines; where knots are chosen by the user and the fitting procedure uses univariate splines in a stepwise procedure.

When smoothing splines are extended to higher dimensions the number of coefficients to be determined in a full tensor product representation requires an exorbitant amount of data. As an alternative, Wahba (1986) put forward interaction splines, in which the model consists of linear combinations of products of at most two univariate splines. Friedman (1991) proposed multivariate adaptive regression splines (MARS) in which GCV is used in a stepwise procedure to select knots and the degree of the univariate products. The model uses truncated power basis functions, which are not numerically robust. Breiman (1991) introduced the PI method. Again a stepwise procedure is used to select univariate product degree with a large number of knots. Knots or univariate elements are then eliminated in a reverse stepwise procedure. GCV is used to control both procedures. Thin plate splines have a radial basis function representation, i.e.

\[
f(x) = \sum_{i=1}^{n} a_i \| x - X_i \|^{2} \log \| x - X_i \|. \tag{2.15}
\]
The a_{i} are found by replacing f(x) by g(X_{i}), i = 1, 2, \ldots, n and solving the resulting linear system (Barton, 1992, 1994).

Generally, the computation of spline smooth models is a sophisticated numerical procedure because of the possibility of ill-conditioned matrices occurring in the process. To a far greater degree than with kernel smooths, the availability of good computer codes is a major determinant of which spline smooth to implement. Gu (1989, 1992) has made available a bundle of routines written in Ratfor (Rational Fortran) and FORTRAN which enable thin plate splines and interaction splines to be implemented with single or multiple smoothing parameters.

The computational approach recasts equation 2.14 as:

\[
\min \left\{ \frac{1}{n} \sum_{j=1}^{n} (y_j - L_j f)^2 + \lambda J(f) \right\} \tag{2.16}
\]

where \( f \in \mathcal{H} \), a Hilbert space in which the \( L_j \)'s are bounded linear functionals, and \( J(f) \) is the square of a semi-norm in \( \mathcal{H} \) which measures the "roughness" of the fit (Gu, 1989; 1992).

A full description of the solution method for this formulation can be found in Gu (1989, 1992).

**SIMULATION OPTIMIZATION**

The computer-intensive nature of simulation and hence simulation optimization has meant that approaches to the problem of optimizing the performance of a simulation model have been closely related to the development of computer power and accessibility. The development of the first simulation models took place in third generation languages such as FORTRAN. By the late 1960's simulation languages had been developed that incorporated features common to all simulation models, such as a simulation event list and clock. During this period, FORTRAN-based GPSS was developed by IBM. Its competitors included SIMSCRIPT and Simula 67. The early simulation languages were mainframe-based, as this was the dominant computing environment and the only one that could handle "large" simulation models. In the late 1980's simulation languages began to use graphical interfaces (SLAM II) and allow graphical model development through the placement of icons on the screen. Nearly all simulation model development languages were ported to personal computers, which offered a superior interface and reasonable computing power in comparison to the mainframe. The 1990's have seen the introduction of simulators (Law and
Kelton, 1991), which allow simulation models for particular application areas to be built without writing any code. Simulators are object-based development systems oriented towards specific applications such as manufacturing or communications. These systems often have sophisticated animation capabilities and presentation quality graphics in order to more easily purchase management confidence in the model. Examples of state of the art simulators are ProModel and COMNET III by Promodel Corporation and CACI Products, Inc., respectively. Also available are object-oriented general purpose simulation languages such as MODSIM II, the successor to CACI Products' SIMSCRIPT II.5.

Simulators allow simulation models to be developed more quickly and with personnel less skilled in simulation than with traditional language-based models (Law and Kelton, 1991). The ability to customize the code is very limited with some products. The movement towards object-based graphical environments in simulation model development means that, practically speaking, most simulation models will have to be considered black boxes, with a set of inputs and outputs. Custom-built simulation models will be relatively expensive to build because of the greater sophistication of the development team required and the use of fewer pre-built objects in the code, but they can offer custom enhancements that can give efficient code and analysis of results.

The development of techniques for simulation optimization has progressed at a slower rate because its computational demands are many times that of a single simulation run and the problem itself is difficult in general. Smith (1973a) carries out an empirical investigation of simulation optimization techniques. Of the seven search techniques that he evaluates, four of them are based on RSM, two are single-factor at a time searches and one is a random search. The choice of procedures made by Smith reflects the paucity of available techniques that were suited to search for local optima with noisy data. Smith tests the techniques on response surfaces derived from high order deterministic functions with added error, again reflecting the state of mind at the time in conventional optimization that a good search must be able to perform efficiently in high dimensional space and that local optima are usually small. The results of the tests showed RSM dominating the other techniques when local optima were absent but performing very poorly when local optima were present. Smith (1973b, 1974, 1976) then developed an automated simulation optimization system that used heuristic rules to control the application of RSM-based search techniques.

Daughety and Turnquist (1978) realized that the traditional search techniques take too many simulation runs to achieve optimality. They utilize hub and spoke design point allocation and a single factor at a time search using interpolating splines to provide pseudo-data between simulation runs. A global polynomial function is fit to both true and pseudo-data to provide a metamodel of the response surface. Their approach considers
the number of simulation runs left in the budget at each stage. Unfortunately, their technique relies on the response surface possessing pseudo-convexity and so in the presence of local optima only a local optimum can be guaranteed. Daughety and Turnquist (1981) enhance their technique by concentrating on the problem of run allocation in stages. They utilize a probabilistic measure to distribute points among the stages of the search and distribute points spatially within each stage using a star network. The algorithm is tested on five functions with added noise, only one of which is multimodal, with reasonable results on the unimodal functions.

Meketon (1987) surveys developments in the early 1980's, though with a heavy emphasis on stochastic approximation and perturbation analysis techniques. A brief review of traditional techniques concludes that the performance of the Nelder-Mead algorithm improves with increasing noise, but is outperformed by gradient-based technique at lower noise levels. Because convergence to the optimum is not guaranteed using the Nelder-Mead algorithm, Meketon spends more time discussing methods that are guaranteed to converge and can deal with noisy data. RSM is dismissed as requiring too many simulation runs, and the bulk of the discussion centers on the possible application of stochastic approximation and perturbation analysis in combination. He prefers this approach as the optimization can be carried out during one long simulation run by altering the "input" parameters dynamically. However, the approach is derivative-based and hence has no provision for dealing with responses that are multimodal, and also assumes that the sequence of values of the response is a regenerative process. Meketon's views are extreme in terms of the importance placed on the simulation run budget. Note also that the stochastic approximation method is an invasive technique, in that the optimization process is embedded in the simulation model. Meketon's ideas ran counter to the direction that mainstream simulation modeling was going in: i.e., taking advantage of more accessible and more powerful personal computer systems with modular, graphical, simulation model development environments.

Jacobsen and Schruben (1989) attempted a more comprehensive survey of simulation optimization techniques. Whereas Meketon classified simulation optimization techniques by the amount of known structure in the simulation problem, Jacobsen and Schruben classify techniques by search methodology. Their classification consists of four categories: path search (gradient-based techniques), pattern search (no gradient used, but use set designs), random search (no set designs for point allocation) and integral search (space covering, deterministic functions). They note that only the last category is inherently global in its approach to search, but that the enhancements to it to enable it to cope with noisy problems result in slow convergence and a lot of work for each iteration of the algorithm. The discussion on path search methods is dominated by RSM-based approaches, with stochastic approximation and perturbation analysis also being...
mentioned. Research casting doubt on the general validity of the perturbation analysis approach is reported. Jacobsen and Schruben conclude that derivatives are difficult and expensive to estimate. The patterns methods reviewed included Hooke and Jeeves search and the Nelder-Mead Simplex method, and it was noted that both use functional evaluations and do not assume continuity of response. Random methods were considered slow to converge and to require too many iterations.

Safizadeh (1990) surveys simulation optimization and adds to the review of Jacobsen and Scruben through mentioning the development of variance-reduction strategies to improve the efficiency of RSM-based techniques. He concludes by observing that simulation optimization is so varied in its skill and knowledge requirements that there is a need for software to assist users in the choice of an optimization technique.

A broad review of the field is given by Azadivar (1992). Besides giving a general formulation of the simulation optimization problem, he summarizes the major issues, forms classes of problems and reviews the techniques applicable to each class of problem. Azadivar re-iterates Safizadeh's concern for the difficulties in interfacing simulation models to optimization routines and states that this is an issue researchers need to address. In the technique review, new techniques and approaches reported include a simulated annealing based search procedure and several multi-criteria optimization approaches.

Sargent (1991) is probably the first in the simulation optimization field to appreciate the changes in the computing environment and the effect they would have on simulation optimization. He refers to the polynomial models produced in RSM as metamodels, that is, models of the simulation model. Sargent views metamodels as a step further in the process of abstraction from the real system than the simulation model and as providing a substitute input-output model. RSM is often criticized for needing too many simulation runs, but Sargent appeals to increases in computing power and parallel processing to provide the data for metamodeling techniques. The paper focuses on the research issues for metamodels that need to be addressed and briefly states eleven questions that cover experimental design, factor screening, match of metamodel type to problem type, sequential design and accuracy, precision and validity of the metamodel.

The shift from traditional search-based approaches to simulation optimization over to the metamodeling approach gained momentum with the review of metamodeling techniques by Barton (1992). In this review Barton covered general-purpose metamodeling techniques, avoiding coverage of traditional techniques which seek to exploit special structure in the simulation model or response function. The discussion covered polynomial response surface models and a Taguchi style approach, generalized linear models, tensor product splines, interaction splines, Multivariate Adaptive Regression Splines (Friedman, 1991), the Π
method (Breiman, 1991), radial basis functions, kernel smoothing, spatial correlation models and frequency domain approximations (Fourier and wavelet). Barton concludes that polynomial metamodels are inadequate and unstable for global metamodeling; spline, kernel and radial basis function methods are robust and are capable of global metamodeling. He perceives the main issues for non-traditional methods to be experimental design and model analysis. Barton (1994) gives an updated version of the Barton (1992) review and concludes that radial basis functions, which also cover some spline models, are very promising but a methodology for creating experimental designs for these methods is lacking.

The recent shift in emphasis in the area of simulation optimization has come as desktop computing power has come to rival that of mainframes of a few years ago. Work in the areas of spline and kernel smoothing has also matured in the late 1980's. The combination of computing power and robust models has made it possible to think of simulation optimization as a feasible task, though problems involving more than six dimensions still seem extremely large. Despite the specification of interesting research topics by Sargent and Barton, no work to our knowledge has been published that evaluates or explores nontraditional metamodeling techniques in the context of simulation optimization and particularly in the area of knowledge-based simulation optimization.

**KNOWLEDGE-BASED APPROACHES TO SIMULATION OPTIMIZATION**

The tremendous amount of knowledge required to participate in the development of a simulation model for optimization has served as the impetus for research into ways that of incorporating expert systems into the process. Expert systems are computer applications that incorporate the following characteristics (Rolston, 1988):

1. Extensive domain-specific knowledge
2. Ability to apply search techniques
3. Use heuristic analysis
4. Can infer new knowledge from old knowledge
5. Use symbolic processing
6. Can deliver explanations of the system's reasoning

A general expert system architecture is presented in figure 2.2. Knowledge is often stored in the knowledge base in terms of IF-THEN rules (heuristics) and statements of fact. The inference engine is able to use the rules and facts in the knowledge-base in conjunction with facts added to the system by the user to perform inference, which in turn generates more knowledge or is capable of initiating some process.
Figure 2.2: Architecture of an expert system (Rolston, 1988)
Shannon, Mayer and Adelsberger (1985) perceive the marriage of expert systems and simulation as inevitable and due principally to their complementary natures. Simulation can provide the ability to dynamically predict future events to expert systems, while expert systems can provide simulation with the capacity to store domain knowledge and heuristics. Shannon, Mayer and Adelsberger envision a comprehensive joining of the two methodologies, with the expert system assisting in the simulation model specification, model construction and in output analysis.

O'Keefe (1986) presents a taxonomy for combining expert systems with simulation. The ideas of Shannon, Mayer and Adelsberger (1985) appear to fall into O'Keefe's "expert system as intelligent front end," whereas examples of actual expert system and simulation systems, such as KBS (Reddy, Fox and Husain, 1986), fall into the category of co-operative systems. Fox, Husain, McRoberts and Reddy (1989) have developed a object-oriented Knowledge-Based Simulation (KBS) system that assists in the specification of a simulation model and in the use of the model to achieve user specified goals. In particular, KBS uses path analysis to explore a local region of the response and to indicate the changes in inputs required to move towards satisfying the goals. This process is not intended to be an optimization procedure.

Crouch (1992) describes a knowledge-based simulation optimization system with learning that also falls into O'Keefe's co-operative systems category. Here the simulation model is developed separately, without the aid of the expert system. The expert system is focused on the optimization process. In particular, the parameters supplied by the user for a simulation optimization problem are augmented by information derived from a metamodel fitted to responses from an initial experimental design. The resulting parameter set is then said to classify the response surface, and based on the values of the parameters the knowledge-base selects a search technique. The search is carried out until the user's goals are met or until the search encounters conditions under which it is invalid, in which case the classification process is invoked and the process iterates. Crouch uses a backpropagation neural network for synthesizing a metamodel, but does not necessarily favor the use of this technique in every problem.

The literature recognizes a need for assistance with the optimization of simulation problems and work has began on specifying a knowledge-based system that can accomplish this task. A number of disparate fields need to come together to provide a general knowledge-based approach. Such an approach would embody a knowledge-base containing a set of rules, initially a mixture of analytical and heuristic results, that can guide or automate the implementation of a comprehensive suite of search techniques and metamodelling strategies so as to achieve a user's goals. Inhibiting the detailed development of such an approach is a lack
of knowledge concerning the relative performances of the various search and metamodeling techniques and the circumstances under which one is favored over another.
CHAPTER 3

A PERFORMANCE-BASED STUDY OF NONPARAMETRIC-METAMODELING TECHNIQUES FOR THE CLASSIFICATION OF RESPONSE SURFACES IN SIMULATION OPTIMIZATION

INTRODUCTION

The simulation optimization problem

Simulation is a modeling tool that is often used to investigate problems that are analytically intractable. The variables in the model are either controllable or uncontrollable, depending on whether the user is able to set values of the variable or not. Uncontrollable variables \( Z \) are usually present in a model and are an implicit component of the output. The simulation model is generally stochastic in nature as its constituent processes contain uncertainty in the form of random error. This causes the set of responses \( \{ Y_j, j = 1, 2, \ldots, m \} \) to be random variables, resulting in a distribution of responses for any set of input variables, \( \{ X_i, i = 1, 2, \ldots, n \} \). The simulation model does not, on its own, give any indication of the values of the inputs that will maximize or minimize any response of interest. Simulation optimization is the problem of finding the values of \( n \) controllable input variables that optimize the values of \( m \) response variables. The simulation optimization problem is a constrained optimization problem with the constraints defining a feasible region \( S \) of the controllable factors \( X \). The general simulation optimization problem can be stated as:

Optimize: \[ E(Y) \text{ over the region } S \subset \mathbb{R}^n, \] \hspace{1cm} (3.1)
where the domain of \( S \) may be either continuous, or discrete, or mixed, and
\( X = (X_1, X_2, \ldots, X_n) \in S \)

Subject to: \[ g(X) \leq 0 \] \hspace{1cm} (3.2)
where \( g(X) \) is a vector of deterministic constraints that are typically of the form:
\[ l_i \leq X_i \leq u_i \hspace{1cm} i = 1, 2, \ldots, n \] \hspace{1cm} (3.3)
\[ l_{eq} < f(X) < u_{eq} \hspace{1cm} q = 1, 2, \ldots, b \] \hspace{1cm} (3.4)
and \( b \) is the number of constraints involving more than one controllable factor.

These problems are known to be difficult to solve because of simulation models' stochastic nature and often lengthy run time requirement.
Approaches to the simulation optimization problem

Determining the position of the optimum response is usually carried out by employing an appropriate search technique over the region S. When the region S is continuous and the response is real valued, gradient-based search techniques can be attempted. For example, Response Surface Methodology (RSM) (comprehensively treated in Box and Draper, 1987) is a mature, sophisticated set of techniques that uses low-order polynomial functions to model a small subregion of S to provide a good gradient estimate for a steepest ascent procedure. Second-order polynomial models are used to model the region around the optimum, where curvature is normally present. If the response surface is multimodal, then RSM is not guaranteed to find the global optimum because its procedures act on small subregions. If the region is discrete or multimodal in nature then a search technique, such as random search, may be employed in spite of its poor convergence properties. Smith (1973) examines the performance of a range of search techniques and concludes that no single search technique can guarantee success.

Response characteristics such as the presence of several local optima, high variance and heteroscedasticity coupled with a relatively high computation cost for each simulation run make simulation optimization a difficult problem to solve using a single technique. Jacobsen and Schruben (1989) conclude that certain search techniques do well in certain situations, but that there is no general procedure available. These difficulties suggest two alternate approaches. The first is a global approach to modeling the response, so that the presence of local optima can be detected. The second is a multi-strategy search approach that can select a search technique depending on the information supplied by a global metamodel. A metamodel in this context is a function that acts as a proxy for the simulation model, i.e., it is a model of the simulation model. A consideration of these proposals raises a number of questions. If metamodeling is to be used, which technique should be employed and under what conditions? How can the metamodeling process and the search methods cooperate on the simulation optimization problem? The literature suggests the directions to be followed to answer these questions.

The recent literature recognizes the key role of metamodeling in investigating the behavior of simulation models. Sargent (1991) lists research issues for metamodeling, with an emphasis on polynomial-based metamodels. Barton (1992, 1994) reviews metamodeling types such as polynomials (regression), Fourier series, nonparametric regression (kernel smoothing and spline smoothing), radial basis functions, regression splines and spatial correlation models. He concludes that nonparametric methods are flexible, robust and provide good fits while their parametric counterparts tend to be unstable, inflexible and poor at modeling arbitrary shapes.

Chapter 3 : A Performance-Based Study ...
RSM is a methodology that integrates metamodeling and search techniques, but it is a single strategy approach employing first- and second-order parametric models and gradient-based search techniques exclusively. While the technique is very powerful when the conditions suit its methodology, for example when variance is low in the region of interest and only a single optimum exists (Smith 1973), it can easily fail outside these conditions. Crouch, Greenwood and Rees (1993) detail a classifier knowledge-based simulation optimization system (CKBOS), a multi-strategy approach that uses a metamodel-then-search process (see figure 3.1). The function of the Crouch, Greenwood and Rees classifier (the Classifier) is to metamodel (synthesize) the response surface from simulation model output taken at various combinations of the input variables so that the values of certain characteristics, χ, of the surface can be determined (characterized). The characteristics vector, {χ₁, χ₂, ..., χₙ}, of the response surface places the surface at a point in χ space. Each point in χ space is associated with a set of search procedures (the search process in figure 3.1) and their corresponding performance measures. These relationships are stored as rules in a knowledge base. The rules are used to select the best search technique for the surface. Crouch, Greenwood and Rees demonstrate that the Classifier within their knowledge-based simulation optimization system can be used to switch search strategies during the optimization process, taking advantage of the strengths of each search technique. Currently, they use a backpropagation neural network to carry out the metamodeling, but recognize that several metamodeling approaches need to be considered, either singly or in conjunction with each other.

Barton (1994, p. 242) concludes that "extensive computational comparison of the methods is needed," where the methods referred to are the nontraditional or nonparametric methods. He states that the main impedance to such an undertaking is the lack of generally available computer codes. This paper carries out such a comparison with a focus on the use of nonparametric metamodeling in a multi-strategy framework for simulation optimization. The rest of this paper is organized as follows. In the next section we review nonparametric regression techniques, then we describe the Classifier and develop the research questions for the paper. The methodology to be used, results, analysis and conclusions are followed by an example.

**REVIEW OF METAMODELING TECHNIQUES**

Barton's (1992, 1994) conclusions regarding nonparametric methods are supported by examples in the literature. Hardle (1990) provides numerous examples of the superiority of nonparametric methods over parametric methods in fitting a function of arbitrary shape. One example concerns an investigation into

*Chapter 3 : A Performance-Based Study...*
Figure 3.1: The Classify-Then-Search process (Crouch, Greenwood and Rees, 1993)
the income density of British households over a period of thirteen years. A unimodal parametric model was fitted to the data and judged to have adequate fit. A kernel smooth of the data showed a bimodal distribution that changed the relative sizes of its peaks over time. In another example, data on human height growth were modeled using both parametric and nonparametric techniques. The nonparametric technique revealed an extra growth spurt that was missed by the parametric fit.

Nonparametric regression techniques share several characteristics. They are all functions that compute, given a sample of data, an estimated response at a particular point, x, using only the data that are local to x. Each method requires the determination of a smoothing parameter that regulates the size of the neighborhood around x used to estimate the response. With a neighborhood of zero, the resulting surface would be composed of the original data. With a neighborhood equal to the domain of interest, the resulting surface would be a linear regression. The optimal neighborhood size (or bandwidth or smoothing parameter) in a least squares sense is determined through a procedure called cross-validation. For a range of neighborhood values, estimates are calculated for the data points that leave out the data point itself from the estimation process. The cross-validation measure is the sum of the squared deviations of the leave-one-out estimates from the actual responses over a specified subregion of the domain. The neighborhood size that minimizes the cross-validation measure provides the best least squares smooth. It is these characteristics that give nonparametric methods their ability to model arbitrary shapes. Because these methods are not as widely known as the parametric methods, we describe briefly the principal techniques: K-Nearest Neighbor smoothing, kernel smoothing and smoothing splines.

K-Nearest Neighbor (K-NN) smoothing is described in Härdle (1990). Given a sample of data, equally spaced or unequally spaced, the K-NN smooth estimates the response y, at a point x, as the weighted sum of the responses of the K nearest data points to x. Here the value of K is the value of the smoothing parameter, determined by cross-validation. Note that the size of the neighborhood in Euclidean terms is dependent on the density of the data, i.e., it is adaptive. The bias and variance of the resulting metamodel vary across the surface, depending on the data density. In the equally spaced data case, the K-NN smooth is dominated by the kernel smooth in terms of its bias and variance properties. For this reason, we will exclude K-NN smoothing from this research.

Kernel smoothing (for example, see Härdle (1990)) expresses the estimate of the response y, at a point x, as the weighted sum of the responses of the data points that lie within a distance h of x. h is the bandwidth of the function and it controls the amount of smoothing. The Nadaraya-Watson form of the kernel smooth function in the univariate case is

Chapter 3: A Performance-Based Study . . .

31
\[
\hat{m}(x) = \frac{\frac{1}{n} \sum_{i=1}^{n} K_h(x-X_i)Y_i}{\frac{1}{n} \sum_{i=1}^{n} K_h(x-X_i)}
\]  

(3.5)

where \( \hat{m}(x) \) is the estimate of the response at \( x \). The kernel weight function, \( K_h(u) \), can be one of a number of functions, but the differences in performance between them are small (Härdle, 1990). The choice of bandwidth is of much greater importance to the quality of the fit and is selected by cross-validation.

Smoothing splines (Schoenberg (1964), for a discussion see Silverman (1985)) are piecewise cubic polynomials that are joined smoothly at fixed points called knots so as to cover the region of interest. The optimal knot locations in \( X \) space are at the data locations. The coefficients for the spline functions are found from the simultaneous solution of a set of equations that constrain the splines to have continuity in the first and second derivatives at the knots. Univariate smoothing splines result naturally from the solution of the following optimization problem:

\[
\min_{g(x)} S_\lambda(g) = \frac{1}{n} \sum_{i=1}^{n} (Y_i - g(X_i))^2 + \lambda \int (g''(x))^2 \, dx
\]

(3.6)

where \((X_i, Y_i)\) is a data point, and \( g(\cdot) \) is the function to be solved for. \( S_\lambda(g) \) has two components: the first term is a least squares fit measure, the second a roughness penalty controlled by the smoothing parameter \( \lambda \).

For a function that is very "wiggly," the gradient of the function (the first derivative) changes its value rapidly, especially at the peaks and troughs of the function. As the second derivative is a measure of the rate of change of the first derivative, the magnitude of the integral of its values over the range of the function will be larger the "wigglier" the function. Thus \( g'' \) in the second term acts as a roughness measure, whose importance relative to the first term is adjusted through the parameter \( \lambda \). The unique solution to the problem is the cubic spline (Schoenberg, 1964). Cross-validation is used to determine \( \lambda_{\text{opt}} \), the optimal smoothing parameter that balances the fit to the data against the smoothness of the resulting function. Although splines appear to be very different from kernel smooths, splines have been shown to have a kernel representation (Silverman 1984).

A large number of analytical results can be found in the literature for kernel and spline smooths (Härdle 1990, Eubank 1988). The great majority of the results concern the behavior of the techniques under an additive error model with known conditions such as homoscedasticity and normality of the model error term. It is possible to make general statements about the asymptotic bias or variance of the metamodel
under known conditions for the kernel and spline smooths and this ability can be useful for the validation of a metamodel. However, Greenwood, Siochi and Rees (1993) present simulation output results for an inventory problem that has heteroscedasticity in the response over the region of interest. This implies that when a metamodel is to be used in a situation where little prior information is known about the response surface, it is necessary to know something about the performance of the metamodel when the usual assumptions do not hold.

**METAMODELING AND THE CROUCH, GREENWOOD AND REES CLASSIFIER**

The Crouch, Greenwood and Rees approach to the simulation optimization problem is to take a sample of simulation model outputs over the region S and use a metamodel to synthesize a response surface. Information obtained from the metamodel is subsequently used to choose the most appropriate search technique. Validation procedures check to see if the search technique remains valid as information about the problem changes. If it becomes invalid, the metamodeling procedure is invoked again. The Classifier is an important part of their paradigm and is used in two situations. First, it is used to gain an initial characterization of a surface. Second, it is used to recharacterize the surface in another region or in a sub-region. The initial characterization has to contend with a possible total lack of information regarding the form of the surface, whereas subsequent characterizations have estimates of various characteristics obtained from the prior syntheses of the surface that may need refining or updating. With little information to provide guidance, the question of which metamodeling technique to use is difficult. An appropriate initial choice of metamodeling technique would be one that can satisfy the requirements of the Classifier under a wide range of conditions, i.e. a robust technique.

In order to choose a search technique, the Classifier example provided in Crouch, Greenwood and Rees, which is based on Smith (1973a), requires one or more of the following characteristics to be known:

1. presence or absence of local optima
2. magnitude of the error in the response
3. distance of the starting point from the true optimum
4. number of controllable factors
5. number of simulation runs available
6. relative activity of the controllable inputs

The metamodel does not need to provide all the characteristics required by the Classifier. For example, the assessment of the relative activity of the inputs could be determined by a factor screening procedure that
precedes the optimization. Filtering out the inactive factors results in a more efficient and less computationally burdensome investigation. Screening may be carried out using simple two-level experimental designs to determine the activity of the factors across the surface (Box and Draper 1987), or frequency domain analysis (Morrice & Schruben 1993). The number of available simulation runs should initially be provided by the user of the system, as should the number of controllable factors. The magnitude of the pure error variance at a point can be estimated by replication of simulation runs at that point. This leaves the determination of the distance from the starting point and the detection of local optima to the metamodeling procedure. The metamodeling problem in the context of the Crouch, Greenwood and Rees Classifier is as follows:

A. Specify the proportion of the simulation runs available for metamodeling, \( p_m = \frac{s}{N} \), the locations of those runs, \( \{ X_i, i = 1, 2, \ldots, s \} \), and the number of replications at each location, \( r \), to form a sample from which the metamodel \( \hat{m}(X|Z) \) will be constructed and the estimates of the surface variance determined. In this research we will set \( r = 1 \), as it is of interest to test metamodeling techniques under low data densities because of the large cost associated with simulation model runs. The determination of a variance estimate can be carried out by replication at a single design point if necessary.

B. Construct a metamodel \( \hat{m}(X|Z) \) that has the following properties:

1. It provides an estimate of the optimum, \( (\hat{X}_{opt}, \hat{Y}_{opt}) \), that can be used as a starting point for a search.
2. It provides a good fit to the true surface by placing all the significant optima close to their true locations while preserving the significance and relative magnitudes of the optima.
3. It does not have any inherent, significant, capability for constructing optima that are not present in the true function.
4. The representation of characteristics is robust in the face of an initial lack of data concerning the response surface.

While Crouch, Greenwood and Rees consider such issues as the choice of metamodeling technique, layout of the simulation runs and determination of the characteristics from the metamodel, more research is needed in these areas: this is the motivation for this research.
Research questions

In this paper the behaviors of nonparametric metamodeling techniques are explored to assess their ability to supply a surface Classifier with information referred to above. Four main research questions will be addressed:

1. How successful is each metamodeling technique at identifying a good starting point for a gradient-based (classical) search strategy?
2. a). How effective is each metamodeling technique at capturing essential features of the surface?
   b). Does use of the metamodel lead to generation of misinformation about the surface characteristics?
3. How good a fit is the metamodel to the expected value of the surface?
4. Is any one metamodel more robust than the others in providing information required by a Classifier under the test conditions?

METHODOLOGY

The performance assessment of the metamodeling techniques is based on a small-sample simulation experiment. There are two main factors to consider in judging the appropriateness of this approach. First, the performance evaluation of nonparametric techniques in the statistical literature focuses on asymptotic results, and practical examples of the techniques are usually confined to large data sets (Härdle 1990, Silverman 1985). Second, in simulation studies, the simulation model runs can be expensive, making methodologies that use few runs to accomplish their objective attractive. It therefore seems sensible to examine metamodel performance over a range of possible simulation model conditions using relatively small sample sizes.

In general, this study examines different metamodeling techniques as a function of the number of the simulation runs used to see which techniques perform well for simulation optimization. Four different surface characteristics are varied in a full-factorial-design experiment to provide different surfaces as a "test bed." The analysis of variance (ANOVA) procedure is used for the statistical analysis of the results.

The experiment uses a two-dimensional, deterministic function with the addition of an error term to represent a simulation model with two controllable inputs and a single response. This enables full control of the response surface characteristics. The experiment is a balanced full factorial block design with six
factors partitioned into two categories: surface-type factors and decision factors (see table 3.1). The surface-type factors are the modality of the surface, the distribution of the variance, the error distribution and the magnitude of the error variance. The decision factors are the number of "simulation model" runs (the run "budget" for the metamodeling) and the metamodeling technique used.

**Design of the experiment: factors**

The purpose of the experiment is to provide a set of surfaces, with a broad spectrum of properties, that will test the ability of the metamodeling techniques to perform well according to the performance measures previously defined. As mentioned, the experiment tests six factors.

The first factor, modality, has three levels: a unimodal, a bimodal, and a tetramodal response surface. It is of particular interest to look at the performance of nonparametric techniques when applied to surfaces of more than one optimum because it is under these conditions that classical search methods perform most poorly. If a metamodeling technique can start a search approach on the largest peak because of its ability to resolve the different optima in X-space and accurately assess the magnitudes of the responses, then the search technique may successfully complete the search.

The experiment uses surfaces produced by mathematical functions, rather than actual simulation models, as a means of controlling response characteristics; the deterministic part of the surface equation determines the modality of the surface. The three surfaces are designed to give well-defined optima, with one of the optima being the global optimum. Function values at the perimeter of the region [-1 ≤ x1, x2 ≤ +1] are set to zero to enable "basins of attraction" (see below) to be defined more easily. Although the optima are distributed roughly symmetrically over the region, the positions of the peak values are made asymmetric to some degree. The coordinates of the optima for the three surfaces are:

<table>
<thead>
<tr>
<th>Modality</th>
<th>x1</th>
<th>x2</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimodal</td>
<td>0.1926</td>
<td>0.1926</td>
<td>1.0000</td>
</tr>
<tr>
<td>Bimodal</td>
<td>0.6045</td>
<td>-0.5733</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>-0.5364</td>
<td>0.5843</td>
<td>0.7498</td>
</tr>
<tr>
<td>Tetramodal</td>
<td>0.6990</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>0.0140</td>
<td>-0.6891</td>
<td>0.8536</td>
</tr>
<tr>
<td></td>
<td>0.0140</td>
<td>0.6891</td>
<td>0.8536</td>
</tr>
<tr>
<td></td>
<td>-0.6747</td>
<td>0.0000</td>
<td>0.7069</td>
</tr>
</tbody>
</table>

*Chapter 3: A Performance-Based Study...*
Table 3.1: Factors and levels for the experimental design.

<table>
<thead>
<tr>
<th>Level Code</th>
<th>Factor code</th>
<th>i</th>
<th>j</th>
<th>k</th>
<th>l</th>
<th>m</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Modality (# of Optima of f(X))</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Magnitude of Var(ε)</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>Variance Model</td>
<td>Homogeneous</td>
<td>Heterogeneous</td>
<td>High</td>
<td>Homogeneous</td>
<td>Heterogeneous</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>f(ε)</td>
<td>Normal</td>
<td>Exponential</td>
<td>Normal</td>
<td>Exponential</td>
<td>Normal</td>
<td>Exponential</td>
</tr>
<tr>
<td></td>
<td>s, the # of Simulation Runs</td>
<td>16</td>
<td>25</td>
<td>36</td>
<td>49</td>
<td>81</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>Meta-modeling Technique</td>
<td>Kernel</td>
<td>Spline</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chapter 3: A Performance-Based Study . . .
Three-dimensional plots of the surfaces are given in figures 3.2, 3.3 and 3.4.

The homoscedasticity or heteroscedasticity of the variance over the surface is a second factor influencing the ability of the techniques to determine the true natures of the surfaces; hence, different variance models are used as factors as described in table 3.2. The behavior of kernel and spline smooths has been investigated in the literature mainly for the additive model with homogeneous variance, so it is of interest to see how the performance of the techniques is affected if the true model has heterogeneous variance.

The third factor, the random variable distribution \( f(\varepsilon_i) \), is chosen to be representative of the types of distributions met in practice. The Normal and Exponential distributions are selected as the levels of this factor, representing a symmetric and a right-skewed distribution respectively. Kernel- and spline-smooth behaviors are known for the normal error case, though the techniques are expected to be robust to the error distribution as long as the errors are independent. The issue here is with the relative performance of the two techniques.

The ability of the metamodeling techniques to produce an accurate and precise model of the surface is expected to be influenced by the amount of noise in the sample. The magnitudes of \( \text{Var}(\varepsilon) \), the last surface-type factor, are defined relative to a coefficient of variation of ten per cent, which we consider to be a significant level of error. We denote this level of error as medium and for convenience define levels denoted low and high as values an equal distance below and above the medium level. Let the standard deviation of \( \varepsilon \), be \( \text{s.d.}(\varepsilon) \), \( Y = f(X) \), where \( X \in S \) (the region of interest), and let \( Y_{\text{max}} \) and \( Y_{\text{min}} \) be the maximum and minimum responses of the deterministic functions over \( S \). Then the three levels of the factor magnitude of variance are defined as:

- **High** if \( \text{s.d.}(\varepsilon) = \frac{1}{6}(Y_{\text{max}} - Y_{\text{min}}) \)

- **Medium** if \( \text{s.d.}(\varepsilon) = \frac{1}{10}(Y_{\text{max}} - Y_{\text{min}}) \)

- **Low** if \( \text{s.d.}(\varepsilon) = \frac{1}{30}(Y_{\text{max}} - Y_{\text{min}}) \)

Note that using a value of zero for the standard deviation results in a deterministic model, which is useful in the validation of the metamodeling procedure.
Figure 3.2: Deterministic function, unimodal surface
Figure 3.3: Deterministic function, bimodal surface
Figure 3.4: Deterministic function, tetramodal surface
Table 3.2: Variance models

<table>
<thead>
<tr>
<th>Homogeneous</th>
<th>Heterogeneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_i = f(X) + e_i$</td>
<td>$Y_i = f(X) + e_i$</td>
</tr>
<tr>
<td>$X = \begin{pmatrix} X_1 \ X_2 \end{pmatrix}$</td>
<td>$X = \begin{pmatrix} X_1 \ X_2 \end{pmatrix}$</td>
</tr>
<tr>
<td>$L_{X_i} \leq X_1 \leq U_{X_i}$</td>
<td>$L_{X_i} \leq X_1 \leq U_{X_i}$</td>
</tr>
<tr>
<td>$L_{X_i} \leq X_2 \leq U_{X_i}$</td>
<td>$L_{X_i} \leq X_2 \leq U_{X_i}$</td>
</tr>
<tr>
<td>$Y_i \in \mathbb{R}$, $X \in \mathbb{R}^2$</td>
<td>$Y_i \in \mathbb{R}$, $X \in \mathbb{R}^2$</td>
</tr>
<tr>
<td>$E(e_i) = 0$, $\text{Var}(e_i) = \sigma_i^2 &lt; \infty$</td>
<td>$\text{Var}(e_i) = \sigma_i^2 &lt; \infty$</td>
</tr>
<tr>
<td>$\text{Cov}(e_i,e_j) = 0, i \neq j$</td>
<td>$\text{Cov}(e_i,e_j) = 0, i \neq j$</td>
</tr>
<tr>
<td>$e_i \sim \text{Normal}(0, \sigma_i^2)$</td>
<td>$e_i \sim \text{Normal}(0, \sigma_i^2 \text{Expo}(1.0))$</td>
</tr>
<tr>
<td>$e_i \sim \text{Expo}(\sigma) - \sigma$</td>
<td>$e_i \sim \text{Expo}(\sigma)(\text{Expo}(1.0)) - \sigma$</td>
</tr>
<tr>
<td>$f$ continuous over $(-\infty, \infty)$</td>
<td>$f$ continuous over $(-\infty, \infty)$</td>
</tr>
</tbody>
</table>

$i, j = 1, 2, \ldots, s$
The fifth factor, and first decision factor, is the number of simulation runs, $s$, available for use in synthesizing the surface. This is an important factor to consider in choosing a metamodeling technique as it is desirable for the technique to be an efficient performer as well as be accurate and precise. Asymptotic rates of convergence have been determined for the nonparametric methods but it is not clear which technique will be the best performer with small samples of data. Preliminary experiments have indicated that the smallest sample size for two input variables should be 16. The other levels of the factor have been chosen to allow a regular grid to be easily constructed.

The $s$ points that comprise the sample set of data from the model are taken at the intersections of a regular grid that covers the region of interest for the surface functions. The discussion of experimental designs for nonparametric metamodels in the literature is very limited when compared to that for parametric metamodels. Recent texts (Härdle 1990, Eubank 1988) discuss the use of most nonparametric techniques on a regular grid. Müller (1984) shows that the asymptotic optimal density for kernel smooths when using a global bandwidth (the same bandwidth over the region) is uniform when errors are independent and identically distributed (i.i.d.) and when edge effects are ignored. The question of experimental designs for nonparametric metamodels is an area for further research (Sargent 1991).

Finally, as shown in table 3.1, the two metamodeling techniques chosen for comparison are the Epanechnikov kernel smooth with constant bandwidth and the thin plate spline. These two techniques provide a contrast in that the spline is the equivalent of a variable bandwidth kernel smooth of high degree while the kernel is of low degree and constant bandwidth. It is hoped that the different characters of these methods will result in a variety of behaviors over the range of conditions set up in the experiment. Codes are not generally available for the kernel smooth, but the procedure is relatively easy to code and does not require a high degree of numerical sophistication. Microsoft FORTRAN v5.1 was used as the programming language and development environment. A multivariate version of the Epanechnikov kernel was implemented with the Rice (1984) boundary kernel. The optimal bandwidth was selected by a golden section search over the bandwidth for values of the cross-validation sum of squares. The computation of spline smooths is numerically sophisticated but an excellent set of FORTRAN and Ratfor codes has been produced by Gu (1989, 1992) and bundled into a package called Rkpack. Rkpack is used to compute a single smoothing parameter, thin-plate spline that uses generalized cross-validation to select the smoothing parameter.

Each factor-level combination or cell is replicated forty-one times. A preliminary experiment indicated that using blocking on the random number streams led to a loss of orthogonality in the design and compromised
any statistical analysis, so independent random number streams were used. The design ensures a power of 0.95 with an alpha risk of 0.05 for detecting differences in response of one standard deviation when comparisons between up to six treatments or factor level combinations are carried out (Bratcher, Moran and Zimmer, 1970).

**Design of the experiment: ANOVA model**

The response consists of a three-dimensional response surface generated by a metamodelling technique. Six performance measures, calculated for each response, make up the data to be analyzed to answer the research questions. An ANOVA is carried out using the values of the performance measures in the model:

\[
y_{ijklmnr} = \mu + \alpha_i + \beta_j + \gamma_k + \delta_l + \zeta_m + \eta_n + \text{interaction terms} + \varepsilon_{ijklmnr}
\]  

(3.7)

where the levels corresponding to the subscripts \(i, j, k, l, m, n\) are given in table 3.1, \(r = 1, 2, \ldots, 41\), and

- \(\mu\) is the grand population mean,
- \(\alpha_i, \beta_j, \gamma_k, \delta_l, \zeta_m, \eta_n\) are the factor effects
- \(\varepsilon_{ijklmnr}\) is the error associated with \(y_{ijklmnr}\).

Hypothesis tests are carried out on the appropriate performance measure data to provide answers to the four research questions. For each performance measure, the mean response for each cell is represented by \(\mu_{ijklmnr}\) where the subscripting follows the coding of the factors in table 3.1. The inclusion of all the levels of a factor in the mean is indicated by a dot subscript, for example:

\[\mu_{.,.,.,.} \text{ represents the mean of all cells that have factor } i \text{ at level 1.}\]

As cell means are being used in the hypothesis testing with reasonable sample sizes, the normality assumption is justified through a central limit theorem.

A study of the cell means is carried out to determine results that may be of practical significance and which could be incorporated into the knowledge base of a Classifier in the form of rules. A detailed explanation of the performance measures and the selected factors follows.

Chapter 3: A Performance-Based Study
Design of the experiment: metamodel performance measures

In order to answer the four research questions above, six performance measures are defined. Each performance measure involves the comparison of the true deterministic surface with the response surface produced by the metamodel. Analyses involve a statistical and a practical assessment of the significance of the results.

Research question 1 asks: "How successful is the metamodel at suggesting a starting point?" If the simulation response surfaces are represented by mathematical functions, the values of the true local optima can be easily determined. This allows a comparison of the starting position for search indicated by the metamodel, \( (\hat{X}_{opt}, \hat{Y}_{opt}) \), to the true optimum. Let the basin of attraction \( B(\cdot) \) of a true optimum \( Y_{opt} \) located at \( X_{opt} \) in the set \( S \) be defined as

\[
B(\hat{Y}_{opt}) = \left\{ X : \frac{\partial f(\lambda X + (1-\lambda)X_{opt})}{\partial x_i} \neq 0, \; i = 1, 2, \ldots, n, \; \lambda \in (0, 1) \right\}.
\]  
(3.8)

If the metamodel supplies an estimate of the location of the global optimum that is in the basin of attraction of the true global optimum, then this may be sufficient for a gradient-based search technique to achieve success. A metamodel that can specify the correct number of optima and locate them within the basin of attraction of the true optima will have desirable properties of accuracy and precision. A measure of the success of the metamodel in providing a starting point would indicate how near to the true global optimum the point is and whether the point is within the basin of attraction of the global optimum. Our first performance measure and the measure of success used here is:

\[
M = \frac{||\hat{X}_{opt} - X_{opt}||}{||X_B - X_{opt}||},
\]  
(3.9)

where \( || \cdot || \) denotes Euclidean distance, \( X_{opt} \) is the true global optimum and \( X_B \) is the point where the vector through the true global optimum and the metamodel estimate of the true global optimum intersects the boundary of the basin of attraction (see figure 3.5). The measure will be zero if the estimate is equal to the true global optimum and one if the estimate is on the edge of the basin of attraction. Scores of less than one denote a good starting place for a gradient based search, and scores greater than one a poor choice. A hypothesis test is carried out to determine whether the mean distance for one technique is significantly different from the other.
Figure 3.5: Geometric illustration of the components involved in performance measure M
Research questions 2a) and 2b) concern the fit of the metamodel to the true surface in terms of surface characteristics. The surface characteristics of greatest interest in optimization are the peaks and troughs of the surface. The usual global measures of goodness of fit such as MSE and maximum absolute deviation do not provide a good metric for fit of characteristics, because of their global nature. The question of a goodness-of-fit measure to use with a metamodel is an open research question (Sargent 1991). In figure 3.6, for example, metamodel A is clearly superior in its representation of the characteristics of the true function when compared to metamodel B, even though B is associated with the same MSE and a lower maximum absolute deviation (compare $|Y_a|$ to $|Y_b|$). If the metamodel is to provide a good starting point for a search and select a search based on the modality of the surface, then it is important for the metamodel to be able to reproduce the number, locations and magnitudes of the local optima present in the true surface. In this experiment, the true surfaces are known and so metrics can be constructed that compare the metamodel to the true surface. Given the true surface and the metamodel estimate of the true surface, two sets of local optima can be constructed:

$$M_o = \{(x_{m1}, y_{m1}), (x_{m2}, y_{m2}), \ldots, (x_{mj}, y_{mj})\} \tag{3.10}$$

$$T_o = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \ldots, (x_{ni}, y_{ni})\} \tag{3.11}$$

where $M_o$ is the set of local optima of the metamodel estimate of the true surface, $T_o$ is the set of local optima of the true surface, $i$ is the true number of local optima and $j$ is the number of local optima of the metamodel. The members of $T_o$ and $M_o$ are found by evaluating the metamodel functions on a dense grid and determining the grid local optima. The resulting grid coordinates are used as starting points for Hooke and Jeeves searches which determine the positions of the local optima to greater than single precision accuracy. In exploratory work with the metamodels produced by the kernel and spline techniques, it was discovered that the kernel smooths produced substantially more local optima than the spline smooth. A kernel smooth can be imagined to be a surface that is covered with small pimples, whereas the spline smooth is a highly smooth surface with deliberate local optima. Many of the local optima on the kernel metamodel were of extremely low relief, which complicated the selection of the optima to be paired with the true optima. This problem was resolved by computing the kernel smooth of the kernel smooth, which smoothed out the smallest, insignificant local optima and retained the larger local optima. The larger optima were used as the starting points for a Hooke and Jeeves search using the original Kernel Smooth function for functional evaluations used in the search. It was felt reasonable to match only metamodel local optima of a reasonable size to the true optima, and disregard those optima that are due to the noise in the data. For each point in $T_o$, the distance in X space is calculated to each member of $M_o$. 

Chapter 3: A Performance-Based Study...
Figure 3.6: Metamodel comparison.
Each member of $T_0$ is assigned points from $M_0$ that lie closer to it than to any other member of $T_0$. Each
member of $M_0$ is assigned to one member of $T_0$. This may result in some members of $T_0$ not being assigned
any points, while other members of $T_0$ have one or more points assigned to them. Of the points assigned to
a member of $T_0$, the one nearest in Euclidean distance to the member of $T_0$ is paired with it to form $P_k = \{(X_m, Y_m), (X_a, Y_a)\}$. The pairs of points $P_k$ form a new set $P_0 = \{P_1, P_2, \ldots, P_p\}$ with $p \leq i$ members. The number of points left unpaired in $T_0$, $U = i - p$, is one measure of the fit of the characteristics of the
metamodel to the true surface, with $U = 0$ indicating a good fit in terms of the number of optima. The set of
paired points allows the deviations between the points in $X$ space and $Y$ space to be calculated. We define
the second and third performance measures as follows:

$$\text{Location bias per optimum, } L = \frac{\sum_{k=1}^{p} |X_{mk} - X_{ak}|}{p}.$$  \hspace{1cm} (3.12)

$$\text{Bias in expected response per optimum, } R = \frac{\sum_{k=1}^{p} (Y_{mk} - Y_{ak})}{p}.$$ \hspace{1cm} (3.13)

The bias metrics provide a measure of how accurate the metamodels estimates of the location and
magnitude of the optima are. A perfect metamodel would have bias measures of zero. In practice, factors
such as the magnitude of the variance, sample size and the method of construction of the metamodel affect
the measures. A two-sided test is carried out to answer research question 2a by determining whether the
mean bias measure for one technique is significantly different from the mean bias measure for the other
technique for both bias measures.

Research question 2b is answered, for each level of modality, by testing the mean $U$ for one technique to see
if it is significantly different from the mean $U$ for the other technique. The lower the value of $U$, the more
faithfully the metamodel is reproducing the optima of the true surface. High values of $U$ indicate that the
metamodel is understating the complexity of the response surface by failing to represent the correct number
of local optima for the true surface. We record the raw number of metamodel optima for both the spline and
kernel metamodels and call this number $W$. $U$ and $W$ are our fourth and fifth performance measures.

Research question 3 concerns the goodness of fit of the metamodel surface to the true surface. A global
measure of fit and our final performance measure is the average squared error (ASE) determined at the
design points, defined as:
\[
\text{ASE} = \frac{\sum_{j=1}^{s} (y_{m_j} - y_{j})^2}{s},
\] (3.14)

where \(y_{m_j}\) is the metamodel estimate of the response and \(y_{j}\) is the true response at the same grid location. A small ASE means a better global fit. The ASE performance of the kernel and spline metamodels is compared to determine which technique performs significantly better than the other.

Research question 4 concerns the performance of the metamodel over a wide range of conditions. The answer to this question lies in the ANOVA. If the ANOVA has no or only a few low order interaction effects, then it may be possible to choose a metamodeling technique that has good performance over a wide range of conditions. If higher order interactions are prevalent, this would indicate that the performance of the metamodeling technique can only be deduced from an examination of the appropriate cell means.

Besides using the above measures to look at the relative performance of the two metamodeling techniques, it is of interest to use them to gauge their absolute performance. We define two measures of absolute performance:

1. An indicator variable that takes a value of true if the metamodeling technique has \(U = 0\) for all replications in a particular cell or group of cells, and a value of false otherwise. A value of true means that the metamodel always possesses the number of true optima in roughly the right locations, and is in a rough sense faithfully synthesizing the true surface. For example, if all experimental runs at a particular combination of levels of factors \(i, j, k, l, m\) and \(n\) have a value of \(U = 0\), then the metamodel is placing the correct number of optima in the correct part of the surface.

2. An indicator variable that takes a value of true if the metamodeling technique has a value of \(M\) that is more than two standard deviations less than one, and a value of false otherwise. A value of true means that the metamodel gives an estimate of the position of the global optimum that is almost certainly within the basin of attraction of the true optimum. A gradient search starting from the estimated position would then have a good chance of success in finding the global optimum.

RESULTS

This section is organized as follows. First, we indicate the amount and method of computation. We then detail the statistical analysis of the results for the six performance measures. The presentation of the results
is described and then the results are interpreted. Finally, implications of the results for a knowledge-based system and other metamodeling issues are given.

Computation

There are 504 factor level combinations, each of which was replicated 41 times, giving 20,664 results for each of the six performance measures. The experiment involved generating 10,332 spline smooths and an equal number of kernel smooths. The code was written and compiled in Microsoft FORTRAN v5.1 (1993) and run on an Intel-based 80486 66MHz PC clone. Statistical analysis was carried out in a mainframe environment using SAS (1985).

Statistical analysis

Although the intention in this analysis is to consider each of the response measures independently, a MANOVA was carried out on all the results to test the two-sided null hypothesis:

\[ H_0 : \text{All means for all responses do not differ.} \]

All four MANOVA tests, i.e., Wilks’ Lambda, Pillai’s trace, the Hotelling-Lawley trace and Roy’s maximum root, returned p-values of 0.0001, and so the null hypothesis was rejected at the \( \alpha = 0.05 \) significance level. We conclude that there are significant differences between the cell means for some of the responses.

We proceeded to examine the individual ANOVA for each of the six responses. In each case, the overall F-test for the null hypothesis that all cell means were equal was rejected at the \( \alpha = 0.05 \) significance level, with all tests returning p-values of 0.0001. The F-test values for each of the ANOVA model terms were then inspected, and it was found that interaction terms up to order five were significant at \( \alpha = 0.05 \) for each response. It was concluded that further analysis would have to be at the level of groups of cell means and use multiple comparison procedures. At this point in the analysis the validity of the assumptions of normality and homogeneity of variance was investigated. The overall F-test is robust to departures from these assumptions for balanced designs, but multiple comparison tests are not robust, with the exception of the Scheffé procedure which uses the F distribution (Neter, Wasserman and Kutner, 1990, pp. 623-624). Plots of the individual cell means versus their standard deviations showed substantial variation in location and scale of the cell data for each response. The cell means for the kernel and spline levels were tested for
the hypothesis of homogeneity of variance using Levene's test and for the hypothesis of normality using the Shapiro-Wilk test. The results of these tests were mixed for all responses, with some cells failing to reject the null hypothesis and others rejecting it at the \( \alpha = 0.05 \) level of significance; hence, a conservative approach to the choice of multiple comparisons procedures was adopted.

As five-way interactions were significant in each ANOVA, it was decided to group the cell means according to the factor-level combinations that were in the five-way interaction. The factor that was neglected was error distribution. Thus 252 groups of cell means were constructed and the multiple comparison procedure was to test 126 pairs of cell groups for the two-sided hypothesis:

\[
H_0: \mu_\text{kernel} = \mu_\text{plane}
\]

For each response, five multiple comparison tests were carried out using the Bonferroni procedure for a family confidence coefficient of 0.975. The five tests were the Scheffé multiple comparison test, two t-tests, each using a different variance estimate, and two nonparametric procedures, one using Wilcoxon scores, the other using the Median test. Medians replaced the means in the hypotheses used for the nonparametric tests. The two estimates of variance used by the t-tests were the MSE and the pooled sample variance of the cells.

The results of the multiple comparison tests for each response showed a great deal of agreement. In particular, no pair of tests for a hypothesis rejected the null hypothesis and came to opposite conclusions. There were sets of results in which the null hypothesis was rejected by one or more tests while it was not rejected by other tests. The final results were taken to be the results of the multiple comparison procedure using the Median test. This procedure is the most conservative of those used in terms of the assumptions that it makes (independent data, same shape distribution) and its results were representative of the results from the other procedures in general.

The results are presented in two sets of tables. Table 3.3 summarizes the results for the absolute performance of the two metamodelling techniques. Table 3.4 summarizes the results of the multiple comparison tests on the relative performance of the two metamodelling techniques for all six criteria. Results are presented in this way to convey the general performance trends of the metamodells. The two sets of tables follow the same method of presentation. Each entry in the tables consists of a letter that corresponds to a result for a pair of cell groups. For example, a pair of ANOVA cell groups would be:
Table 3.3: Tables showing absolute performance results for spline and kernel metamodeling techniques

<table>
<thead>
<tr>
<th>M - The Distance from Global Optimum to Basin Edge</th>
<th>SAMPLE SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td>MODALITY</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>L</td>
</tr>
<tr>
<td></td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>H</td>
</tr>
<tr>
<td></td>
<td>L</td>
</tr>
<tr>
<td>MODALITY</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>H</td>
</tr>
<tr>
<td></td>
<td>L</td>
</tr>
<tr>
<td>MODALITY</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>H</td>
</tr>
</tbody>
</table>

Table 3.3(a): Absolute performance with respect to starting point (research question 1)

<table>
<thead>
<tr>
<th>U - The Number of Unassigned Optima</th>
<th>SAMPLE SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td>MODALITY</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>L</td>
</tr>
<tr>
<td></td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>H</td>
</tr>
<tr>
<td></td>
<td>L</td>
</tr>
<tr>
<td>MODALITY</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>H</td>
</tr>
<tr>
<td></td>
<td>L</td>
</tr>
<tr>
<td>MODALITY</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>H</td>
</tr>
</tbody>
</table>

Table 3.3(b): Absolute performance on surface characteristic misinformation (research question 2b)
Table 3.4: Tables showing relative performance results for spline and kernel metamodeling techniques

<table>
<thead>
<tr>
<th>M - The Distance from Global Optimum to Basin Edge</th>
<th>SAMPLE SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td>L</td>
<td>KB</td>
</tr>
<tr>
<td>M</td>
<td>BK</td>
</tr>
<tr>
<td>H</td>
<td>BB</td>
</tr>
<tr>
<td>L</td>
<td>KK</td>
</tr>
<tr>
<td>M</td>
<td>KK</td>
</tr>
<tr>
<td>H</td>
<td>KK</td>
</tr>
<tr>
<td>L</td>
<td>B</td>
</tr>
<tr>
<td>M</td>
<td>KB</td>
</tr>
<tr>
<td>H</td>
<td>KK</td>
</tr>
</tbody>
</table>

Table 3.4(a): Relative performance with respect to starting point (research question 1)

<table>
<thead>
<tr>
<th>L - Location Bias per Optimum</th>
<th>SAMPLE SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td>L</td>
<td>SB</td>
</tr>
<tr>
<td>M</td>
<td>BB</td>
</tr>
<tr>
<td>H</td>
<td>BB</td>
</tr>
<tr>
<td>L</td>
<td>KK</td>
</tr>
<tr>
<td>M</td>
<td>KK</td>
</tr>
<tr>
<td>H</td>
<td>KK</td>
</tr>
<tr>
<td>L</td>
<td>BK</td>
</tr>
<tr>
<td>M</td>
<td>KK</td>
</tr>
<tr>
<td>H</td>
<td>KK</td>
</tr>
</tbody>
</table>

Table 3.4(b): Relative performance in capturing essential features of the surface (research question 2a)

<table>
<thead>
<tr>
<th>R - Response Bias per Optimum</th>
<th>SAMPLE SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td>L</td>
<td>SS</td>
</tr>
<tr>
<td>M</td>
<td>SS</td>
</tr>
<tr>
<td>H</td>
<td>BB</td>
</tr>
<tr>
<td>L</td>
<td>KB</td>
</tr>
<tr>
<td>M</td>
<td>BS</td>
</tr>
<tr>
<td>H</td>
<td>SS</td>
</tr>
<tr>
<td>L</td>
<td>BS</td>
</tr>
<tr>
<td>M</td>
<td>SS</td>
</tr>
<tr>
<td>H</td>
<td>SS</td>
</tr>
</tbody>
</table>

Table 3.4(c): Relative performance in capturing essential features of the surface (research question 2a)

Chapter 3: A Performance-Based Study...
Table 3.4, cont'd.

<table>
<thead>
<tr>
<th>M O D A L I T Y</th>
<th>1</th>
<th>V A R I A N C E</th>
<th>S A M P L E S I Z E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
<td>SS</td>
<td>SB</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>BB</td>
<td>SS</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>BS</td>
<td>SS</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>SS</td>
<td>SS</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>SS</td>
<td>SS</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>SS</td>
<td>SS</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>SS</td>
<td>SS</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>SS</td>
<td>SS</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>SS</td>
<td>SS</td>
</tr>
</tbody>
</table>

Table 3.4(d): Relative performance in capturing essential features of the surface (research question 2b)

<table>
<thead>
<tr>
<th>M O D A L I T Y</th>
<th>1</th>
<th>V A R I A N C E</th>
<th>S A M P L E S I Z E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
<td>BB</td>
<td>BB</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>BB</td>
<td>BB</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>BB</td>
<td>BB</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>BB</td>
<td>BB</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>BB</td>
<td>BB</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>BB</td>
<td>KK</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>KK</td>
<td>BB</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>KK</td>
<td>BB</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>KK</td>
<td>BB</td>
</tr>
</tbody>
</table>

Table 3.4(e) Relative performance on surface characteristic misinformation (research question 2b)

<table>
<thead>
<tr>
<th>M O D A L I T Y</th>
<th>1</th>
<th>V A R I A N C E</th>
<th>S A M P L E S I Z E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
<td>SS</td>
<td>SS</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>SS</td>
<td>SB</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>SS</td>
<td>BB</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>KK</td>
<td>SS</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>BB</td>
<td>SS</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>BB</td>
<td>SB</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>BB</td>
<td>KK</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>BB</td>
<td>KK</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>BB</td>
<td>KK</td>
</tr>
</tbody>
</table>

Table 3.4(f): Relative performance in overall fit of the surface (research question 3)

Chapter 3 : A Performance-Based Study . . . 55
Modality = 1, Magnitude of Variance = low, Variance Model = Homogeneous, Sample Size = 16, Metamodelling Technique = Kernel

Modality = 1, Magnitude of Variance = low, Variance Model = Homogeneous, Sample Size = 16, Metamodelling Technique = Spline.

Each entry in a table gives the result of evaluating an indicator variable (table 3.3) or the result of a multiple comparison test for two ANOVA cell groups (table 3.4). The table cells have two letters per cell. The letter in the first position denotes the result for the Variance Model = Homogeneous level while the letter in the second position denotes the result for the Variance Model = Heterogeneous level. The other level combinations that complete the description of a group can be read off the table row and column headings. In table 3.3, the letters are interpreted according to the following key (recall that 'true' means that the metamodel estimate of the global optimum is almost certainly within the basin of attraction of the true global optimum in table 3.3(a); in table 3.3(b) "true" indicates that \( U = 0 \) for all 41 replications in that cell):

- B -- Both metamodels give a value of true
- K -- The kernel metamodel gives a value of true
- S -- The spline metamodel gives a value of true
- N -- Neither of the metamodels give a value of true

In table 3.4, the letters are interpreted according to the following key:

- B -- Both techniques performed equally well
- K -- The kernel metamodel performed significantly better than the spline technique
- S -- The spline metamodel performed significantly better than the kernel technique

The purpose of the table cell shading is to highlight the regions of differing metamodel performance.

**Interpretation of the results**

The factor-level space in table 3.3(a) can be divided into two regions. The lower left-hand corner of the table is the region where neither technique places its estimate of the global optimum in the basin of attraction with any certainty. The rest of the table is the other region. The boundary between these two regions is shown as a thin black line in table 3.4. Similarly, table 3.3(b) can be divided into two regions.

**Chapter 3: A Performance-Based Study**
The upper right corner of the table is the region in which at least one of the techniques faithfully reproduces the correct number of optima in the correct area of X-space. The lower left corner is the second region. Note that the faster rate of convergence of the spline smooth to the true function is evident in this table. The boundary between these two regions is indicated in table 3.4 by a thick black line. The two lines mark, with differing degrees of severity, the boundaries between the two regions of the tables. We define the alias region as the lower left corner of the tables. In this region the metamodels are aliased representations of the true surface, i.e., the true surface is not faithfully represented by the metamodel. The top right corner of the tables is defined as the fidelity region, where the metamodels do, in a rough sense, faithfully represent the true surface. The relative performance results for each research question will be interpreted separately for the two regions.

Research question 1 may be asked by the user whose primary objective is to optimize the response. Table 3.3(a) shows that there are large areas of the factor space over which both techniques can supply a good starting point. Spline metamodels are the only choice when the modality of the response surface is high and the sample size is just sufficient. The advantage of spline metamodels when enough data is available is most probably due to its equivalence to a high-order kernel function (Silverman, 1984), which has the property of a faster asymptotic convergence rate than a low-order kernel. Neither technique gives a faithful rendition of the true surface under the conditions of low sample size and high response activity.

On examining table 3.4(a), the results for the relative performance of the two metamodeling techniques in the fidelity region are mixed. The kernel metamodeling technique is seen to be a relatively good performer when the sample size is low or when the sample size is large and the magnitude of the variance is low. The regions of medium sample sizes and high modality appear to be the province of the spline metamodel. Kernels dominate the low sample size table cells in the alias region.

The effectiveness of a technique at capturing the essential features of the surface is summarized in tables 3.4(b), 3.4(c), 3.4(d) and 3.4(e). The relative ability of the techniques to place the optima correctly in X-space follows the same pattern as the results for indicating a good starting point. In terms of fitting the response at the optima, the spline metamodel generally performs better than the kernel metamodel. The key factor in the spline metamodel's ability to outperform the kernel metamodel is its equivalence to a variable bandwidth kernel, which allows it to have less bias in its estimate of the response at optima. The kernel metamodel is on an equal footing with the spline metamodel under conditions of unimodality with medium or high variance and generally when the sample size is large.
The entries in table 3.3(b) clearly define the alias and fidelity regions. Spline metamodels perform well at high modality of the response function and at medium to high sample sizes. Analysis of the data behind table 3.4(d) indicates that the kernel metamodel nearly always has more optima than the spline metamodel. A large number of these optima are of very low relief. If the kernel metamodel is kernel smoothed again, it still has more optima than the spline smooth. This property of kernel metamodels may well make them unattractive for use in procedures where search is performed on the metamodel surface. The many local optima may require a large number of starting points for a gradient search routine in order to guarantee its success in finding the metamodel's global optimum. However, table 3.4(e) indicates that both techniques manage to place the correct number of optima in roughly the correct place in the fidelity region. Note that the spline metamodel has superior performance when the sample sizes are medium and the modality of the response function is high in the fidelity region. In the alias region of table 3.4(e) the kernel metamodel's ability to synthesize more optima gives it a performance advantage.

The question of the fit of the metamodel to the true surface is answered by the results in table 3.4(f). The spline metamodel's equivalence to a variable bandwidth kernel is suggested as the reason for its relatively superior performance.

To answer the fourth and final research question we look at the fidelity region results for each performance measure independently. It is easy to conclude that spline metamodel is relatively more robust than the kernel metamodel from the frequency of its symbol of superior performance in the tables.

The performance of the metamodel techniques in the alias region of the tables is important as it has implications for the implementation of sequential design procedures. It is very likely that such a procedure would start with a design that would result in a metamodel of an alias of the true surface. As the sequential procedure adds points to the initial design, it is of interest to know when the metamodel has moved into the region of fidelity. The region of fidelity is reached when the number of optima is correct and the positions of the optima are roughly correct. The optima-producing behavior of kernel metamodels makes it difficult to track significant optima, while the relative stability of the kernel position estimates makes it difficult to see the transition to the fidelity region by tracking changes in the positions. The variable bandwidth and faster convergence of spline metamodels give them the ability to "chase" the response. It may be possible to detect when the response "settles down" by tracking the properties of the spline metamodel as points are added. When the properties of the metamodel are stable it may indicate that the transition from the alias region to the fidelity region has occurred.
Implications of the results for a knowledge-based approach to simulation optimization

The results contain information that could be used to select a nonparametric metamodel approach over another search approach. Moreover, it could enable a choice to be made between using a spline metamodel, a kernel metamodel, or both. Table 3.3 gives information that could be used to select a uniform-design based nonparametric metamodel if some idea of the complexity of the response was known or if the run budget was known. This information could easily be put into a knowledge base in the form of If-Then rules. Similarly, information regarding the relative performance of the metamodeling techniques could be stored. For instance, if it were known that the variance was low and that the position of the global optimum was of high interest, then a rule such as

IF Variance = Low and Global Position = Priority 1 THEN Execute spline and kernel routines

would be fired. Both metamodels would give an estimate of the position of the global optimum and that information would be used to fire another rule that dictates the strategy from that point on. There is sufficient evidence of the superiority of the spline metamodel that it could reasonably function as a default metamodel in cases where little or no information was available on the response.

The absolute performance tables make it clear that the number of runs required for a uniform design to support a desired level of metamodel performance is quite high. A requirement of 25 runs to model two inputs would translate into a 3,125 run requirement to model five inputs. It is clear that the attractiveness of using metamodels for simulation optimization would increase if sequential design procedures were available. These procedures would allow the knowledge-based system to start with a small uniform design and add design points sequentially according to some criterion until the metamodel had moved from the alias region into the fidelity region. We know of only one attempt to produce a sequential design procedure for nonparametric regressors (Faraway, 1990), which deals with the issue for kernel smoothes.

EXAMPLES USING SIMULATION MODELS

Example 1.

The first example is a simple inventory model taken from a paper by Greenwood, Siochi, and Rees (1995). The model is a continuous-review stochastic EOQ model with both lead time and interarrival demand being
random variables. The objective of the simulation optimization is to determine the values of the two controllable variables, order quantity (Q) and re-order point (R), that give the inventory policy with the lowest total cost. Ordering cost, holding cost and backorder cost are the components of the total cost modeled in this example. Greenwood et al. parameterize and constrain their model as follows:

Parameters

Inter-demand time ~ Gamma(1, 0.2)
Lead time ~ Normal(6, 2)
Backorder Cost = $5/item/year
Holding Cost = $10/item/year
Ordering Cost = $50/order
1 year = 250 days
Demand Quantity = 1 item

Constraints

0 < Q <= 400
-400 <= R <= 0
| R | <= Q.

Note that the constraints restrict the model decision variable space to a triangular region.

The simulation was coded in ProModel for Windows. A one-year warm-up period was used followed by a four-year run. The total cost was computed over a whole number of order cycles during the run period, with costs starting to accumulate at the start of the first cycle after the warm-up period of 250 days and finishing at the end of the last complete cycle in the run period of 1000 days. As all parts of the decision-variable space are of equal interest, an equi-spaced design was used over the triangular region with one run to be carried out at each design point, resulting in a sample size of 27 runs. The small sample size was chosen because the inventory problem was thought to be unimodal, and it can be seen from table 3.3 that both metamodeling techniques are successful at modeling under these conditions. As the initial experiment is likely to be a pre-cursor to further experimental runs, it is of interest to be able to determine the likely position of the global minimum in decision (X-space) space. Table 3.4(b) indicates that the spline will be the better performer if the response variance can be considered low, and it is as good as the kernel method at higher variances. As the costs used in the model are deterministic and the random part of the response is
due to an aggregate of random variables, it is likely that the variance of the response will be low. The data from the experiment are shown on a scatter plot in figure 3.7. Figure 3.8 shows a contour plot of the spline smooth metamodel of the response surface. It clearly shows an operating area or area for further runs running from co-ordinate \( (Q = 125, R = -60) \) to co-ordinate \( (Q = 280, R = -170) \), roughly defined by the $2800 contour. Additional points placed on the periphery of this region would be helpful in more accurately defining a region of low cost operation. The kernel smooth metamodel is also worth examining in this situation as the variance of the response does seem to be perhaps more than low (figure 3.7). Figure 3.9 shows the kernel smooth metamodel. It has a shallower minimum than the spline model and does not have such steep sides to the minimum. Both place the minimum in the same area, but the region indicated by the kernel smooth is more circular in shape than the region indicated by the spline smooth. Relying on the X-space performance advantage of the kernel smooth, more runs might be added along the $3100 contour running from \([170, -120]\) to \([270, -180]\). The small cost differential involved here would have to be weighed against the costs of further simulation runs.

**Example 2**

The second example is also based on an inventory system. The model is a stochastic inventory model with non-instantaneous receipt of orders. Orders are received in \( B \) batches one day apart with the first batch arriving after the lead time, which is stochastic. Times between demands on the inventory and the quantities of those demands are also stochastic. The objective of the simulation optimization is to determine the number of batches, \( B \), and the order quantity, \( Q \), (the decision variables) that give the lowest total cost inventory policy. Note that \( B \) and \( Q \) are integer variables, but that \( Q \) is more fine grained than \( B \), and can be modeled as continuous. \( B \) can be thought of as a measure of production rate, and so its integer character is not as coarse as it seems. The model is parameterized and constrained as follows:

**Parameters**

- Inter-demand time \( \sim \) Gamma(0.4,0.4)
- Lead time \( \sim \) Binomial (5,0.8)
- Demand quantity \( \sim \) Poisson(5)
- Backorder cost = $10/item/year
- Holding cost = $5/item/year
- Setup cost = \$25 if batch size \( \geq 100 \)
  \$15 if batch size < 100
Figure 3.7: Scatter plot of raw data from example 1.
Figure 3.8: Spline smooth metamodel of data for example 1.
Figure 3.9: Kernel smooth metamodel of data from example 1.
Constraints

\[ 100 \leq Q \leq 600 \]
\[ B \in \{1, 2, 3, 4, 5, 6\}. \]

The measurement of total cost, the warm-up period and the simulation run length were as in example 1. The surface is not known to be unimodal, but it is in the family of inventory models, which are not usually multimodal. Thus, a small equi-spaced design was chosen with one run to be carried out at each design point, i.e., at combinations of \( Q = 100, 200, 300, 400, 500, 600 \) and \( B = 1, 2, 3, 4, 5, 6 \). This resulted in a sample size of 36. Using table 3.3, if the surface is unimodal, then kernel and spline smooth metamodels perform equally well. If there is interest in establishing locations for future experimental runs, the kernel smooth provides the more accurate results in X-space (table 3.4(b)). Raw data from the experiment are shown on a scatter plot (figure 3.10). The kernel and spline smooths are shown in figures 3.11 and 3.12 respectively. Note that the kernel smooth has a minimum at \( Q = 250, B = 1 \), while the spline smooth places the minimum at \( Q = 325, B = 1 \). The kernel suggests that runs should be carried out at \( Q = 250, B = 1 \) and \( Q = 250, B = 2 \) to validate the indicated minimum. The spline indicates that runs are needed at \( Q = 350, B = 1 \), and \( Q = 350, B = 2 \) and would ignore the area indicated by the kernel smooth. Both metamodels suggest runs be carried out at \( Q = 350, B = 3; Q = 350, B = 4; Q = 450, B = 3 \) and \( Q = 450, B = 4 \). The spline would indicate that additional runs be carried out at higher values of \( Q \) and \( B \) because of the elongated minimum in that region. It would be better if there were a more formal criterion for use in indicating further design points -- we have found only one paper (Faraway, 1990), which deals with the issue for kernel smoothing, but have begun to develop a sequential procedure of our own (Keys and Rees, 1995).

**CONCLUSIONS**

Nonparametric metamodeling techniques are seen as promising for use in simulation optimization (Barton, 1992, 1994). Two major nonparametric regression techniques, spline smoothing and kernel smoothing, are compared by evaluating their metamodeling performance using six measures. The measures are calculated for each technique over a wide range of response conditions in a full factorial experiment. A statistical analysis of the results is carried out to determine significant performance differences between the two techniques and the outcome of the analysis is expressed in tabular form. The tables allow the regions of factor space in which a technique performs well to be seen.

**Chapter 3 : A Performance-Based Study**
Figure 3.10: Scatter plot of raw data from example 2.
Figure 3.11: Kernel smooth metamodel of data from example 2.

Chapter 3: A Performance-Based Study
Figure 3.12: Spline smooth metamodel of data from example 2.
Spline metamodels are found to be relatively more robust than kernel metamodels over all the performance measures considered. If fit to the response is considered important, the spline generally offers better performance than the kernel metamodel. When the placement of response surface characteristics in X-space is important, then kernel metamodels are able to outperform spline metamodels. The use of kernel metamodels should be considered at low sample sizes and at high sample sizes and especially when the magnitude of the variance is low.

An important characteristic of kernel smooths is the relatively large number of local optima they synthesize compared to a spline smooth (table 3.4(d)). Under circumstances where both techniques fail to represent the response surface faithfully (the surface produced is an alias), kernel smooths are able to provide candidate points for the true optima when spline smooths are not. Thus a kernel smooth looks as if it is performing better under measures such as L and U, when in fact its performance falls short of providing a faithful representation of the response surface.

A knowledge-based simulation optimization system could easily capture the results presented in tables 3.3 and 3.4 in the form of If-Then rules. The rules would decide whether the optimization problem required the selection of a metamodeling technique, and if so, whether to employ one or several of the techniques depending on the goals of the user.

For the two-input, single-response case, 16 runs are needed to obtain a rough representation of the response surface for a unimodal function, 36-49 runs are required for the bimodal function and 81-121 runs are required for the tetramodal function when uniform designs are employed (table 3.3(a)). If the activity level of the surface is high, then the uniform design has a high cost in terms of simulation runs. For simulation models in higher dimensions, this expenditure increases with the power of the dimension and effectively rules out the use of the techniques with a uniform design. Sequential design procedures are seen as a very important research topic.

The results of the simulation experiments were supported by inspection of kernel and spline metamodels of two response surfaces produced by inventory models of differing complexity. The characteristics expected of the two surfaces were as predicted by the parameters of the simulation optimization problem.

This research has followed a lead indicated by Barton (1992). The results obtained here indicate that nonparametric techniques are capable of successfully metamodeling a wide range of response functions. Depending on the particular characteristics of the response surface that are of interest, the spline or kernel
or both techniques may be employed to synthesize a metamodel. Uniform designs are shown to be expensive for simulation optimization, and sequential design methods should be investigated. The use of nonparametric methods for problems with more inputs would seem to be restricted to dimensions no larger than four or five unless simulation runs are cheap or parallel processing is available and viable.

FUTURE RESEARCH

If spline and kernel smoothing are to be employed in simulation optimization, the most pressing question is how to allocate design points using other than a uniform design. A sequential design procedure, as is available for RSM, would allow a minimal initial uniform design to be augmented one or more design points at a time, leading to a more efficient metamodeling technique.

A major problem in modeling surfaces where the nature of the response is largely unknown is that of aliasing. Using too small a sample of design points generates a metamodel that does not faithfully represent the character of the true response surface. Future research needs to determine measures of validity of the metamodel. This issue also has a bearing on the specification of a stopping rule for a sequential design method.

Although the experiment carried out here was insightful, the conclusions need to be supported by further work using problems in higher dimensions, i.e., up to four or five dimensions. Such an extension could use uniform designs in the pursuit of supporting data and possibly compare selected results to a sequential design method (Faraway, 1990).

The role of nonparametric metamodels needs to be explored in the context of other searches and especially in the context of knowledge-based approaches to simulation optimization. Questions of interest would center around whether nonparametric techniques should be used on their own or in conjunction with specific other search techniques. The influence of the simulation run budget on the use of nonparametric techniques and other searches is a particularly interesting area of further research.

Nonparametric metamodels can provide much information about the simulation response function that could be used by a machine learning mechanism within a knowledge-based simulation optimization system (KBSOS). Research is needed regarding the types of information that can be captured and used to enhance the performance of the KBSOS.

Chapter 3: A Performance-Based Study
CHAPTER 4

IMPLEMENTING NONPARAMETRIC METAMODELING TECHNIQUES IN A CLASSIFIER KNOWLEDGE-BASED SIMULATION OPTIMIZATION SYSTEM

INTRODUCTION

Simulation optimization and a classifier knowledge-based approach

Simulation is a modeling technique that is often employed to gain insights into systems that are analytically intractable. A simulation model can provide values for system responses of interest if the values of the input parameters are given and the model is run. The model does not give any indication of the values of the input parameters that are required to maximize the response, as is desired in simulation optimization. The usual approach to solving the simulation optimization problem is to use a single search strategy. For example, Response Surface Methodology (RSM) (see Box and Draper, 1987) is a search strategy that uses first- and second-order polynomial models in conjunction with a gradient-based search to locate the optimal input settings. Jacobsen and Schruben (1989) conclude that particular single strategy approaches tend to produce good results for certain types of simulations. Azadivar (1992) and Safizadeh (1990) review simulation optimization techniques and conclude that there is a need for programs to assist with or automate the simulation optimization process. Smith (1976) constructed an early simulation optimization system but it was integrated with the simulation model and limited in the techniques that it used. Azadivar and Safizadeh perceive the support system to be separate from the simulation model, which is treated as a black box. O'Keefe (1986) presents a taxonomy of expert system/simulation model combinations. Fox, Husain, McRoberts and Reddy (1989) develop a knowledge-based simulation system which assists with the development of the simulation model and also controls the use of the model to achieve the goals of a user. The system does not aim to be a simulation optimization system per se, and so does not embody any sophisticated optimization procedures. Crouch, Greenwood and Rees (1993) describe a Classifier Knowledge-Based Simulation Optimization System (CKBSOS) which integrates scientific methods of analysis with heuristics contained in knowledge bases to provide a flexible multi-strategy approach to the solution of simulation optimization problems. An essential element of their system is a Classifier. The Classifier has two main functions, surface synthesis and surface characterization. A sample of responses obtained at various settings of the inputs of the simulation model is used by a metamodeling technique to synthesize a response surface. Classification of the simulation response surface is carried out by

Chapter 4: Implementing Nonparametric Metamodeling...
characterizing the surface in terms of attributes that the heuristics in the strategy-selector knowledge base can use to decide on a search strategy.

Crouch, Greenwood and Rees (1993) indicate that surface synthesis may be carried out using a number of methodologies, including regression metamodeling, kernel smoothing and neural networks. They state that several different methods should be used co-operatively to obtain the best synthesis of a particular response surface. However, in their example problem a neural network approach is used in isolation. The architecture for the selection of a method of surface synthesis and the impact of such an architecture on the knowledge-bases in the CKBSOS are left unspecified.

**Nonparametric methods**

Sargent (1991) is an early paper discussing the role of metamodels, i.e., models of models, in simulation optimization. He defines a metamodel as a model that is an abstraction of the simulation model. Sargent presents lists of research issues and desirable properties of metamodels. Barton (1992, 1994) concentrates on reviewing metamodeling techniques and points out that polynomial metamodels are unable to provide a global fit to response functions that are of an arbitrary shape. His review emphasizes nonparametric metamodeling techniques. Nonparametric metamodels use functions with a local basis that model the response in terms of a weighted sum of surrounding responses. Barton (1992) concludes that nonparametric techniques have the most promise for creating successful metamodels in simulation optimization. Barton (1994) suggests that more examples of the use of these methods are needed, along with computational comparisons of the methods. This paper will consider the use of two nonparametric techniques, kernel smoothing and spline smoothing, in a CKBSOS. Although spline smoothing can be described in terms of a kernel smooth model (Silverman 1984), each technique has a different method of implementation and different requirements for implementation. This paper will define the requirements of each nonparametric metamodeling technique and specify how the requirements are to be built into a classifier.

The contributions of this research are to describe how nonparametric methods can be used cooperatively within a CKBSOS and specify the architecture, rules and procedures required for their implementation. In particular, a taxonomy to aid in the automatic selection of an appropriate nonparametric metamodeling technique for surface synthesis is presented, and a procedure for iterative surface synthesis using a nonparametric method is developed and illustrated in an example.

Chapter 4 : Implementing Nonparametric Metamodeling...
The rest of this paper is organized as follows: First, the Classifier of Crouch, Greenwood and Rees (1993) ("the Classifier") is reviewed. Second, both nonparametric techniques are defined and their properties outlined. Next, the integration of the methods into the Classifier architecture is specified and a taxonomy of search procedures is defined. Finally, an example of the Classifier using nonparametric techniques for surface synthesis is given.

THE CLASSIFIER

The purpose of the Classifier is to provide information that can be used to select the search technique that will perform best, given the present characterization of the surface. When a search technique becomes invalid or ineffectual, the Classifier is re-invoked, the sample of data used to construct the present surface is augmented by further data, and the process is repeated. The iterative nature of this "Classify-then-Search" process is shown in figure 4.1. The system is driven by the need to obtain the key characteristics of the response surface that are known to differentiate the performance of the search techniques. The possibility of the surface possessing characteristics that do not result in a mapping to an existing search technique is considered. Characteristics are obtained from the user (via the user module) as well as from an analysis of the surface synthesis. A typical run through the Classifier during the solution of an optimization problem might be as follows:

1. The user supplies the necessary information to start the process: the number of simulation runs in the budget, the nature and range of the input parameters, constraints on the region to be searched, termination conditions and the simulation model itself.
2. The system decides, based on the user information, how to collect a sample of responses over the region of interest for use by the surface synthesizer.
3. A metamodel is constructed from the sample data by using a neural network. (Other metamodeling techniques are mentioned as possibilities as well.)
4. An analysis of the metamodel is carried out using the neural network in recall mode and the characteristics of the surface are determined.
5. Rules are used to select an appropriate search technique given the values of the characteristics obtained.
6. The search technique is employed until some measure of validity determines the technique is no longer valid, or the optimum is found, or the goal is achieved, or the number of runs allowed is exhausted.
Figure 4.1: The Classify-Then-Search process (Crouch, Greenwood and Rees, 1993)
7. If the search technique is declared invalid or ineffectual, then more runs are carried out and a revised metamodel is synthesized. The additional runs are usually an augmentation of the previous set of runs.
8. The process is repeated from step 3.

Step 5 utilizes the concept of $\chi$ space. Problem characteristics are labeled $\chi_i$ and each characteristic vector $\chi$ maps to a performance vector of search techniques. The search technique selected is the one with the maximum performance measure.

Note that the example in Crouch, Greenwood and Rees (1993) defaults to the use of a neural network approach for surface synthesis (step 3), though the conceptual framework outlined recognizes that selection of a synthesizer should take place, that the selection should depend on the nature of the simulation surface, and that several different methods should be used co-operatively.

While we accept the general architecture of the knowledge-based approach to simulation optimization of Crouch, Greenwood and Rees, we prefer to view surface synthesis as a search technique in itself. This paper will look at the integration of nonparametric search techniques into a process that differs from that of Crouch, Greenwood and Rees by having the techniques function as stand-alone searches selected by considering the characteristics of the problem. We do not rule out the use of nonparametric techniques in a multi-strategy approach to optimization. If the nonparametric technique were able to define the neighborhood of the global optimum, then it may be advantageous, i.e., efficient, to switch to a local optimization technique. However, the investigation of this issue is left to future research.

NONPARAMETRIC TECHNIQUES

Nonparametric metamodeling techniques have the advantage of not being bound to a fixed global form. This allows them to model arbitrarily shaped functions more accurately than polynomial-based metamodels. A review of metamodeling techniques for simulation was carried out by Barton (1992). Of the nonparametric techniques (or nonparametric regressors), kernel smoothing and spline smoothing have received the most attention by statisticians, and many of their statistical properties have been established under a wide range of conditions. Splines have also received much attention by mathematicians and engineers who may be familiar with cubic splines as the solution to the position of lowest energy of a rod or sheet constrained in position and attitude at its ends or edges. All the techniques model the response at a
particular location as a weighted sum of the responses in the neighborhood of the location of interest. A brief description of each of the methods follows.

**Kernel smoothing (Härdle 1990)**

The Kernel smooth expresses the estimate of the response $Y_i$ at a point $X_i$, as the weighted sum of the responses of the data points that lie within a distance $h$, called the bandwidth of the function, of $X_i$; $h$ controls the amount of smoothing. The Nadaraya-Watson form of the Kernel smooth function in the univariate case is

$$
\hat{m}(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x-X_i)Y_i
$$

where $\hat{m}(x)$ is the estimate of the response at $x$. The Kernel weight function, $K_h(u)$, can be one of a number of functions, but the differences in performance among them are small. Of much greater importance is the choice of bandwidth, $h$, which is selected by a cross-validation procedure. At the boundaries of the region the bandwidth of the kernel weight function covers fewer data points, leading to increased variance of the estimate of the response. In addition, at the boundaries the data points covered by the bandwidth are more numerous on one side of the position of the estimate, leading to possible increased bias in the estimate of the response. Usually, a correction to the kernel function is employed when estimating near the boundaries of the region to minimize these effects; see for instance Rice (1984). A major advantage of kernel smoothing is the relative computational simplicity of the technique compared to spline smoothing.

**Spline smoothing (Härdle 1990, Silverman 1985)**

Smoothing Splines are piecewise cubic polynomials that cover a region by joining smoothly at fixed points called knots. The optimal knot locations in $x$ space are at the data locations. The coefficients for the spline functions are found from the simultaneous solution of a set of equations that constrain the splines to have continuity at the knots in the first and second derivatives. Univariate smoothing splines result naturally from the solution of the following optimization problem:

$$
\min_{g(x)} S_2(g) = \sum_{i=1}^{n} (Y_i - g(X_i))^2 + \lambda \int (g''(x))^2 dx
$$

Chapter 4: Implementing Nonparametric Metamodeling . . .
where \( g(x) \) is the local spline function and \( g''(x) \) is the second derivative of \( g \) with respect to \( x \).

\( S_\lambda(g) \) has two components. The first term is a least squares fit measure while the second term is a roughness penalty. The relative importance of the two terms is controlled by the smoothing parameter \( \lambda \). The unique solution to the problem is the cubic spline. Cross-validation is used to determine \( \lambda \). Craven and Wahba (1979) developed an improved cross-validation procedure called generalized cross-validation. Although splines appear to be very different from kernel smooths, splines have been shown to have a kernel representation (Silverman 1984). The computation of spline smooths is a sophisticated numerical procedure using the theory of reproducing kernel Hilbert spaces (Wahba, 1990). We use the excellent set of Ratfor and FORTRAN codes available in Rkpack (Gu, 1989, 1992), to compute multivariate thin plate splines which are higher dimensional analogs of the smoothing spline.

**RESEARCH QUESTIONS**

The research questions for this paper arise naturally from the preceding walk through of the Classifier.

1. How should the architecture of the Classifier be altered to incorporate metamodel selection?
2. How are the following issues associated with metamodel selection, i.e., those requiring specific knowledge about the metamodeling procedures, to be resolved?
   a. How is the question of whether or not to metamodel to be decided?
   b. How is an appropriate metamodel to be chosen given the input information?
   c. What course of action is to be followed if metamodeling is not possible or is inappropriate?
3. How are the experimental design issues to be addressed? The main issues are:
   a. How are the locations and number of simulation runs to be allocated?
   b. What criteria are to be used in collecting the sample data?
   c. What should be the locations and number of the simulation runs used to augment the original sample?
   d. What are the processes required when a search terminates?
4. How are the accuracy and precision of the metamodel to be judged? Important questions are:
   a. What should the criteria be for the reduction in size of the search region?
   b. What analysis procedure is needed to determine whether the metamodel is a sufficiently accurate and precise representation of the activity in the region?
METHODOLOGY

It is recognized that particular search techniques are suited to certain problems (Jacobsen and Schruben, 1989). The selection of a search technique by the Classifier should be guided primarily by the characteristics of the problem and secondarily by the characteristics of the search technique (Zhigljavsky, 1991). To help clarify the structure of the process needed to incorporate nonparametric techniques as searches in the Classifier, a taxonomy of stochastic search procedures was developed. Figure 4.2 shows the proposed taxonomy.

The levels of the taxonomy are chosen to parallel the level of consideration of the problem, with the top levels covering strategic considerations, middle levels covering tactical considerations and the bottom levels covering operational considerations. Strategic elements of the problem are whether the problem is stochastic in nature, whether the variables are integer or continuous and whether the search is global or local in scope. Search techniques may tackle the problem at a tactical level by running simulations at design points selected by an experimental, sequential or fixed design, a probabilistic process or the next step on the path indicated by a gradient or pattern search algorithm (Jacobsen and Schruben, 1989; Barton, 1994). At the operational level, the search may be conducted using gradient information or by using evaluations of the simulation at set design points. Note that there are few search techniques listed at the bottom of the taxonomy tree because of the scarcity of techniques that have been developed for stochastic problems (Zhigljavsky, 1991). To use the taxonomy for choosing a search technique it is necessary to have the appropriate information about the problem. As a practical matter, the appropriate information is not always available and the choice may be restricted by economic factors. For example, it may not be known whether the problem requires a local or global approach and the number of simulation runs that can be made may be restricted. With these factors in mind, an overview of a Classify-Then-Search process that addresses the research questions is proposed in the next section. The details of the procedures implementation are covered in detail via an example given in a subsequent section.

A Classify-Then-Search architecture to incorporate nonparametric metamodeling

The Classifier architecture of Crouch, Greenwood and Rees is changed to accommodate nonparametric methods and the implications of the taxonomy. The proposed process is illustrated in figure 4.3. Note that here the term "classify" pertains to the classification of the problem and not the classification of the response surface produced by a metamodeling technique. A broad outline of the process is given in the following walk-through.

Chapter 4: Implementing Nonparametric Metamodeling...
Figure 4.2: The taxonomy of stochastic search techniques
Figure 4.3: Classification and search process

Chapter 4: Implementing Nonparametric Metamodeling...
Following the flow in figure 4.3 and using terms and notation from the Crouch, Greenwood and Rees Classifier,

1. The User module directs input from the user to the Select Search module. The inputs may consist of:
   a. The dimensionality of the problem
   b. The nature and number of the input variables, i.e., whether they are discrete or continuous
   c. The ranges of the variables and any constraints involving more than one variable
   d. The number of simulation runs available or, equivalently, the time available for the optimization
   e. The number of responses
   f. The nature of each response, i.e. is it multimodal or unimodal?
   g. Goal levels and tolerances for each response
   h. Optimization objectives for each response.
   i. Starting position(s) for the search

For the purposes of the examples in this paper all inputs will be considered continuous and only a single response will be considered.

2. The Default module assigns values to the inputs that are required to make a search selection. For instance, if the User does not indicate whether the response is multimodal or unimodal, then the Default module will supply a value that enables a global search technique to be selected.

3. The Select Search module takes the problem characteristics, which classify the problem, as supplied by the User and the Default module and maps the $\chi$ values to a search technique. If a previously selected search has failed or is invalid and has returned control to the Select Search module, a new search technique is selected based on the $\chi$ values returned by the failed or invalid search technique.

4. Once a search technique is selected, it is passed the $\chi$ values and the search is initialized.

5. The search performs one iteration. This may be a first order design and its analysis in the case of Response Surface Methodology or one step in a simulated annealing algorithm.

6. The $\chi$ values are updated using the most recent results from the search process.

Chapter 4 : Implementing Nonparametric Metamodeling...
7. A check is made to see if the search has reached one of its termination conditions. Crouch, Greenwood and Rees list the following termination conditions:
   a. user-specified termination conditions
   b. user-specified goals have been accomplished
   c. search has encountered the boundary of the current search area
   d. the search has stopped inside the current search area because of violations of the search technique's premises.

Conditions a and b result in the termination of the process and the provision of results to the user. Condition c results in the termination of the search concerned, the updating of $\chi$ values is carried out and control is transferred to the select search module. If it is the only search in progress then the process terminates and results are provided to the user. Condition d results in an update of the $\chi$ values and return of control to the Select Search module.

8. If there is no termination, then the information provided by the search is analyzed to see if any reduction in the region of search can be carried out. If region reduction is possible, the region is reduced and the $\chi$ values updated to reflect the new problem. Control is returned to the Select Search module. If region reduction is not possible then another iteration of the search is carried out. Control is returned to step 5.

By incorporating the information required by the taxonomy for selection of a search process into the Select Search module, all stochastic search methods, including nonparametric metamodeling procedures, can be selected. The presentation of this procedure answers research question one.

**Knowledge concerning nonparametric metamodels**

The Select Search knowledge-base will guide the choice of a search technique by incorporating in rule form theoretical and empirical results concerning the capabilities and performance of each search method. Empirical results for nonparametric regression metamodels will be obtained from experimental results on the performance of spline and kernel smooths (Keys, Rees and Greenwood, 1995) and from the experiences of experts in the field (Härdle 1990, Silverman 1985) reported in the literature. Theoretical results in the literature will be used to place bounds on the use of the techniques; for example, using nonparametric regression when the problem type is stochastic with integer variables may lead to a misleading representation of the response surface.
Experimental design for nonparametric metamodels

On the first iteration of the Classify-Then-Search process the optimal layout of design points to carry out simulation runs for nonparametric metamodels is a regular grid over the region (Müller 1984). It is proposed that the number of runs used on the grid should be the minimum number that allows a metamodel to be used, given the dimensionality of the problem. For two input dimensions, for example, if the response surface were known to be unimodal, then a second order (quadratic) polynomial could be fit using 9 runs. If the response function was thought to be multimodal, then the minimum number of runs would need to exceed the nine required for a quadratic model and be sufficient to give a uniform design for a nonparametric model. Keys, Rees and Greenwood (1995) carried out a simulation study of nonparametric metamodels and found that sixteen runs was the minimum number required. It will not be known whether the initial design has sampled the region densely enough to allow a true representation of the response surface to be metamodeled. If the sample is too small the metamodel will give an alias of the true surface. It is not possible to tell if a metamodel is an alias of the true surface from examining it. Determining whether the response surface is a true representation requires a sequential design procedure, with design points added to the initial design according to some meaningful procedure. In this paper we follow the ideas of J. J. Faraway (1990) for sequential design using kernel smooths and apply them to spline smooths in a detailed example described in a later section. Certain measures of the response surface characteristics need to be tracked during the sequential design process to detect when the response surface is giving a reasonably faithful representation of the true response surface.

The accuracy and precision of nonparametric metamodels

The convergence of a nonparametric regression to the true underlying function is guaranteed in an asymptotic sense as the number of observations used in the regression goes to infinity (Härdle, 1990). The techniques are quite robust to violations of the assumptions, (e.g. normality of error, continuity of the underlying function) used in the derivation of theoretical results for their behavior, although such violations generally lead to slower convergence to the true function. Results from the experiments by Keys et al. (1995) show that nonparametric regression requires more design points to effectively model the true function if the response is multimodal, has high variance or if the variance is heterogeneous. The construction of confidence intervals for response surfaces is problematic, given that the error distribution may be non-normal and that the variance is typically heterogeneous. Here we focus on the accuracy and precision of the characteristics that are important in optimization, that is the positions, number and relative magnitudes of the response surface optima. The accuracy and precision of the response surface metamodel
are assessed by tracking the positions and number of the optima present on the surface during the sequential design process. When user supplied tolerances on the average movement of each optimum per additional point are achieved and the number of significant optima has remained steady for a set number of iterations, the sequential design procedure stops and announce that the true surface has been found. The sequential design procedure may terminate before tolerances and stability of the optima are obtained if the run budget for the sequential design section of the optimization process is reached. The position and value of the response at the global optimum given by the metamodel is then confirmed by making simulation runs at the coordinates of the global optimum. If no other local optimum has a response within two standard deviations of the global optimum, the optimization procedure terminates and presents its results. If a local optimum has a response close enough to the global optimal response, then three repetitions are run at that location. The optimization procedure then terminates and presents results. Further details on this process are given in the next section.

IMPLEMENTING SPLINE SMOOTHING IN THE CLASSIFY-THEN-SEARCH PROCESS

Search selection

We now present the Classify-Then-Search procedure in detail when using a thin plate spline. The conditions that would lead to the selection of spline smoothing as a search technique are the following:

a. All input variables and the response are continuous
b. The scope of the search is global
c. The efficiency of design is required
d. The run budget for the simulation optimization is sufficient to support the technique

Initialization of the search

The spline smooth procedure is initialized by laying down design points uniformly on a grid that covers the region of interest. The number of runs on the grid will be the size that exceeds the number of runs needed for a quadratic polynomial. A quadratic polynomial may model a unimodal surface but more runs are required if a multimodal surface is to be modeled. In the case of two inputs, the number of points for the grid will be sixteen (Keys et al.). Simulation runs are made at each design point on the grid and the responses used to synthesize a metamodel of the response surface. The metamodel of the surface is then
analyzed to determine the positions and magnitudes of the optima on the surface. The analysis comprises a search over a fine grid to find candidate points for local optima, followed by a Hooke and Jeeves search that finds the positions and magnitudes of the local optima to any desired accuracy. The Hooke and Jeeves search has been altered to be able to find optima on constraint boundaries by terminating the search algorithm when it attempts to repeatedly leave the region of interest. The optima found are recorded.

An iteration of the search procedure

The procedure now enters the sequential design phase. We follow the ideas of Faraway (1990) for kernel smooths and apply them to the spline smooth case. Faraway uses a variable bandwidth kernel smooth to metamodel the surface and a higher order kernel function to obtain the values of second derivatives over the region of interest. Müller (1984) gives the optimal density of design points for a variable bandwidth kernel smooth as being proportional to the density of the square of the second derivative. Faraway constructs a density for the second derivative and obtains a number of quantiles, \( x_{q_i} \), equal to the number of existing design points plus one. The existing design points \( x_i \) are taken as candidate points for the quantiles of the design point density. The next design point \( x_{n+1} \) is then chosen as the value that minimizes

\[
\sum_{i=1}^{n+1} \left| x_{q_i}'' - x_0'' \right|^2
\]

(4.3)

where \( x_0 \) indicates a permutation of the design points to minimize the quantity.

In our implementation developed here, we use a result by Silverman (1984) which gives an equivalence between spline smooths and a high order variable bandwidth kernel function to substitute a spline smooth for the kernel smooths in Faraway's procedure. In constructing the density of the square of the second derivative we approximate Faraway's procedure by using the metamodel to compute the second derivative over a fine grid of 961 points and then sort the square of the values obtained. The second derivatives are calculated using the following approximation (Conte, 1965):

\[
f''(x) = \frac{f(x-d)-2f(x)+f(x+d)}{d^2},
\]

where we use \( d \) equal to approximately 0.0065. As our example is in two dimension, we obtain a Hessian from which we compute an approximate density value using:

\[
density value = f'(x_1)^2 + f''(x_2)^2.
\]
If we have $n$ data points, then $n+1$ quantiles of the density are estimated using the ranks of the values. For instance, a quantile equal to the 50th percentile is taken as the value of the 48th ranked grid point. Each quantile is associated with a grid point coordinate, so that the distribution of quantiles in $X$-space can now be compared to the present distribution of design points in $X$-space. We formulate the problem of matching the existing design points plus the next design point to the quantiles of the second derivative density so as to minimize equation 4.3 as the following assignment problem:

\[
\begin{align*}
\text{Minimize} & \quad \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} c_{ij} x_{ij} \\
\text{subject to} & \quad \sum_{i=1}^{n+1} x_{ij} = 1 \\
& \quad \sum_{j=1}^{n+1} x_{ij} = 1 \\
& \quad x_{ij} \geq 0 \\
\text{and} & \quad c_{ij} = \left| x_{qi}'' - x_{ij} \right|^2,
\end{align*}
\]

where $x_{ij} = 1$ if quantile $i$ is assigned to design point $j$, 0 otherwise.

The assignment problem is solved in our example using the Hungarian method of assignment. This method suffices for problems where the number of observations is not large. For larger problems, more sophisticated assignment algorithms can be used (Bazaraa, Jarvis and Sherali, 1990). The grid point associated with the quantile assigned to the $(n+1)$th design point gives the co-ordinates of the next design point. Note that the new design point may coincide with a previous design point because of the discretization of $X$-space by the fine grid. Faraway does not specify a solution procedure for the problem above, but states that some permutation of the design points should be used to minimize the function 4.3. The simulation model is run at the co-ordinate values and the response obtained is appended along with the design point co-ordinates to the original set of design points. A new spline smooth is synthesized from the new data set.

Chapter 4: Implementing Nonparametric Metamodeling
Updating the characteristics of the problem

The number, position and magnitudes of the optima on the spline metamodel surface are determined using the techniques described in the section on search initialization. Any values of $\chi$ that are known at this point to be in error can be updated. For instance, the starting points for search can be set to the current positions of the optima detected on the metamodel surface. The modality of the spline smooth of the surface cannot be used to update the $\chi$ value for the modality of the response until it is certain that the spline metamodel is a reasonably accurate and precise representation of the true surface. Accuracy and precision of the spline smooth are gauged using two tracking mechanisms:

1. The number of significant optima is determined from the list of optima found from the search process carried out on the spline metamodel surface. An optimum is called significant if it has a magnitude that is at least 20% of the range of magnitudes found on the spline smooth metamodel. This ensures that small optima due to noise are filtered out but optima that are due to "signal" are recorded. The user or the default procedure sets a parameter we call the window. This parameter is the number of past iterations over which tracking information is stored. If the number of significant optima is constant over the entire period of the window, then a flag is set to indicate that the number of optima on the surface is considered a reflection of the true number of optima the response possesses.

2. After each iteration, the significant optima are ranked and the Euclidean distance between optima at the same rank is recorded. When the window is filled with distances for a particular rank, the average distance over the window is calculated and compared to a user set tolerance. Tracking the distances allows stability in the positions and number of optima to be detected. The spline smooth is particularly well suited to this type of tracking because of its relatively good performance in fitting locally to response values. In the early stages of the sequential design process, the points added to the data set tend to cause the spline smooth to model noise as well as signal. This makes the number and ranks of significant optima change frequently leading to large apparent movement of optima. As the number of points increases, the effect of additional points is not to add to noise but to improve "signal," so that the average movement of optima decreases and the number of optima stabilizes.

The stopping criteria are used to terminate the sequential design process and initiate assessments of goal achievement and region reduction. The stopping criteria are that:

Chapter 4: Implementing Nonparametric Metamodelling
1. The number of significant optima has remained constant over the window.
2. Given 1 is true, the average displacements over the window of the optima fall below the tolerance set by the user.
3. Notwithstanding the status of 1 and 2, the number of simulation runs used has met the budget for the spline smooth initial and sequential design procedures.

When the spline smooth sequential procedure terminates, runs are carried out at the positions of the optima that are greater than 20% of the magnitude of the global optimum. The spline smooth is then recomputed and the relative magnitudes of the optima on the response surface are observed. An estimate of the variance of the data, calculated by the spline smooth algorithm, and the assumption that the mean responses are normally distributed are used to compute p-values for the differences between the global optimum and the other optima. P-values of less than 0.90, say, should lead to further search on the optima concerned. Whether the procedure continues or not is dependent on the number of runs left in the run budget. The availability of further runs can allow region reduction and the updating of the \( \chi \) values to reflect the new nature of the search regions and their run budgets. For example, two optima would give rise to two searches on unimodal surfaces each with its own run budget. The existing run budget could be split between the two searches or wholly applied to the search that looks most promising, perhaps because its mean response is the greatest. As nonparametric metamodeling techniques can underestimate the magnitudes of optima, especially with small sample sizes, it is probably preferable to follow the latter method of run allocation. We now present the results obtained from a practical implementation of the preceding procedures.

EXAMPLE

For ease in examining the feasibility and performance of the nonparametric techniques in the Classify-Then-Search framework we employ a simulation model which comprises a deterministic function with added error. The spline smooth computations were carried out by FORTRAN routines from Rkpack (Gu, 1992). The procedures required by the sequential design procedure and the simulation were written in Microsoft FORTRAN v5.1 and executed on a 80486 class Intel-based PC.

The procedure was tested on the following "simulation" model. The model has two inputs, \( x_1 \) and \( x_2 \), and a single response, \( y \).

\[
y_i = 363935 \left( 1 - x_{1i}^2 \right) \left( 1 - x_{2i}^2 \right) (0.99^{x_1} - 0.99^{x_2})^2 \left( \frac{d_{xy}}{100} \right) + \varepsilon_i
\]

Chapter 4: Implementing Nonparametric Metamodeling...
where \(-1 \leq x_{1l}, x_{2l} \leq 1\)

and \(\epsilon_l \sim \text{Normal}(0, (0.1 \exp(1.0))^2)\)

Heteroscedasticity of variance is emulated through the random scale factor applied to the standard deviation of the error. Each design point will then be generated using a different scale factor and hence a different variance. The magnitude of the standard deviation for the model is 10% of the maximum response, a value considered high in the field of statistics. Figure 4.4 shows the form of the simulation model response function when no error is added.

The user parameters that were set at the beginning of the optimization process are:

1. There are two real valued, continuous inputs
2. There is one real valued, continuous response
3. The region of interest is bounded by \(x_l = 1, x_i = -1, x_2 = 1, x_3 = -1\)
4. The scope of the search is global
5. The user is satisfied if the optima are not moving more than 0.05 on average between iterations

The run budget for the model is set at 38 runs.

The window size required in the stopping rule is a difficult parameter to set. Using a large value for the parameter increases the probability that the true surface will be modeled faithfully, but may be expensive in terms of using more runs than are necessary. Generally, the more complex and noisy the response surface is thought to be, the larger the window size that will be needed. For this model, a satisfactory level of confidence was felt at a window size of eight.

A uniform design of 16 design points laid out on a regular grid was used as an initial design. The sequential design process used an additional 17 runs for a total of 33 "simulation" runs. Contour plots of spline smooth metamodels are shown for the initial design, two stages in the sequential design process and the final design in figures 4.5, 4.6, 4.7 and 4.8 respectively. The data points for each metamodel are indicated on the figures. Note that most of the improvement in the representation occurred with the addition of the 25th run. Subsequent runs served to build confidence in the result obtained at the termination of the sequential design process.
Figure 4.4: Contour plot of function with no added error

Chapter 4: Implementing Nonparametric Metamodelling...
Figure 4.5: Spline smooth metamodel using initial design of 16 runs
Figure 4.6: Spline smooth metamodel using a total of 24 runs
Figure 4.7: Spline smooth metamodel using a total of 25 runs

Chapter 4: Implementing Nonparametric Metamodeling...
Figure 4.8: Spline smooth metamodel at termination of the sequential design process, 33 runs used
The optima with magnitudes greater than 20% of the global optimum at the end of the sequential design procedure are:

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Optimum</td>
<td>0.6551</td>
<td>-0.3995</td>
<td>0.9609</td>
</tr>
<tr>
<td>Other Candidate</td>
<td>-0.4363</td>
<td>0.6480</td>
<td>0.7033</td>
</tr>
</tbody>
</table>

After carrying out two runs at the positions of the two optima, the spline model gave the following co-ordinates for the two optima:

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Optimum</td>
<td>0.6738</td>
<td>-0.2773</td>
<td>0.8796</td>
</tr>
<tr>
<td>Other Candidate</td>
<td>-0.4433</td>
<td>0.6456</td>
<td>0.7163</td>
</tr>
</tbody>
</table>

The standard deviation of the data was estimated to be 0.1029 by the spline smooth routine. The p-value obtained from a one sided t-test of the hypothesis that the mean global response is greater than the other candidate’s response is 0.0571. As the standard deviation of the data is greater than the standard deviation of the surface we set the significance level at $\alpha = 0.10$. It is concluded that further search should be confined to a region around the global maximum. Design points could be placed around the estimated position of the global optimum. If more points were available, another course of action may be more advantageous. Further research needs to be done on the issue of budget-based choices of search routines. An issue of particular interest is how to balance the benefits to be obtained by staying with an initial search routine against the benefits that may be obtained by switching techniques.

Figure 4.4 should be compared to figure 4.8 to visually assess the accuracy and precision of the spline smooth metamodel. The true locations and responses of the model function without error are compared to the final results by calculating the discrepancies in response values and the Euclidean separation of metamodel optima and true optima.
The decisions made in the search process are vindicated by the comparison to the actual values. Usually, the more peaked the optimum, the more response bias nonparametric techniques tend to exhibit. The surplus runs in the example would be well spent in the vicinity of the global optimum in an attempt to obtain more precision in the x coordinates.

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y$</th>
<th>$x$ discrepancy</th>
<th>$y$ discrepancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Optimum</td>
<td>0.6045</td>
<td>-0.5733</td>
<td>1.0000</td>
<td>0.3040</td>
<td>-0.1204</td>
</tr>
<tr>
<td>Other Optimum</td>
<td>-0.5364</td>
<td>0.5843</td>
<td>0.7498</td>
<td>0.1115</td>
<td>-0.0335</td>
</tr>
</tbody>
</table>

CONCLUSIONS

From the incorporation of nonparametric metamodeling techniques into a taxonomy of stochastic search methods, several answers to the research questions can be inferred. The architecture of a classifier knowledge-based simulation optimization system has to be changed so that the metamodeling technique takes its place as a search technique in its own right. A general process flow for such an architecture is given in Figure 4.3.

The placing of nonparametric methods in a taxonomy that categorizes search techniques according to the characteristics of the simulation optimization problem allows questions surrounding the selection of the metamodel to be easily answered. Metamodeling techniques will be selected when the characteristics of the simulation optimization problem match the conditions under which the metamodeling technique can be successful. For example, RSM could be selected if the response was stochastic, the input variables were continuous, the search scope is local and a design-centered approach was affordable. If the attributes of the problem kept the same values except for the search scope taking on the value global, then nonparametric metamodels could be chosen. Factors not in the taxonomy at present such as the simulation run budget and the dimensionality of the problem may dictate whether a design, pattern or random search attribute is selected as appropriate.

Given that a nonparametric metamodel is to be implemented and that there is equal interest in each part of the search region, then a uniform design should be chosen initially (Müller, 1984). The density of coverage of the initial design should be as low as possible given the dimensionality of the problem. For an input

Chapter 4: Implementing Nonparametric Metamodeling...
dimension of two, we recommend starting with a sample size of 16. Runs can be added one at a time through a sequential design procedure that places the points so as to match the density of the second derivative squared to the density of the design points (Müller, 1984; Faraway, 1990). This approach appears to work successfully for both kernel and spline metamodels. On termination of the metamodel based search procedure there appear to be three alternatives:

1. Quit the search as the user's goals have been achieved.
2. Perform region reduction and continue the nonparametric metamodel search.
3. Perform region reduction and continue the optimization process using another search.

The choice between the alternatives is made by the knowledge-base, using the updated values of the problem characteristics.

Determining the accuracy of the metamodel's representation of the response surface is difficult. The initial uniform designs often produce aliased representations of the true surface when the modality of the true response surface is greater than one. We use a parameter called the window in conjunction with tracking the number and movement of the positions of significant optima to judge when to terminate the metamodel search process. The window ensures that the surface representation has remained stable over a number of simulation runs judged to be a reasonable trade-off between confidence in the result and expense. The value of this parameter is difficult to set because an effective value for it depends on the magnitude of the response variance and the dimensionality and modality of the response surface. A window value of eight sufficed for a two input problem with moderately high variance and a bimodal response.

Statistical tests are used to determine if the optima synthesized by the metamodel are significantly different in response value. The regions around these optima would constitute the reduced region. At present we use the variance of the raw data for the statistical tests, but this approach is unduly conservative. Using a confidence interval on the metamodel surface would be an improvement. Our implementation of the thin plate spline code did not enable the construction of metamodel surface confidence intervals.

FUTURE RESEARCH

The choice of the window parameter is an issue for further research. It would appear that its selection would be influenced by a number of factors including the difficulty of the problem and the run budget. Another issue that is influenced by the run budget is the nature of the knowledge required by the Search Select process to choose a search technique after region reduction has taken place. Is it more efficient to continue
with the metamodelling approach in the reduced region or to switch to another search technique such as a local gradient search?

The performance of the sequential design method when applied to higher dimensional problems, of the order of four or five dimensions, is of interest. Factor screening is difficult to carry out, and many simulation optimization problems have many input variables.
CHAPTER 5
CONCLUSIONS AND FUTURE WORK

CONCLUSIONS

Increasingly powerful computers, the development of application-specific simulators and the promise of parallel computing combine to make the cooperative expert system/simulation model (O'Keefe, 1986) an attractive and viable approach to simulation optimization. Search techniques that are non-invasive fit into this paradigm neatly, by allowing the simulation model to be developed separately from the search technique. The maturation of the field of nonparametric regression has provided techniques such as kernel and spline smoothing that are robust to the nature of the response and are able to fit functions of arbitrary form. This dissertation brings these developments together and answers a number of questions that arise in doing so.

The first part of this dissertation conducted an empirical investigation into the relative performance of two nonparametric regressors, kernel smooths and spline smooths, for two input, single response pseudo-simulations. The regressors are compared by evaluating their metamodeling performance using six measures. The measures are calculated for each technique over a wide range of response conditions in a full factorial experiment. A statistical analysis of the results is carried out to determine significant performance differences between the two techniques and the outcome of the analysis is expressed in tabular form. The tables allow the regions of factor space in which a technique performs well to be seen.

Spline metamodels are found to be relatively more robust than kernel metamodels over all the performance measures considered. If fit to the response is considered important, the spline generally offers better performance than the kernel metamodel. When the placement of response surface characteristics in X-space is important, then kernel metamodels are able to outperform spline metamodels. The use of kernel metamodels should be considered at low sample sizes and at high sample sizes and especially when the magnitude of the variance is low.

An important characteristic of kernel smooths is the relatively large number of local optima they synthesize compared to a spline smooth (table 3.4(d)). Under circumstances where both techniques fail to represent the response surface faithfully (the surface produced is an alias), kernel smooths are able to provide candidate points for the true optima when spline smooths are not. Thus a kernel smooth looks as if it is performing
better under measures such as L and U, when in fact its performance falls short of providing a faithful representation of the response surface.

A knowledge-based simulation optimization system could easily capture the results presented in tables 3.3 and 3.4 in the form of If-Then rules. The rules would decide whether the optimization problem required the selection of a metamodeling technique, and if so, whether to employ one or several of the techniques depending on the goals of the user.

For the two-input, single-response case, 16 runs are needed to obtain a rough representation of the response surface for a unimodal function, 36-49 runs are required for the bimodal function and 81-121 runs are required for the tetramodal function when uniform designs are employed (table 3.3(a)). If the activity level of the surface is high, then the uniform design has a high cost in terms of simulation runs. For simulation models in higher dimensions, this expenditure increases with the power of the dimension and effectively rules out the use of the techniques with a uniform design. Sequential design procedures are seen as a very important research topic.

The results of the simulation experiments were supported by inspection of kernel and spline metamodels of two response surfaces produced by inventory models of differing complexity. The characteristics expected of the two surfaces were as predicted by the parameters of the simulation optimization problem.

This research has followed a lead indicated by Barton (1992). The results obtained here indicate that nonparametric techniques are capable of successfully metamodeling a wide range of response functions. Depending on the particular characteristics of the response surface that are of interest, the spline or kernel or both techniques may be employed to synthesize a metamodel. Uniform designs are shown to be expensive for simulation optimization, and sequential design methods should be investigated. The use of nonparametric methods for problems with more inputs would seem to be restricted to dimensions no larger than four or five unless simulation runs are cheap or parallel processing is available and viable.

The second part of the dissertation looks at implementation issues raised when nonparametric metamodeling techniques are used in a classifier knowledge-based simulation optimization system. From the incorporation of nonparametric metamodeling techniques into a taxonomy of stochastic search methods, several answers to the research questions investigated in this part of the dissertation can be inferred. The architecture of a classifier knowledge-based simulation optimization system has to be changed so that the metamodeling
technique takes its place as a search technique in its own right. A general process flow for such an architecture is given in Figure 4.3.

The placing of nonparametric methods in a taxonomy that categorizes search techniques according to the characteristics of the simulation optimization problem allows questions surrounding the selection of the metamodel to be easily answered. Metamodeling techniques will be selected when the characteristics of the simulation optimization problem match the conditions under which the metamodeling technique can be successful. For example, RSM could be selected if the response was stochastic, the input variables were continuous, the search scope is local and a design-centered approach was affordable. If the attributes of the problem kept the same values except for the search scope taking on the value global, then nonparametric metamodels could be chosen. Factors not in the taxonomy at present such as the simulation run budget and the dimensionality of the problem may dictate whether a design, pattern or random search attribute is selected as appropriate.

Given that a nonparametric metamodel is to be implemented and that there is equal interest in each part of the search region, then a uniform design should be chosen initially (Müller, 1984). The density of coverage of the initial design should be as low as possible given the dimensionality of the problem. For an input dimension of two, we recommend starting with a sample size of 16. Runs can be added one at a time through a sequential design procedure that places the point so as to match the density of the second derivative squared to the density of the design points (Müller, 1984; Faraway, 1990). This approach appears to work successfully for both kernel and spline metamodels. On termination of the metamodel based search procedure there appear to be three alternatives:

1. Quit the search as the user's goals have been achieved.
2. Perform region reduction and continue the nonparametric metamodel search.
3. Perform region reduction and continue the optimization process using another search.

The choice between the alternatives is made by the knowledge-base, using the updated values of the problem characteristics.

Determining the accuracy of the metamodel's representation of the response surface is difficult. The initial uniform designs often produce aliased representations of the true surface when the modality of the true response surface is greater than one. We use a parameter called the window in conjunction with tracking the number and movement of the positions of significant optima to judge when to terminate the metamodel search process. The window ensures that the surface representation has remained stable over a number of simulation runs judged to be a reasonable trade-off between confidence in the result and expense. The value
of this parameter is difficult to set because an effective value for it depends on the magnitude of the response variance and the dimensionality and modality of the response surface. A window value of eight sufficed for a two input problem with moderately high variance and a bimodal response.

Statistical tests are used to determine if the optima synthesized by the metamodel are significantly different in response value. The regions around these optima would constitute the reduced region. At present we use the variance of the raw data for the statistical tests, but this approach is unduly conservative. Using a confidence interval on the metamodel surface would be an improvement. Our implementation of the thin plate spline code did not enable the construction of metamodel surface confidence intervals.

FUTURE RESEARCH

There is much work to be done in the simulation optimization domain. Based on this dissertation, however, the following avenues of investigation are of primary importance.

If spline and kernel smoothing are to be employed in simulation optimization, the most pressing question is how to allocate design points using other than a uniform design. A sequential design procedure, as is available for RSM, would allow a minimal initial uniform design to be augmented one or more design points at a time, leading to a more efficient metamodeling technique.

A major problem in modeling surfaces where the nature of the response is largely unknown is that of aliasing. Using too small a sample of design points generates a metamodel that does not faithfully represent the character of the true response surface. Future research needs to determine measures of validity of the metamodel. This issue also has a bearing on the specification of a stopping rule for a sequential design method.

Although the experiment carried out here was insightful, the conclusions need to be supported by further work using problems in higher dimensions, i.e., up to four or five dimensions. Such an extension could use uniform designs in the pursuit of supporting data and possibly compare selected results to a sequential design method (Faraway, 1990).

The role of nonparametric metamodels needs to be explored in the context of other searches and especially in the context of knowledge-based approaches to simulation optimization. Questions of interest would
center around whether nonparametric techniques should be used on their own or in conjunction with specific other search techniques. The influence of the simulation run budget on the use of nonparametric techniques and other searches is a particularly interesting area of further research.

Nonparametric metamodels can provide much information about the simulation response function that could be used by a machine learning mechanism within a knowledge-based simulation optimization system (KBSOS). Research is needed regarding the types of information that can be captured and used to enhance the performance of the KBSOS.

The taxonomy (figure 4.2) shows that there are other choices of a global metamodeling technique. A competitor of the techniques examined here is the backpropagation neural network, which is used for functional synthesis. It is of interest to investigate how a backpropagation neural network would perform relative to nonparametric regressors. Such an investigation could result in rules to guide the choice of metamodeling technique under various problem characteristics.

The evaluation of metamodeling techniques for local (unimodal) and global (multimodal) response surfaces is also of interest, so that the different approaches of the searches in the taxonomy can be ranked. For example, is a design-based approach always superior to a path-based approach? More development of the taxonomy is also required to place all known simulation optimization techniques in its structure and more accurately delineate the characteristics of the problem, search technique and run budget that justify a search technique's place in the taxonomy.

A more detailed evaluation of the sequential design procedure needs to be undertaken. The procedure requires a rigorous argument for its convergence properties and testing under a wide variety of response surface conditions. The issues of feasibility and performance of both nonparametric regression and the sequential design procedure need to be addressed for problems with higher dimensional inputs but with a pragmatic upper bound of five dimensions. The efficiency of these techniques relative to more widely known procedures will be an acid test of their usefulness to simulation optimization.
BIBLIOGRAPHY


Bibliography


**Bibliography**


VITA

Anthony Curtis Keys was born on June 30, 1957, in Derby, England. From 1968 to 1975 he attended first Balby High School and then Doncaster Grammar School, Doncaster. He attended Reading University, England, from 1975 to 1978, graduating with a Bachelor of Science (honors) in geological geophysics.

After obtaining a Postgraduate Certificate in Education from Southampton University during 1980-81, Mr. Keys was employed by Berkshire County Council Education Department at Furze Platt Comprehensive School, Maidenhead, from 1981 to 1985 as a teacher of physics. He was then employed at Langley Grammar School, Slough until 1988.

In 1989, Mr. Keys entered the MBA program at Shenandoah University, Winchester, Virginia, where he worked as a student loan administrator in the Financial Aid Department until he graduated in 1991. Since then he has been employed as an instructor and graduate assistant in the Department of Management Science at Virginia Tech while working on his Ph.D.

Mr. Keys is a member of the Decision Sciences Institute, the Institute for Operations Research and the Management Sciences, and the American Association for Artificial Intelligence.