USING HIERARCHICAL BAYESIAN OPTIMIZATION TO LEARN
AND EXPLOIT THE DEPENDENCY STRUCTURES OF
COMBINATORIAL MANY-OBJECTIVE DECISION PROBLEMS

A Thesis in
Industrial Engineering and Operations Research
by
Ruchit Aswin Shah

© 2010 Ruchit Aswin Shah

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2010
The thesis of Ruchit Aswin Shah was reviewed and approved* by the following:

Patrick M. Reed  
Associate Professor of Civil Engineering  
Affiliate Professor of Industrial and Manufacturing Engineering  
Thesis Co-advisor

Timothy W. Simpson  
Professor of Industrial and Manufacturing Engineering  
Professor of Mechanical Engineering  
Thesis Co-advisor

Paul Griffin  
Peter and Angela Dal Pezzo Department Head Chair  
Head of the Department of Industrial and Manufacturing Engineering

*Signatures are on file in the Graduate School.
This thesis analyzes multiobjective d-dimensional knapsack problems (MOd-KP) within a comparative analysis of three multiobjective evolutionary algorithms (MOEAs): the \( \varepsilon \)-nondominated sorted genetic algorithm II (\( \varepsilon \)-NSGAII), the strength Pareto evolutionary algorithm 2 (SPEA2) and the \( \varepsilon \)-nondominated hierarchical Bayesian optimization algorithm (\( \varepsilon \)-hBOA). This thesis contributes new insights into the challenges posed by correlated instances of the MOd-KP that better capture the decision interdependencies often present in real-world applications. A statistical performance analysis of the algorithms uses the unary \( \varepsilon \)-indicator, the hypervolume indicator, and success rate plots to demonstrate their relative effectiveness, efficiency, and reliability for the MOd-KP instances analyzed. Our results indicate that the \( \varepsilon \)-hBOA achieves superior performance relative to \( \varepsilon \)-NSGAII and SPEA2 with increasing number of objectives, number of decisions, and correlative linkages between the two. Performance of the \( \varepsilon \)-hBOA suggests that probabilistic model building evolutionary algorithms have significant promise for expanding the size and scope of challenging multiobjective problems that can be explored.

A major benefit of the \( \varepsilon \)-hBOA is that while solving problems it provides users with an interesting hypotheses on the controls and dependencies impacting their system’s performance across many-objectives, simultaneously. Efficient methods of exploiting and presenting the statistical information captured in the \( \varepsilon \)-hBOA’s Bayesian network models has significant potential to enhance the formulation and solution of environmental monitoring problems. This thesis explores the conditional dependency structure of one such environmental monitoring problem and provides interesting hypotheses on the relationships and controls impacting how alternative sampling strategies can attain near optimal tradeoffs. The thesis contributes a visualization framework to understand the value of information within
the evolving Bayesian networks and presents a significant contribution towards ambitions in the fields of machine learning, optimization, and artificial intelligence. The comprehensive analysis of the Bayesian networks provides the potential for advancing our observational hypotheses while simultaneously helping to overcome the computational barriers limiting our use of Pareto efficiency as an integrated measure of the value of information when designing environmental observation networks.
# TABLE OF CONTENTS

List of Figures vii  
List of Tables x  
Acknowledgments xii  

Chapter 1  
Introduction 1  

Chapter 2  
Multiobjective Optimization and Evolutionary Algorithms 7  
2.1 Multiobjective Optimization Problems 7  
2.2 Multiobjective Evolutionary Optimization 8  
2.2.1 The Strength Pareto Evolutionary Algorithm (SPEA2) 9  
2.2.2 The Epsilon-Nondominated Sorted Genetic Algorithm II (ε-NSGAII) 9  
2.2.3 The Epsilon-Nondominated Hierarchical Bayesian Optimization Algorithm (ε-hBOA) 11  

Chapter 3  
The Knapsack Case Study 15  
3.1 Introduction 15  
3.2 Multiobjective d-dimensional Knapsack Problem 19  
3.2.1 Definition 19  
3.2.2 Random Problem Instances 19  
3.2.3 Correlated Instances 20
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>A two objective illustration of non-domination or Pareto efficient fronts.</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Schematic diagram of the $\varepsilon$-hBOA (adapted from Kollat et al. (2008a)). The figure shows the connected runs and dynamic population sizing concepts of the $\varepsilon$-NSGAII combined with Bayesian network model building and simulation (both features of the hBOA). In the figure, $N$ represents population size and $A$ represents $\varepsilon$-non-dominance archive size.</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>Comparisons of solutions across the two repair mechanisms (Maximum Ratio Repair approach and Weighted Scalar Repair approach) on a 2 objective problem with 750 decision variables. Solutions in blue represent the results attained using the Maximum Ratio Repair approach and the red solutions were attained using the Weighted Scalar Repair approach.</td>
<td>22</td>
</tr>
<tr>
<td>3.2</td>
<td>Reference set for the uncorrelated instance of the 4 knapsack test case containing 250 items. Blue cones show the solutions contributed by $\varepsilon$-hBOA, green cones show the solutions contributed by $\varepsilon$-NSGAII and the red cones show the solutions contributed by SPEA2.</td>
<td>26</td>
</tr>
<tr>
<td>3.3</td>
<td>The network graphic depicts examples of the hierarchical Bayesian rules proposed by the $\varepsilon$-hBOA when searching for Pareto efficient sampling strategies in the correlated enumerated instance.</td>
<td>29</td>
</tr>
<tr>
<td>3.4</td>
<td>Error histograms for rules ranging in complexity from $1^{st}$ to $3^{rd}$ order of hierarchy for the uncorrelated and the correlated instances. Errors are computed as the absolute percentage deviations between the probabilities proposed by the $\varepsilon$-hBOA for the sampling rules and the actual probabilities that were attained by analyzing the Pareto efficient solution set.</td>
<td>30</td>
</tr>
</tbody>
</table>
3.5 Success plots for the 24 problem instances. The plots indicate the probability of each algorithm to cross the 85% quantile on ε-Indicator additive metrics for each problem instance. 35

4.1 True Pareto set for the 25 Well test case. COST, ERROR, and UNCERT are plotted on the X-, Y-, and Z-axes, and MASS is plotted using color 41

4.2 A sample decision tree. 43

4.3 Strong rules derived from a Bayesian network. The numbered circles represent potential decisions for sampling one of 25 wells. Green designates wells that are sampled and red represents locations that are not sampled. The interior most circles represent the dependent variables in the rules subject to the decisions made in the outer circles. If the conditions on the outer circles are met there is a probability of 98% or more that the dependent well represented by the innermost circle will be sampled. 46

4.4 Strong rules derived from a Bayesian network. The numbered circles represent potential decisions for sampling one of 25 wells. Green designates wells that are sampled and red represents locations that are not sampled. The interior most circles represent the dependent variables in the rules subject to the decisions made in the outer circles. If the conditions on the outer circle are met there is a probability of 98% or more that the dependent well represented by the innermost circle will be sampled. 48

4.5 The first-order hierarchy rules generated across 25 Bayesian networks. Green and red represents wells being sampled and not sampled respectively. Rules are to be read from from the outermost edge towards center with the interior most circles represent the dependent variables in the rules subject to the decisions made in the outer circles. Rules A,B,C are displayed in their spatial representation in the figure 4.8. 49

4.6 The second-order hierarchy rules generated across 25 Bayesian networks. Green and red represents wells being sampled and not sampled respectively. Rules are to be read from from the outermost edge towards center with the interior most circles represent the dependent variables in the rules subject to the decisions made in the outer circles. 50
4.7 The tenth-order hierarchy rules generated across 25 Bayesian networks. Green and red represents wells being sampled and not sampled respectively. Rules are to be read from from the outermost edge towards center with the interior most circles represent the dependent variables in the rules subject to the decisions made in the outer circles. ........................................ 51

4.8 The spatial representation of the rules generated by Bayesian networks of \(\varepsilon\)-hBOA. Rules A, B and C correspond to those in the Figure 4.3 to present how the rules translate into the spatial context of the problem. ........................................ 52

4.9 Verification of strong rules with respect to the Pareto set. The strong rules correspond to the rules with condition probabilities greater than 0.98. Each subplot represent deviations observed with respect to the Pareto set for each level of hierarchies. ........ 53

4.10 Verification of all the rules with respect to the Pareto set. Each rule irrespective of the conditional probabilities associated with it is verified against the Pareto set and percentage deviations are plotted. Each subplot represent deviations observed with respect to the Pareto set for each level of hierarchies. ........ 54

A.1 Reference set for the uncorrelated instance of the 4 knapsack test case containing 250 items. Blue cones show the solutions contributed by \(\varepsilon\)-hBOA and red cones show the solutions contributed by \(\varepsilon\)-NSGAII. SPEA2 did not contribute any solutions towards the generation of this reference set. ........................................ 61

A.2 Nondominated fronts generated by the \(\varepsilon\)-hBOA, the \(\varepsilon\)-NSGAII and the SPEA2 for the correlated problem instance of 2 knapsack test cases (250,500,750 items). ........................................ 62

A.3 Nondominated fronts generated by the \(\varepsilon\)-hBOA, the \(\varepsilon\)-NSGAII and the SPEA2 for the correlated problem instance of 3 knapsack test case containing 750 items. A indicates the comparison of fronts for the uncorrelated instance. A1 indicates the comparison of fronts generated by the \(\varepsilon\)-hBOA and SPEA2. A2 comprises the fronts generated by the \(\varepsilon\)-hBOA and \(\varepsilon\)-NSGAII. B indicates the comparison of fronts for the uncorrelated instance. B1 indicates the comparison of fronts generated by the \(\varepsilon\)-hBOA and SPEA2. B2 comprises the fronts generated by the \(\varepsilon\)-hBOA and \(\varepsilon\)-NSGAII. .................. 63

A.4 Success plots for the 24 problem instances. The plots indicate the probability of each algorithm to cross the 85% quantile on \(\varepsilon\)-Indicator additive metrics for each problem instance. ................ 66
LIST OF TABLES

3.1 Algorithm specific parameter settings for \(\varepsilon\)-hBOA, \(\varepsilon\)-NSGAII and SPEA2. ................................................. 24

3.2 Initial population sizes used for various test instances by \(\varepsilon\)-NSGAII and SPEA2. .................................................. 24

3.3 Percentage of Pareto optimal sets captured by the \(\varepsilon\)-hBOA, the \(\varepsilon\)-NSGAII and the SPEA2. The values in the table indicate the mean metric value across 25 algorithm runs with the standard deviation mentioned in the brackets. ............................................ 28

3.4 Percentage contributions of the \(\varepsilon\)-hBOA, the \(\varepsilon\)-NSGAII and the SPEA2 towards the generation of reference sets. .................................................. 31

3.5 Performance of the \(\varepsilon\)-hBOA, the \(\varepsilon\)-NSGAII and the SPEA2 with respect to the \(\varepsilon\)-indicator and hypervolume metrics for the uncorrelated problem instances. Highlighted values designate the statistically significant (to the 95-percent confidence level) best performing algorithm for the problem instance. ............................................. 32

3.6 Performance of the \(\varepsilon\)-hBOA, the \(\varepsilon\)-NSGAII and the SPEA2 with respect to the \(\varepsilon\)-indicator and the hypervolume metrics for the correlated problem instances. Highlighted values designate the statistically significant (to the 95-percent confidence level) best performing algorithm for the problem instance. ............................................. 33

3.7 The best performing algorithm in each of the test instance. The decision is based on the statistical analysis of the performance metrics and the run-time success dynamics of the three algorithms. .......................... 37

A.1 Percentage contributions of the \(\varepsilon\)-hBOA, the \(\varepsilon\)-NSGAII and the SPEA2 towards the generation of reference sets. Uncorrelated instance reference sets for 2 Knapsack 100 items, 250 items, 500 items and 3 Knapsack 100 items were made available by Zitzler and Thiele. ............ 60
A.2 Performance of the $\varepsilon$-hBOA, the $\varepsilon$-NSGAII and SPEA2 with respect to the $\varepsilon$-indicator and hypervolume metrics for the uncorrelated problem instances. Highlighted values designate the statistically significant (to the 95-percent confidence level) best performing algorithm for the problem instance. ................... 64

A.3 Performance of the $\varepsilon$-hBOA, the $\varepsilon$-NSGAII and the SPEA2 with respect to the $\varepsilon$-indicator and the hypervolume metrics for the correlated problem instances. Highlighted values designate the statistically significant (to the 95-percent confidence level) best performing algorithm for the problem instance. ................... 65
I would like to thank several people who have helped me make this achievement possible. First and foremost, I would like to thank my advisors Dr. Patrick Reed, and Dr. Timothy Simpson for their support and guidance throughout the past two years on this work. Dr. Patrick Reed, who challenged me to grow academically and grow beyond my perceptions of what I could achieve, has been a great mentor and instrumental in helping me succeed in my journey. My accomplishments over the past two years would not have been possible without his timely advice and suggestions. I want to acknowledge Dr. M. Jeya Chandra for helping me plan my courses for the Operations Research program. Another piece of great support to develop as a researcher in the graduate school came from Dr. Joshua Kollat and Joe Kasprzyk, I want to thank them with great sincerity. Their feedbacks and ideas on the analysis and presentation of this work have been invaluable. It has been a rewarding experience to share a cubicle with them. I would also like to acknowledge Keith Sawicz, Riddhi Singh, Christa Kelleher, Adam Ward, Tom Voltz, Mukesh Kumar, George Holmes, Susan Gregg, Alisha Fernandez, Rachel Urban, and Xuan YU for their friendship and support.

My friends and colleagues at Penn State have helped me along the way and made my stay enjoyable. A special thanks to Kalpesh Munot, Minal Lalpuria, Shivangi Nangia, and Avanti Jain, who have been close and dear friends and have made the graduate school experience a memorable one. I would like to thank them for putting up with my long odd working hours and for their patience in listening about my research. I also express sincere appreciation and love to my parents for providing me the opportunity and gracious support to pursue my dreams.
DEDICATION

To my parents.
CHAPTER 1

INTRODUCTION

Hydrologic systems can be classified as having organized complexity (Dooge, 1986; Weinberg, 1975). As noted by Dooge (1986) this intermediary classification places water resources systems between highly structured systems that can be well predicted with deterministic analytical models (e.g., planetary motion) and those that are wholly unorganized where statistical mechanics’ theory of large numbers can be employed in prediction (e.g., molecular diffusion of gases). Consequently, hydrologic systems pose a severe observation and prediction challenge where the only certainty we can scientifically support is that all of our models are wrong to varying degrees and that our ability to clarify this issue with observations is strongly constrained (Beven, 2006; Langbein, 1979; Moss, 1979b; Neuman, 2002). In combination, these issues provide a strong justification for Bayesian frameworks that acknowledge the biases, errors, and uncertainties in our data and models while simultaneously seeking to extract the useful information they may provide (Drecourt et al., 2006; Gupta et al., 2008; Kollat et al., 2008b; McLaughlin, 2002). Advancing our use of Bayesian frameworks is particularly relevant to the growing ambitions of the environmental field’s suggested research agendas (for example see Research and Education (2009)), which place a strong emphasis on observing, predicting, and managing critical “transitions” and “tipping points”. In hydrological and environmental contexts, these terms are a reference to highly nonlinear systems where critical thresholds lead to a loss of resilience and a sustained change in their observable dynamics (Folke et al., 2004; Folke, 2006).
Our detection, prediction, and management of critical environmental gradients is fundamentally dependent on our ability to design and manage observation networks. As noted by Reed et al. (2006) environmental change necessitates a shift from myopic, non-adaptive long-term observation strategies towards adaptive design frameworks that link our observation and predictions of evolving human-natural systems. Key to this challenge is properly posing and analyzing the question: what environmental observations are necessary to detect, predict, and manage the risks posed by environmental change? Although a more holistic assessment is justified, at present our national, regional, and local observation strategies are largely ad-hoc, non-adaptive, and generally disconnected from evolving water resources policy and management needs, a condition that has long been recognized (Davis et al., 1979; Langbein, 1979; Moss, 1979a).

Thirty years ago Moss (1979a) eloquently acknowledged these challenges and framed the need for future observation network design strategies to use a “...more integrated measure of information...[that] results from a complex interaction of both the hydrologic knowledge and the procedures that are used to incorporate the knowledge into decisions” (p. 1673). Moss’s recommendation represents a major departure from the more commonly employed statistical information measures (Kiefer, 1959; Shannon, 1948) by seeking to understand the value of observables for advancing knowledge while simultaneously characterizing their value to the procedures used to make decisions.

In general detecting, predicting, and managing critical environmental gradients given finite resources, requires that our observation network design frameworks strike a balance between their evolving scientific and management objectives. We propose that the Moss (1979a) call for a more integrated measure of information linking observations and decisions can be realized formally by advancing our understanding and use of the concept of Pareto efficiency to characterize the spatiotemporal evolution of water resources systems’ science and management trade-offs as well as their underlying dependencies on alternative observation decisions. Informally, Pareto efficiency refers to alternatives that cannot be exceeded in performance in all of their component objectives and are better in at least one. The mathematical partitioning rule that defines Pareto efficiency has far broader implications and potential for facilitating discoveries. This thesis explores two related
propositions when judging the value of observation systems. First, real-world budgetary constraints within observation network design problems yield resource allocation conflicts across space, time, and competing foci that are at least as complex as the multiobjective d-dimensional knapsack problem (MO-dKP). Consequently, determining the Pareto efficiency of observation networks has a NP-Complete complexity (Nondeterministic Polynomial time-complete), which means that globally optimal tradeoffs cannot be attained with modern computers for non-trivial problem instances. Second, multiobjective hierarchical Bayesian optimization provides a very promising tool for identifying observation alternatives that are approximately Pareto efficient while simultaneously providing insights into the emergent dependencies of our decisions (both science and management oriented) on critical observations.

Observation decisions in their simplest form can be viewed as n-dimensional binary vectors where a 1 means sample and a 0 designates no sampling for a candidate time and/or location. It represents a lower bound in the complexity of the problem in which a linear increase in sampling decisions yields a $2^n$ exponential growth rate of the number design alternatives. Other defensible formulations could include real-valued and/or integer decisions and would have far worse growth rates (e.g., factorial or potentially infinite). In this thesis analyzing the lower bound complexity of the environmental observation network problem demonstrates the severe computational challenge posed by seeking to characterize critical gradients. Since, environmental observation network design problem represents a severely difficult NP-complete problem class [for a more formal discussion see Cook (1971); Garey and Johnson (1979a)] that cannot be solved exactly using modern computers (i.e., deterministic Turing machines) non-trivial instances of observation network design problem can only be solved approximately and global optimality is not attainable for any algorithm. The NP-complete computational complexity of network design problems can be surmised by considering the subset of cost objectives used for the constrained allocation of sampling investments. Equation 1.1 provides a highly simplified n-objective constrained cost formulation which provides a lower bound complexity representation for any full environmental network design problem formulation.
\[
\text{Min } j_{\text{cost}} = [j_{\text{cost}}^1 X_k, j_{\text{cost}}^2 X_k, \ldots, j_{\text{cost}}^n X_k] \tag{1.1}
\]

where \( j_{\text{cost}}^i X_k = \sum_{j=1}^{m} p_{ij} x_j, \forall i \in 1, \ldots, n \)

Subject to:
\[
\sum_{j=1}^{m} p_{ij} x_j \leq c_{ik}, \forall i \in 1, \ldots, n \\
X_k \in 0, 1^n
\]

Although more complex cost equations that incorporate the nonlinearities and complexities that could be associated with a more economic-oriented formulation that accounts for the time value of investment would be defensible, Equation 1.1 provides arguably the simplest meaningful accounting for cost as a simple linear summation of discrete costs for the \( j^{th} \) sample of the environmental \( i^{th} \) state (e.g., concentrations of contaminants, water levels, soil moisture, rainfall, etc.). Taken as a whole, the objectives and constraints of Equation 1.1 represents a special instance of a multiobjective d-dimensional knapsack problem [MOd-KP termed the subset sum formulation where capacity constraints' weights equal items' respective profit coefficients, see (Martello and Toth, 1990; Pisinger, 2005)]. Equation 1.1 moves beyond highlighting that environmental observation network design has similarities to the knapsack problem (Knopman and Voss, 1989; Knopman et al., 1991) and shows that the multiplicity of hydrologic and water quality states of interest in combination with fiscal limits on our observation investments makes the problem at least as complex as a knapsack problem. Consequently, given that the knapsack problem is a classic NP-complete problem, it implies that environmental observation design is an NP-complete problem class. Moreover, alternative instances of the knapsack problem can be vastly more difficult than others when seeking high quality approximate solutions. So the immediate concern for environmental observation network design is answering the question, how hard is our instance of the knapsack?

The historical theoretical work for the knapsack problem provides insights into the difficulty of the observation networks problem class. Prior studies (Martello and Toth, 1990; Pisinger, 2005) have clearly shown that a high degree of correla-
tion or interdependence between the knapsack problem’s binary decisions and/or constraints often dramatically increases the difficulty of finding high quality approximate solutions. These findings represent a severe concern for environmental network design because observation decisions across space-and-time are fundamentally linked and interdependent due hydrologic systems’ socio-physical organization.

Mathematically the concept of hierarchy provides a useful means of capturing how a particular decision to observe at the current location and time influences the impacts of observations at other times and locations. In simple terms, hierarchy may be viewed as a series of probabilistic if-then-else observation rules across space-and-time. It remains an important challenge to discover and exploit these dependencies. Recently, Pelikan and Goldberg (2003) introduced a new form of evolutionary optimization tool termed the Hierarchical Bayesian Optimization Algorithm (hBOA) to provide the capability to learn and exploit Bayesian network models of decision interdependencies while solving problems. Pelikan (2002) demonstrates that the hBOA can attain sub-quadratic computational scaling for severely challenging hierarchically structured single objective problems. In simple terms, this means that the hBOA has reduced computational demands when solving increasingly larger problems with hierarchical dependencies. Moreover, Pelikan and Goldberg (2003) show that more traditional single objective solution tools can have exponentially scaled computational complexities for hierarchically structured problems (i.e., they rapidly fail to attain high quality results unless they resort to enumeration).

In its original inception, the hBOA is a single objective probabilistic model building evolutionary algorithm (Pelikan, 2002; Pelikan and Goldberg, 2003). Kollat et al. (2008a) introduced the Epsilon Dominance Hierarchical Bayesian Optimization Algorithm (ε-hBOA). The ε-hBOA represents a new type of multiobjective evolutionary algorithm. During evolution, the ε-hBOA selects high performing solutions and builds Bayesian network models of the underlying probabilistic dependencies between their decisions. After learning these dependencies, ε-hBOA uses them within its Bayesian network models to generate probabilistic hypotheses on what decision combinations would yield improved candidate solutions. An important contribution of the ε-hBOA is that while solving many-objective prob-
lems, the algorithm is explicitly building an approximate joint probabilistic density function (pdf) model of what makes decisions likely to be nondominated with respect to many-objectives, simultaneously.

This thesis builds on the initial contributions of \( \varepsilon \)-hBOA to explicitly test how the algorithm’s Bayesian model building capabilities improve its scalability when solving challenging instances of the MOd-KP. Moreover this thesis also demonstrates the quality and value of the hierarchical Bayesian network models for discovering decision dependencies that strongly impact solutions’ performance across d-objectives. In this thesis, Chapter 2 provides a general definition of multiobjective optimization and describes the baseline algorithms used to demonstrate the effectiveness of the \( \varepsilon \)-hBOA. Chapter 3 provides a detailed introduction to the MOd-KP problem class and the computational experiment used in this thesis to demonstrate the performance of \( \varepsilon \)-hBOA. The thesis then moves from the theoretical study of the knapsack problems to a real-world many-objective environmental monitoring design application. Chapter 4 introduces the long-term groundwater monitoring (LTM) problem as an instance of the MOd-KP problem class. This chapter provides a detailed verification of \( \varepsilon \)-hBOA’s ability to discover and exploit the hierarchical dependency structure of a real-world problem. Lastly, Chapter 5 concludes with the key findings from the thesis and recommendations for future work.
CHAPTER 2

MULTIOBJECTIVE OPTIMIZATION AND EVOLUTIONARY ALGORITHMS

2.1 Multiobjective Optimization Problems

A general multiobjective optimization problem can be described as the problem of optimizing multiple conflicting objectives simultaneously wherein gain in performance of one objective is at the expense of performance in another objective. Formally, it can be defined as:

\[
\begin{align*}
\text{Max/Min } Z &= f(x) = \{z_1 = f_1(x), z_2 = f_2(x), \ldots, z_k = f_k(x)\} \\
\text{Subject to } x &= (x_1, x_2, \ldots, x_n) \in X
\end{align*}
\]

(2.1)

where \(x\) is a vector of decision variables, \(X\) is the set of feasible solutions in the decision space and \(Z\) is the corresponding objective space. A solution \(x\) when mapped into the objective space is a point \(z^x = f(x) = [z_1^x, z_2^x, \ldots, z_k^x]\), where \(z_j^x = f_j(x)\) for all \(j=1,2,\ldots,k\).

The solution set for a multiobjective optimization problem is a set of nondominated decision vectors called an efficient vector or Pareto optimal vector. In other words, a solution \(x\) is an efficient solution if there is no other solution that dominates \(x\). The concept of dominance can be explained as follows: Assuming, without loss of generality, a maximization problem; then point \(z^1\) dominates \(z^2\), \(z^1 \succ z^2\), if
\[ \forall j \ z_j^1 \geq z_j^2 \] and there exists at least one \( j \) such that \( z_j^1 > z_j^2 \) (Deb (2001)). Solution \( x_1 \) dominates \( x_2 \) if the image of \( x_1 \) dominates the image of \( x_2 \).

Figure 2.1 provides a two-objective illustration of the concepts of non-dominance and Pareto efficient fronts. In the figure, assuming minimization of both the objectives, the goal is to attain the minimum level error for each objective. The shaded boxes designate the objective space dominated by Solutions 1, 2, and 3 in the figure. The full set of nondominated solutions as plotted represent the Pareto efficient frontier or optimal tradeoff between Objective 1 versus Objective 2.

2.2 Multiobjective Evolutionary Optimization

This thesis focuses on MOEAs that exploit population based search to evolve entire Pareto-optimal tradeoffs or generate Pareto-approximate tradeoffs for challenging problems. For a general introduction to MOEAs, the reader is encouraged to refer to the excellent texts by Deb (2001) and Coello et al. (2002). The following sections give a brief summary of the evolutionary algorithms used in this thesis.
2.2.1 The Strength Pareto Evolutionary Algorithm (SPEA2)

The SPEA2 introduced by Zitzler et al. (2002) was selected for comparison as it was the benchmark MOEA that has outperformed other MOEAs on a suite of MOd-KP problems that have been explored in prior literature (Zitzler and Thiele, 1999; Zitzler et al., 2001). SPEA2 is an elitist MOEA with a fitness assignment strategy that incorporates density estimation. It generates an initial random population and calculates the corresponding raw fitness taking into account the number of solutions each population member dominates and the number of solutions that dominate it. To discriminate between individuals having identical raw fitness values, the density approximation technique adapted from the $k^{th}$ nearest neighborhood method is used. Final fitness of an individual is calculated by adding up the density and raw fitness. Subsequently, environmental selection copies the nondominated individuals from the archive and the population to the archive of the next generation. If the nondominated solutions fit exactly into the archive, then the selection process is complete. If the number of nondominated solutions are fewer than the size of the archive, then along with the nondominated solutions the best dominated solutions are used to fill up the archive. If the number of nondominated solutions exceeds the size of the archive, then an archive truncation procedure is invoked which removes the solutions with the minimum distance to the $k^{th}$ nearest neighbor. Termination of the SPEA2 is based on a user-specified maximum runtime in terms of generations. The reader is encouraged to refer to literature published by Zitzler and Thiele (1999) and Zitzler et al. (2002) for a detailed description of the algorithm.

2.2.2 The Epsilon-Nondominated Sorted Genetic Algorithm II ($\varepsilon$-NSGAII)

The $\varepsilon$-NSGAII is based on the NSGAII (Deb et al., 2002), an elitist MOEA. NSGAII uses a non-domination sorting approach to classify solutions according to the level of non-domination and a crowding distance operator to maintain solution diversity across approximation solution sets. The $\varepsilon$-NSGAII developed by Kollat and Reed (2005, 2006) reduces the extensive parameter calibration by using the con-
cepts of $\varepsilon$-dominance archiving (Laumanns et al., 2002; Deb et al., 2003), adaptive population sizing (Harik and Lobo, 1999) and self-termination. The $\varepsilon$-NSGAII has been validated extensively across a suite of test problems and applications (Kollat and Reed, 2005) and has been shown to perform as well or better than other state of art MOEAs (Kollat and Reed, 2006; Tang et al., 2006).

The algorithm generates an initial small random population and uses non-domination and crowding distance to assign fitness to each individual. A non-domination sort is performed across all the solutions and individuals are classified into fronts based on their ranks, with rank 1 assigned to the solutions that are nondominated. Additionally, crowding distance is calculated for all individuals based on the average Euclidean distance between an individual and the individuals within the population which are assigned the same rank. Selection is done using binary tournaments and is based on the rankings and the crowding distances of the individuals with a preference given to larger crowding distance. Individuals with larger crowding distance add to the diversity of the population and helps to ensure that the $\varepsilon$-NSGAII explores the entire trade off landscape. Selected individuals now become parents of the next generation and the evolution process is repeated. These individuals are also eligible to enter an offline archive that stores the best solutions throughout the run. To achieve entry into the archive, individuals should be $\varepsilon$-nondominated with respect to solutions in the archive.

The $\varepsilon$-NSGAII then uses a series of “connected runs” to inject the archive solutions into the population of the next run using a 25% injection scheme. The injection scheme requires that the present archive forms 25% of the next population and the remaining 75% is filled with randomly generated individuals. This assists the performance of $\varepsilon$-NSGAII by directing the search towards previously known good solutions. However the 75% random solutions helps to ensure that the algorithm does not preconverge and encourages the exploration of new regions. The algorithm can increase or decrease its population size as the search progresses and adapts its population based on the solutions obtained.

The $\varepsilon$-dominance archive allows the user to control the computational costs of evolution by specifying their precision requirements for each of the objectives. Based on the user’s preferences, the algorithm applies a grid to the search space of the problem that can significantly reduce its computational costs when solving
multiobjective problems by avoiding unnecessary precision in calculations (Kollat and Reed, 2007a). Larger $\varepsilon$ values result in a coarser grid (and ultimately fewer tradeoff solutions) while smaller $\varepsilon$ values produce a finer grid. The fitness of each solution is then mapped to a box fitness based on the specified $\varepsilon$ values. Non-domination sorting is then conducted using each solution’s box fitness, and solutions with identical box fitness (i.e., solutions that occur in the same grid block) are compared and those that are dominated within the grid block are eliminated. Only a single nondominated solution is permitted in any one grid block, preventing clustering of solutions and promoting a more diverse search of the objective space. For further details one can refer to prior work by Laumanns et al. (2002) and Deb et al. (2003). Dynamic population sizing allows $\varepsilon$-NSGAII to start with a small initial population to pre-condition the search at a low computational cost in terms of the number of function evaluations. When the size of the $\varepsilon$-dominated archive stabilizes, the connected runs are equivalent to a diversity based EA search enhancement recommended by Goldberg (2002) termed time continuation, where diverse search is sustained as long as it is required or feasible. Prior work using the $\varepsilon$-NSGAII by Kollat and Reed (2005, 2006) can be referenced for more details on the algorithm and its dynamic search features.

### 2.2.3 The Epsilon-Nondominated Hierarchical Bayesian Optimization Algorithm ($\varepsilon$-hBOA)

The class of algorithms termed probabilistic model building genetic algorithms (PMBGA) replace the crossover and mutation operators of traditional MOEAs with statistical models of selected promising solutions. New solutions are then generated according to the PMBGA’s probabilistic models of how decision variables should be combined to yield high quality solutions. The models estimate the interdependencies through the linkages between decision variables, which are then refined throughout the evolutionary process. In effect, the statistical models seek to exploit the structure of the problems being solved and use this information for proper mixing and exploration of their search spaces. Prior PMBGA literature has shown that model building algorithms are generally less effective than traditional EAs for easy problems that are linearly separable (i.e., independent
Figure 2.2. Schematic diagram of the \( \varepsilon \)-hBOA (adapted from Kollat et al. (2008a)). The figure shows the connected runs and dynamic population sizing concepts of the \( \varepsilon \)-NSGAII combined with Bayesian network model building and simulation (both features of the hBOA). In the figure, \( N \) represents population size and \( A \) represents \( \varepsilon \)-non-dominance archive size.

Decisions) and increasingly superior as problem difficulty increases (i.e., increasing interdependencies between decisions as shown by Pelikan (2005)).

The \( \varepsilon \)-hBOA was built on the hierarchical Bayesian optimization algorithm (hBOA) developed by Pelikan (2002). The hBOA uses a Bayesian network to model promising solutions and guide search space exploration. The hBOA has the unique characteristic of being able to solve hierarchically difficult problems (Pelikan, 2002; Pelikan and Goldberg, 2003; Pelikan et al., 2006). Mathematically, hierarchy is defined by the ability to break down a system into subsystems, each of which in turn represents a hierarchy themselves, until some decomposable bottom level is reached. It does so by a technique referred to as chunking within the Bayesian network, which allows a group of decision variables that are at a lower hierarchical level to be clustered, and used subsequently to model higher-order interactions. Decision graphs and trees are used to represent the conditional probabilities of the
model. The hBOA has been shown to perform favorably in many test functions and real-world applications (Pelikan, 2002).

Kollat et al. (2008a) developed the ε-hBOA as a multiobjective hierarchical Bayesian optimization algorithm with dynamic population sizing and ε-nondominated archiving options. Figure 2.2 provides an overview of the algorithm, which blends the model building of hBOA and connected runs of ε-NSGAII. The algorithm initially generates a population of \( N \) random individuals and uses non-domination and crowding distance to assign the fitness to each individual. Crowded tournament selection is performed to select promising individuals based on their fitness values. In the crowded tournament selection operator, preference is given to the solutions that possess the best nondominated rank. If two solutions have the same non-domination rank, then diversity is rewarded by selecting solutions based on their crowding distance. Solutions with the highest crowding distance are selected to spread solutions along the full extent of the nondominated set since they are less redundant with their neighbors. In ε-hBOA, Bayesian network models are generated to represent decision variable combinations that are expected to yield improved crowded rank fitness values. In essence, in each generation ε-hBOA is learning an approximate probabilistic model for what decision variable combinations yield diverse solutions with superior nondomination ranks. The Bayesian networks are directed acyclic graphs that model the conditional probabilities between the decision variables in a problem. An edge connecting any two nodes denote a dependency structure between the two nodes.

The Bayesian network in Figure 2.2 illustrates an example of dependency structure between the decision variables. In the example, selecting decision variable 7 is dependent on decision variables 1 and 3 (The node being pointed to depends on the node from which the arrow originates). A similar dependency exists with respect to decision variables 2, 8 and 5. The Bayesian network is iteratively generated so as to maximize the Bayesian network’s ability to produce high quality solutions. In each model building iteration, a greedy local search algorithm performs elementary network operations (such as edge additions and removals) to maximize the improvement in the quality of the model. The quality of the Bayesian network model is monitored through the Bayesian information criterion metric. Complex models that do not significantly increase the predictive behavior of the model are
penalized. Model building is terminated when no significant improvements in predictive behavior are observed. The generated model forms the basis of the next generation. Decision graphs are used to store the conditional probabilities of each decision variable and child solutions are sampled from the model’s joint probability distribution. The algorithm then proceeds as \( \varepsilon \)-NSGAI where Pareto ranking and crowded binary tournament selection is used to fill the new population. Selected children now become the parents for the next generation. Dynamic population sizing of \( \varepsilon \)-NSGAI is utilized and the new population is composed of the archive from the previous generation.

It should be noted that our work deviates from the original \( \varepsilon \)-hBOA developed by Kollat et al. (2008a) in number of ways. In their base dynamic population sizing configuration, they used an \( \varepsilon \)-nondominated archive injection rate of 25%. Thus the new population of every connected run was generated so that 25% of it was composed of the previous archive and 75% of the solutions were randomly generated. When implemented in \( \varepsilon \)-NSGAI, it provides the algorithm with new genetic material which is perturbed by the mating and mutation operators to explore the solution space. However, in the case of the \( \varepsilon \)-hBOA, the randomly generated solutions can adversely impact new models generated by the algorithm’s Bayesian network. The random data destroys the statistical information generated from the previous runs and forces the model to re-evaluate the dependencies and its estimates of their corresponding probabilities. This model building disruption can lead to a substantial number of function evaluations being wasted. Alternatively, we have removed the random injection and are using a 100% injection strategy (i.e., we use the full archive) in the successive runs of the \( \varepsilon \)-hBOA. If the archive size of the previous run is less than the specified lower bound, then the remaining solutions are sampled from the last available model of the previous run.
3.1 Introduction

The “Knapsack Problem” (KP) introduced by Dantzig (1957) has been recognized as “One of the great moments of history of Operations Research” by Gass (2002). The KP has been instrumental to advancements in computational complexity, approximation schemes and reduction algorithms (Pisinger, 1995). As a classical combinatorial optimization problem, the KP has attracted wide attention from researchers and practitioners. Decades of research have shown that even the simplest of the knapsack problems cannot be solved in polynomial time (Martello and Toth, 1990; Pisinger, 1995; Garey and Johnson, 1979b), motivating ongoing research on determining near-optimal or approximate solutions in polynomial time. There is growing interest on moving beyond the traditional single objective formulations of the knapsack problem towards variants such as multiple choice knapsack problems and multiobjective knapsack problems.

This thesis focusses on the multiobjective d-dimensional knapsack problem (MOd-KP) in which the decision-maker (DM) wants to optimize two or more objectives simultaneously as well as satisfy the capacity constraints associated with the problem. To illustrate, consider the following example from Erlebach et al. (2001): “A government agency has to choose a subset out of a given list of different projects subject to monetary restrictions. Each project requires a certain budget (possibly also human resources of the agency, office space, etc.) and yields
a certain profit for different objectives such as employment effect, infrastructure, side effects for private economy, social effects and public opinion.” The DM would like to have a suite of alternatives that explores the tradeoffs between the objectives so that he/she can select the best solution as per his/her priorities. Since knapsack problems and their multiobjective variants cannot be solved in polynomial time, recent research efforts have focused on developing approximation schemes to obtain near-optimal solutions. Multiobjective 1-dimensional knapsack problems have been addressed in prior literature and fully polynomial time approximation schemes (FPTAS) have been developed by Erlebach et al. (2001) and Bazgan et al. (2007). Polynomial time approximation schemes (PTAS) have been developed by Erlebach et al. (2001) for the MOd-KP, however the approximation methods have limitations with respect to the number of objectives and number of decision variables that can be explored in different instances of the problem (Ehrgott and Gandibleux, 2000, 2004).

Limitations in approximation methods for solving the MOd-KP have motivated researchers to approach these problems with evolutionary algorithms (EAs). EAs evolve solutions through a process analogous to Darwinian selection (Goldberg, 1989) with search operators that mimic selection, mating and mutation. Over the past few decades, evolutionary algorithms have been extensively used to address a broad range of single and multiobjective problems. Multiobjective evolutionary algorithms (MOEAs) have been shown capable of approximating solution sets that compose the tradeoffs for highly nonlinear, discrete and non-convex objective space landscapes (Back et al., 2000; Deb, 2001; Coello et al., 2002).

Zitzler and Thiele (1999) were among the first authors to introduce MOd-KP problem instances into the EA literature. They performed a comparative study of five different MOEAs across a suite of test instances. Knowles and Corne (2000b) continued the above mentioned experiment using the same set of test instances and explored the use of a multiobjective memetic algorithm that hybridizes local search with recombination operators. The authors indicated that the success of the memetic algorithms can be attributed to convexity in the search space and subsequently tested other memetic algorithms on the same problem instances (Knowles and Corne, 2000a). Zitzler et al. (2002) introduced the Strength Pareto Evolutionary Algorithm (SPEA2) and demonstrated that it provided higher quality
search results for the MOd-KP problem instances relative to other algorithms of that time. Jaszkiewicz (2002) compared early MOEAs to multiobjective genetic local search (MOGLS) yielding results that indicated the superior performance of the local search method. Vianna and Arroyo (2004) proposed a greedy adaptive search procedure (GRASP) which uses a linear utility function with local search to approximate the efficient or Pareto optimal set. Their results indicate that GRASP could outperform the MOGLS and SPEA2 on some instances of the MOd-KP. However it should be noted that the MOEAs were implemented with the constraint handling procedure suggested by Zitzler and Thiele whereas the MOGLS and GRASP utilized a superior constraint handling procedure introduced by Jaszkiewicz (2002). The constraint handling procedures have a very significant impact on the evolutionary dynamics of the MOEAs. If the repair procedure is modified then it is a wholly new problem formulation. Thus the studies performed by Jaszkiewicz (2002) and Vianna and Arroyo (2004) do not reflect an unbiased comparative analysis of MOGLS and GRASP relative to SPEA2. A detailed description of the two repair strategies is provided in Section 3.4. Different instances of multiobjective knapsack problems have also been approached through multiobjective simulated annealing (MOSA), Pareto simulated annealing (PSA) and Tabu search. Ehrgott and Gandibleux (2000, 2004) provide a more detailed survey of the different approximation methods that have been applied to multiobjective combinatorial optimization problems for interested readers.

This thesis extends the previous comparisons and tests two other algorithms, the $\varepsilon$-nondominated sorted genetic algorithm ($\varepsilon$-NSGAII) from Kollat and Reed (2005, 2006) and the $\varepsilon$-nondominated hierarchical Bayesian optimization algorithm ($\varepsilon$-hBOA) (Kollat et al., 2008a). SPEA2 is used as a baseline algorithm in this thesis to connect to the historical work in this area and the benchmarks provided by Zitzler et al. (2002). This thesis also clarifies how the use of different constraint handling techniques in historical MOEA comparative studies for the MOd-KP (Jaszkiewicz, 2002; Vianna and Arroyo, 2004) biases algorithm comparisons where performance differences cannot be attributed to the algorithms themselves. In this work, we avoid this bias by using consistent formulations and implementations of the MOd-KP for the SPEA2, $\varepsilon$-NSGAII, and $\varepsilon$-hBOA. The $\varepsilon$-NSGAII was built on NSGAII (Deb et al., 2002) through the inclusion of dynamic population sizing.
and \( \varepsilon \)-dominance archiving. It shares the similar concepts of elitism and archiving as SPEA2 and uses the same operators for selection, mating, and mutation. Our comparison highlights the differences between these algorithms with respect to their fitness assignment strategies and the value of \( \varepsilon \)-dominance archiving.

The MOd-KP is a combinatorial optimization problem that requires decision-makers to select a set of items to be added to all of the knapsacks. Traditional MOEA operators like crossover and mutation assume that the problem’s decision variables are statistically independent. It is worthwhile to consider interdependencies in MOd-KP instances given their likelihood of existing in real-world applications. Moreover, these problem instances have the potential to contribute valuable insights on the relationship between MOd-KP problem structures and their concomitant influence on problem difficulty. To explore this characteristic, we have introduced an algorithm in the comparison which can model interdependencies between decision variables with the help of probabilistic Bayesian networks. The \( \varepsilon \)-hBOA built on hBOA (Pelikan, 2002) replaces the variation operators of an evolutionary algorithm with a probabilistic Bayesian network that accounts for the interdependencies in decision variables when generating new alternative solutions. This thesis demonstrates the performance of the three algorithms mentioned above across the suite of MOd-KP problem instances introduced by Zitzler and Thiele (1999) and introduces more difficult correlated instances of the problem. This thesis seeks to determine if the probabilistic model building capability of \( \varepsilon \)-hBOA advances the ability of MOEAs to provide high quality approximations for the MOd-KP for growing objective counts, increasing numbers of decisions, and interdependencies between these two problem properties.

This thesis demonstrates the performance of these three algorithms across the suite of MOd-KP introduced by Zitzler and Thiele and introduces difficult correlated instances of the problem. This thesis aims to determine if the probabilistic model building capability of \( \varepsilon \)-hBOA advances the ability of MOEAs to provide high quality approximations for the MOd-KP for growing objective counts, increasing numbers of decisions, and significant interdependencies between these two problem properties. This thesis explores the effectiveness \( \varepsilon \)-hBOA in solving multiobjective combinatorial optimization problems and tests it against the best performing algorithms available in the literature. This thesis demonstrates the
performance of the three algorithms ($\varepsilon$-hBOA, $\varepsilon$-NSGAII and SPEA2) across the suite of MOd-KP introduced by Zitzler and Thiele and introduces difficult correlated instances of the problem. Superior performance of $\varepsilon$-hBOA would justify the presence of an inter-related decisions within the MOd-KP instances. It would contribute valuable insights on the relationship between MOd-KP problem structures and their concomitant influence on problem difficulty.

3.2 Multiobjective d-dimensional Knapsack Problem

3.2.1 Definition

Formally the multiobjective d-dimensional knapsack problem (MOd-KP) is defined as:

Given a set of $n$ items and a set of $d$ knapsacks:

Maximize : \( \{ f_1(x), f_2(x), ... f_d(x) \} \), where, \( f_i(x) = \sum_{j=1}^{n} p_{ij} x_j \)

Subject to : \( \sum_{j=1}^{n} w_{ij} x_j \leq c_i, \forall i \in \{1, 2, ..., d\} \)

\( x = (x_1, x_2, ..., x_n) \in \{0, 1\}^n \) \hspace{1cm} (3.1)

where, \( p_{ij} = \) profit of item \( j \) corresponding to knapsack \( i \), \( w_{ij} = \) weight of item \( j \) corresponding to knapsack \( i \) and \( c_i = \) capacity of knapsack \( i \). The problem is also simply referred to as a multiobjective knapsack problem or a multiobjective multi-constraint knapsack problem. This thesis follows the definitions and terminologies suggested by Pisinger (1995).

3.2.2 Random Problem Instances

In this thesis, we explore problem instances and test data drawn from Zitzler and Thiele (1999). The instances have two, three, and four objectives with 100, 250, 500 and 750 decision variables. These instances are generated randomly to have
3.2.3 Correlated Instances

Zitzler and Thiele followed the suggestion by Martello and Toth (1990) to generate uncorrelated profits and weights. Martello and Toth have shown that increasing the correlations between the profits and weights increases the expected difficulty of the problem. Various experiments have been conducted by researchers (e.g., see (Martello et al., 2000; Pisinger, 2005)) by introducing instances of correlations between profits and weights in the single criteria KP problem. Their studies have indicated that there is a significant increase in the computational challenges posed by knapsack problems with correlation (between profits and weights) and that various exact/approximate algorithms often fail to find high quality solutions within feasible computational times.

Literature introducing correlations into MOd-KP instances has been limited. Captivo et al. (2003) and Bazgan et al. (2009) have conducted experiments by introducing correlations in bicriteria one dimensional knapsack problems. They have shown that correlated instances are in general harder to solve as compared to uncorrelated instances. Based on their experiments, we have introduced correlations between profits and weights. For each knapsack $i$, the weights ($w_j$) are generated randomly between 10 and 100 (i.e., $w_j \in [10, 100]$) and profits $p_j \in [w_j - 10, w_j + 10]$.

This level of correlation between the profits and weights is a realistic assumption which may hold true in many of the practical applications. Referring back to the example introduced in Section 3.1, it is reasonable to assume that the budget requirement may govern the profits earned by the project and the human resources requirement may influence the employment status of an agency.

3.2.4 Repair Strategies

As has been highlighted in prior work (Zitzler and Thiele, 1999; Zitzler et al., 2002; Jaszkiewicz, 2002; Vianna and Arroyo, 2004), infeasible solutions are repaired until all of the constraints are satisfied. The repair heuristics play a vital role in the formulation and implementation of MOEAs when solving MOd-KP instances given

uncorrelated profits and weights. The capacities are equal to half the total weight of the items.
the challenges posed by capacity constraints.

Zitzler and Thiele (1999) used a repair heuristic which is a simple extension of the approach used in the single objective knapsack problem (Martello and Toth, 1990) to the multiobjective domain. The order of removal is determined by the maximum profit by weight ratio per item across all knapsacks. For item $j$, the maximum profit by weight ratio is given by:

$$q_j = \max(p_{ij}/w_{ij}), \ \forall \ i = 1, 2, 3, \ldots d \quad (3.2)$$

The items are removed in an increasing order of $q_j$ (i.e., the item with the lowest maximum profit by weight ratio being removed first). This method (maximum ratio repair) ensures that it reduces weights with the lowest possible loss in the profit values.

Jaszkiewicz (2002) introduced a repair heuristic where weighted profit by weight ratio (unlike the previous approach which uses the maximum profit by weight ratio) is used to determine the order of removal of items. It uses the scalar coefficients of the linear utility function from MOGLS to calculate the weighted sum of profits. Thus the items are sorted based on the following ratio:

$$q_j = \left( \sum_{j=1}^{J} \lambda_j p_{ij} \right) / \left( \sum_{j=1}^{J} w_{ij} \right), \ \forall \ i = 1, 2, 3, \ldots d \quad (3.3)$$

where $\lambda_j$'s are elements if the weight vector used in the current iteration. The items are removed in an increasing order of $q_j$ (i.e., the item with the lowest weighted profit by weight ratio being removed first). This method (weighted scalar repair) removes items that locally decrease the weights in the knapsacks at the lowest cost of the current scalarizing function.

In some prior algorithm comparisons for the MOd-KP (Jaszkiewicz, 2002; Vianna and Arroyo, 2004), the comparisons allowed the authors’ algorithms to have different constraint handling techniques relative to SPEA/SPEA2, and thus their results do not reflect the true relative performances of the algorithms. Ishibuchi et al. (2005) have shown that the weighted scalar repair approach produces a more diverse set of solutions as compared to maximum ratio repair approach. To further illustrate this point, we have implemented both repair mechanisms across the
Figure 3.1. Comparisons of solutions across the two repair mechanisms (Maximum Ratio Repair approach and Weighted Scalar Repair approach) on a 2 objective problem with 750 decision variables. Solutions in blue represent the results attained using the Maximum Ratio Repair approach and the red solutions were attained using the Weighted Scalar Repair approach.

three algorithms tested in this thesis. Figure 3.1 shows the modified variant of weighted scalar repair implemented on the ε-hBOA, ε-NSGAII, and SPEA2 for the 750 item MOd-KP problem instance. The solutions in blue correspond to the maximum ratio repair approach and the solutions in red correspond to the weighted scalar repair approach. The nondominated fronts were generated across 25 algorithm runs with one million function evaluation in each run. As illustrated in the figure, for each algorithm (SPEA2, ε-hBOA and ε-NSGAII) the constraint handling technique plays an important role in the diversity of the nondominated
front and eventual performance of the algorithm. The \textit{weighted scalar approach} produces a far more diverse front as compared to the \textit{maximum ratio approach} for all three algorithms. Similar trends were observed across all of the other MOd-KP problem instances examined in this thesis. In the studies of Jaszkiewicz (2002) and Vianna and Arroyo (2004) their MOGLS and GRASP algorithms were implemented to use \textit{weighted scalar repair}. However, it should be noted that in these studies SPEA/SPEA2 were implemented with \textit{maximum ratio approach}, which represents a severe bias in these prior results.

To ensure a fair comparison, it is essential that each of the algorithms should have exactly the same implementation of the problem so that the only distinguishing factor in the results can be attributed to the performance of the algorithm. Thus in this thesis, we have opted to do an extensive comparative analysis of performance using both repair schemes in all tested problem instances. This choice links our results to the historical body of work and enhances the value of our comparative analysis. Initially the three algorithms were implemented using the maximum ratio repair approach and the reference sets and performance metrics were calculated. The entire procedure was then repeated by using the weighted scalar repair approach. To maintain brevity, the thesis will emphasize the results attained with the weighted scalar repair since they are overall the best solutions attained for all of the MOd-KP instances considered. Appendix A provides a summative discussion of the results attained using the maximum ratio approach.

### 3.3 Experimental Design

#### 3.3.1 Algorithm Configurations

For consistency, all three algorithms have been parameterized as similarly as possible. SPEA2 was implemented per the parameter settings suggested by its authors. Zitzler recommends the use of binary coded individuals where each bit represents a decision variable. Mating is performed using one-point crossover with the probability of 0.8. Independent bit mutation is used with each bit being flipped with a probability of 0.006. Table 3.1 provides the detailed parameter settings for all three algorithms. Population size and the archive size parameters vary with prob-
Table 3.1. Algorithm specific parameter settings for $\varepsilon$-hBOA, $\varepsilon$-NSGAII and SPEA2.

<table>
<thead>
<tr>
<th>Algorithm Specific Parameters</th>
<th>$\varepsilon$-hBOA</th>
<th>$\varepsilon$-NSGAII</th>
<th>SPEA2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chromosome Encoding</td>
<td>Binary</td>
<td>Binary</td>
<td>Binary</td>
</tr>
<tr>
<td>Initial Population</td>
<td>1000</td>
<td>Refer Table 3.2</td>
<td>Refer Table 3.2</td>
</tr>
<tr>
<td>Selection</td>
<td>Tournament</td>
<td>Tournament</td>
<td>Tournament</td>
</tr>
<tr>
<td>Mutation Type</td>
<td>NA</td>
<td>Independent Bit</td>
<td>Independent Bit</td>
</tr>
<tr>
<td>Mutation Probability</td>
<td>NA</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Bit Turn Probability</td>
<td>NA</td>
<td>0.006</td>
<td>0.006</td>
</tr>
<tr>
<td>Recombination Type</td>
<td>NA</td>
<td>One Point Crossover</td>
<td>One Point Crossover</td>
</tr>
<tr>
<td>Recombination Probability</td>
<td>NA</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>Termination Criteria</td>
<td>NFE &gt; $10^6$</td>
<td>NFE &gt; $10^6$</td>
<td>NFE &gt; $10^6$</td>
</tr>
<tr>
<td>Epsilon Values</td>
<td>10</td>
<td>10</td>
<td>NA</td>
</tr>
</tbody>
</table>

Table 3.2. Initial population sizes used for various test instances by $\varepsilon$-NSGAII and SPEA2.

<table>
<thead>
<tr>
<th>Problem Specific Parameters (Initial Poulations)</th>
<th>Number of Knapsacks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Number of Items</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>250</td>
</tr>
<tr>
<td></td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>750</td>
</tr>
</tbody>
</table>

The $\varepsilon$-NSGAII has been implemented with similar operators to those of SPEA2. Initial populations used for the $\varepsilon$-NSGAII are the same as those used in SPEA2 with the maximum number of generations capped at 250 for each connected run. Epsilon values indicate the precision required on each of the objectives. Lower epsilon values indicate high precision and a finer resolution more computationally demanding version of a search problem (Kollat and Reed, 2007a) whereas large epsilons yield computational savings at the cost of a more approximate representation.
of the Pareto optimal set. This thesis intends to provide an extensive comparative analysis of knapsack problems, and thus epsilon precision settings for each of the objectives were chosen to result in very high precision Pareto approximation sets for each test instance.

The $\varepsilon$-hBOA replaces crossover and mutation from $\varepsilon$-NSGAII with a Bayesian network model and simulates new populations based on the model. Thus, it eliminates the parameters related to these operators. With the introduction of the Bayesian network model, it is imperative to study the effect of selection, dynamic population sizing, and epsilon dominance archiving on model building and the eventual performance of the algorithm. Kollat et al. (2008a) have tested 5 different configurations of $\varepsilon$-hBOA in a groundwater monitoring application that has a very similar structure to the MOd-KP. Their results have shown that the Bayesian network modeling makes the $\varepsilon$-hBOA very sensitive to its lower bound population (requiring approximately 1000 members as specified in Table 1).

Our rational for using the 1000-member lower bound population size for the $\varepsilon$-hBOA is based on the analysis of Pelikan (2002) which showed that the population size required for optimal model building within hBOA is of the order $O(2^k n^{1.05})$ where $k$ is the order of subproblems in the problem decomposition and $n$ is the number of decision variables. Assuming the lowest bound building block complexity and substituting the value of $n$ based on the problem instance, the smallest recommended population size is approximately 1000 for the MOd-KP instances explored in this work. However, as stated this reflects a lower bound complexity and our revised adaptive population sizing with $\varepsilon$-nondominated archive injection will ensure that population sizes can increase as search progresses.

All three algorithms (SPEA2, $\varepsilon$-NSGAII and $\varepsilon$-hBOA) were run for 1 million ($10^6$) function evaluations per trial. Twenty five random seed trials were conducted for each algorithm to maximize the quality of their results and reduce random seed effects. Twenty four problem instances were solved, twelve of which were the uncorrelated instances presented by Zitzler and Thiele and twelve were the correlated instances introduced in this thesis. Each problem instance required 150 algorithm runs ($25 \times 3 \times 2$) to generate the reference sets and evaluate our performance metrics. Since there are two different problem implementations the total algorithm runs required are 300 ($150 \times 2$) for each instance. Overall this
Figure 3.2. Reference set for the uncorrelated instance of the 4 knapsack test case containing 250 items. Blue cones show the solutions contributed by $\varepsilon$-hBOA, green cones show the solutions contributed by $\varepsilon$-NSGAII and the red cones show the solutions contributed by SPEA2.

experiment was composed of 6900 algorithm runs, 3300 of which were used to generate the reference sets and 3600 to calculate the runtime dynamic values.

3.3.2 Reference Set Generation and Metrics of Performance

Reference sets for the 2 knapsack test case containing 100 items, 250 items, and 500 items and for the 3 knapsack test case containing 100 items for the uncorrelated test instances were made available by Zitzler and Thiele (1999). For the remaining problem instances, the reference sets were generated for each test problem by pooling the nondominated solutions across all runs of the three algorithms. A sample reference set for one of the uncorrelated MOd-KP instances (4 knapsack 250 items) is shown in Figure 3.2.
The objective of this comparative analysis is to evaluate the algorithms in terms of their effectiveness, efficiency and reliability using metrics measuring the proximity and diversity of the algorithms’ approximations to the best known reference tradeoffs. Measuring efficiency indicates the computational costs and the overall search dynamics of the algorithm. The $\varepsilon$-indicator (Zitzler et al., 2003) and hypervolume metrics (Zitzler and Thiele, 1999; Knowles et al., 2006) are used to quantify performance differences between the algorithms. The unary additive $\varepsilon$-indicator is defined as the smallest distance that an approximate set has to be translated in order to completely dominate the reference set. The hypervolume indicator indicates the volume between the approximation set and a reference set with respect to a nadir point. Lower values for all the indicators are preferred as it indicates closer approximation to the reference sets. The Kruskal-Wallis (Conover, 1999) test is used to rank the algorithms within the specified confidence limits.

3.4 Results and Discussion

3.4.1 Enumerated Problem

To highlight the relative computational challenge posed by the two problem instances (correlated and uncorrelated), we introduce a smaller demonstrative instance of the MOd-KP problem with 25 items and 4 knapsacks. The size of the problems was chosen in part because it can be enumerated (solved to completion) by evaluating all $2^{25}$ (or over 33.5 million) possible combinations given current computational constraints. The relative difference in the problem difficulties is assessed using the suite of algorithms (SPEA2, $\varepsilon$-hBOA and $\varepsilon$-NSGAII) on each of the enumerated problem instances (correlated and uncorrelated). The results were obtained across 25 runs of the algorithms where each algorithm used one million function evaluations (less than 3% of the decision space). Note that given the same computational expenditure, the difference in the percentage of the Pareto optimal sets captured (see Table 3.3) strongly demonstrates the increased difficulty of the correlated instance of the MOd-KP.

In Table 3.3, a metric value of one indicates that the algorithms were able to capture 100% of the enumerated Pareto optimal sets within an $\varepsilon$ precision of 10.
Table 3.3. Percentage of Pareto optimal sets captured by the $\varepsilon$-hBOA, the $\varepsilon$-NSGAII and the SPEA2. The values in the table indicate the mean metric value across 25 algorithm runs with the standard deviation mentioned in the brackets.

<table>
<thead>
<tr>
<th></th>
<th>SPEA2</th>
<th>$\varepsilon$-NSGAII</th>
<th>$\varepsilon$-hBOA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Uncorrelated Instance</td>
<td>Correlated Instance</td>
<td>Uncorrelated Instance</td>
</tr>
</tbody>
</table>
| $\varepsilon$-perf (Maximize) | 0 (0) | 0.94 (0.0014) | 0.43 (0.0472) | 0.97 (0.0041) | 0.4 (0.0288) |}

The values in the table indicate the mean metric value across 25 algorithm runs with the standard deviation provided in the brackets. The table indicates that SPEA2 failed to identify any solution in the Pareto set. The percentage of the Pareto set identified by $\varepsilon$-NSGAII and $\varepsilon$-hBOA is 94% and 97% respectively for the uncorrelated instance. However, for the correlated instance, the percentage falls by more than half to 43% and 40% respectively for the two algorithms. This reduced performance for the correlated enumerated test case is expected and demonstrates that the algorithms need more search time to overcome the increase in problem difficulty.

Figure 3.3 provides an example network visualization for some strong hierarchical rules that were proposed by the $\varepsilon$-hBOA to be satisfied by greater than 70% of all of the nondominated solutions for the enumerated Pareto optimal set for the correlated case. In the $\varepsilon$-hBOA network illustration in figure 3.3, the numbered circles represent potential decisions for selecting one of 25 items. Green designates items that are selected and red represents items that are not selected. In figure 3.3, the interior most circles represent the dependent variables in the rules subject to the decisions made in the outer circles. The order of hierarchy for each rule is defined by the number of independent decisions that influence the interior dependent sampling rules. For example, item 10 has a second order hierarchy rule which indicates that it should be selected by greater than 70-percent of nondominated solutions if item 13 and 23 are not selected. The rules provided by the $\varepsilon$-hBOA provide decision-makers with a means of discerning how selection decisions translate to optimal/near-optimal tradeoffs. For real-world applications this might be of immense importance for discovering the dependencies and controls for complex planning problems. It presents the decision-makers with the problem knowledge...
Figure 3.3. The network graphic depicts examples of the hierarchical Bayesian rules proposed by the $\varepsilon$-hBOA when searching for Pareto efficient sampling strategies in the correlated enumerated instance.

that traditional solution tools are incapable of capturing.

To test the quality of the $\varepsilon$-hBOA’s probabilistic models, we can use an analysis of the frequencies of decision variables that compose the enumerated Pareto sets to determine if the rules suggested by the $\varepsilon$-hBOA are approximately correct given that in any given trial run, the algorithm sampled less than 3% of the $2^{25}$ possible solutions. Figure 3.4 provides error histograms for the $\varepsilon$-hBOA’s proposed probabilistic rules relative to their true values within the Pareto set. In each subplot of Figure 3.4, the horizontal axis gives the absolute magnitude of the percentage deviation of the probabilities of the true rules in the Pareto set and the probabilities proposed by the $\varepsilon$-hBOA. Subplots are divided based on the hierarchy of the rules and the problem instance. The top three plots of Figure 3.4 indicate the first three hierarchies for the uncorrelated instance and the remaining three plots correspond to the correlated instance. Subplots indicate that close to 70% of the first order hierarchy rules had errors of less than 20% of their true values. Similar trends were seen across higher order hierarchies. Despite the limited amount of information that the $\varepsilon$-hBOA was provided as it solved these problems, it is able to identify the
Figure 3.4. Error histograms for rules ranging in complexity from 1st to 3rd order of hierarchy for the uncorrelated and the correlated instances. Errors are computed as the absolute percentage deviations between the probabilities proposed by the $\varepsilon$-hBOA for the sampling rules and the actual probabilities that were attained by analyzing the Pareto efficient solution set.

decision variable interdependencies with reasonable accuracy. It should be noted that the mere presence of errors implies that the $\varepsilon$-hBOA has successfully learned rules that exist in the true solution set, and it is very easy to discern the value of model building based on the quality of the algorithm’s approximation sets (i.e., poor models will yield poor search performance).

3.4.2 Results for the Correlated and Uncorrelated Test Instances

Table 3.4 shows the percentage contribution of the three algorithms towards the generation of the new reference sets for the MOd-KP correlated and uncorrelated instances that did not have pre-existing benchmark solution sets. The values indicate that $\varepsilon$-hBOA contributed the most nondominated solutions for the larger problem instances (in terms of objective count and the number of decision variables). The percentage contribution of $\varepsilon$-hBOA varies from 78% to 100% while the contributions of $\varepsilon$-NSGAII and SPEA2 were limited to 0% - 22% and 0% - 13.9%, respectively. For the small uncorrelated instances of the 2 knapsack test
Table 3.4. Percentage contributions of the \( \varepsilon \)-hBOA, the \( \varepsilon \)-NSGAII and the SPEA2 towards the generation of reference sets.

<table>
<thead>
<tr>
<th>Items</th>
<th>Uncorrelated Instances</th>
<th>Correlated Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \varepsilon )-hBOA</td>
<td>( \varepsilon )-NSGAII</td>
</tr>
<tr>
<td>100</td>
<td>3.13</td>
<td>95.31</td>
</tr>
<tr>
<td>250</td>
<td>13.82</td>
<td>84.53</td>
</tr>
<tr>
<td>500</td>
<td>78.58</td>
<td>21.13</td>
</tr>
<tr>
<td>750</td>
<td>99.14</td>
<td>0.00</td>
</tr>
<tr>
<td>100</td>
<td>10.74</td>
<td>88.61</td>
</tr>
<tr>
<td>250</td>
<td>69.01</td>
<td>30.99</td>
</tr>
<tr>
<td>500</td>
<td>100</td>
<td>0.00</td>
</tr>
<tr>
<td>750</td>
<td>100</td>
<td>0.00</td>
</tr>
<tr>
<td>100</td>
<td>75.66</td>
<td>24.34</td>
</tr>
<tr>
<td>250</td>
<td>93.86</td>
<td>5.81</td>
</tr>
<tr>
<td>500</td>
<td>99.24</td>
<td>0.1</td>
</tr>
<tr>
<td>750</td>
<td>99.26</td>
<td>0.1</td>
</tr>
</tbody>
</table>

case containing 100 items and 250 items, \( \varepsilon \)-NSGAII is the major contributor with 95.31\% and 84.53\% solutions respectively. The entire reference sets for the correlated instances of 4 knapsack 250 items, 500 items, and 750 items problems were contributed by the \( \varepsilon \)-hBOA.

Figure 3.2 visually illustrates the best known approximation set for the 4 knapsack 250 item problem instance. Figure 3.2 shows the predominant contribution of the \( \varepsilon \)-hBOA (indicated by the blue points) to the profits for knapsack 1, knapsack 2 and knapsack 3 plotted on the X, Y and Z axes, respectively. The fourth objective is represented by the orientation of the solution cones. Cones with an upward orientation represent high profits on knapsack 4, and conversely the cones with a downward orientation represent low profits. The color of the cones reflects which algorithm contributed each particular solution (blue represents \( \varepsilon \)-hBOA, red repre-
Table 3.5. Performance of the \( \varepsilon \)-hBOA, the \( \varepsilon \)-NSGAII and the SPEA2 with respect to the \( \varepsilon \)-indicator and hypervolume metrics for the uncorrelated problem instances. Highlighted values designate the statistically significant (to the 95-percent confidence level) best performing algorithm for the problem instance.

<table>
<thead>
<tr>
<th>Multiplication Factor</th>
<th>Epsilon Indicator - Additive</th>
<th>Hypervolume</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \varepsilon )-hBOA</td>
<td>( \varepsilon )-NSGAII</td>
</tr>
<tr>
<td>2 Knapsacks</td>
<td>100</td>
<td>13.4 (3.76)</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>15.25 (2.05)</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>19.96 (3.27)</td>
</tr>
<tr>
<td></td>
<td>750</td>
<td>34.20 (6.82)</td>
</tr>
<tr>
<td>3 Knapsacks</td>
<td>100</td>
<td>31.16 (2.37)</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>37.49 (4.35)</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>86.12 (8.19)</td>
</tr>
<tr>
<td></td>
<td>750</td>
<td>111.35 (6.85)</td>
</tr>
<tr>
<td>4 Knapsacks</td>
<td>100</td>
<td>69.36 (9.70)</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>118.58 (7.65)</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>238.88 (23.86)</td>
</tr>
<tr>
<td></td>
<td>750</td>
<td>519.20 (53.86)</td>
</tr>
</tbody>
</table>

In the table, \( \varepsilon \)-hBOA, \( \varepsilon \)-NSGAII and SPEA2 are the algorithms being compared. The values in the table represent the performance metrics. The statistical analysis includes the Kruskal-Wallis nonparametric test to determine the statistical significance of performance differences at the 95% confidence level. The results are used to rank the algorithms according to their performance.

Table 3.5 and 3.6 show the end-of-run performance values attained by each algorithm for the unary \( \varepsilon \)-indicator-additive and the hypervolume metrics. Readers are reminded that hypervolume metric measures the volume between the approximation set and the best known reference set with respect to a nadir point. Values in the table show the average metric value for 25 random seed trials and standard deviations are included in the parentheses. In addition, a Kruskal-Wallis nonparametric statistical test was used to test the statistical significance of performance differences at the 95% confidence level for the metric distributions attained by each algorithm. The results of the statistical tests were used to rank the relative performances of each algorithm.

For the uncorrelated instances (see Table 3.5) the \( \varepsilon \)-hBOA seems to be the best performer for most of the test instances followed by \( \varepsilon \)-NSGAII and SPEA2 being highlighted. This indicates that \( \varepsilon \)-hBOA is particularly effective for this type of problem. The \( \varepsilon \)-NSGAII and SPEA2 also perform well, but \( \varepsilon \)-hBOA consistently outperforms them.
Table 3.6. Performance of the $\varepsilon$-hBOA, the $\varepsilon$-NSGAII and the SPEA2 with respect to the $\varepsilon$-indicator and the hypervolume metrics for the correlated problem instances. Highlighted values designate the statistically significant (to the 95-percent confidence level) best performing algorithm for the problem instance.

<table>
<thead>
<tr>
<th>Epsilon Indicator - Additive</th>
<th>Hypervolume</th>
<th>Multiplication Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$-hBOA</td>
<td>$\varepsilon$-NSGAII</td>
<td>SPEA2</td>
</tr>
<tr>
<td>2 Knapsacks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>18.44 (2.08)</td>
<td>18.32 (3.53)</td>
</tr>
<tr>
<td>250</td>
<td>72.06 (8.54)</td>
<td>70.04 (8.89)</td>
</tr>
<tr>
<td>500</td>
<td>141.12 (8.95)</td>
<td>88.36 (15.31)</td>
</tr>
<tr>
<td>750</td>
<td>230.44 (22.19)</td>
<td>192.24 (17.63)</td>
</tr>
<tr>
<td>3 Knapsacks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>21.36 (3.66)</td>
<td>29 (3.07)</td>
</tr>
<tr>
<td>250</td>
<td>42.2 (7.35)</td>
<td>121.04 (23.60)</td>
</tr>
<tr>
<td>500</td>
<td>297.84 (18.82)</td>
<td>272.60 (25.46)</td>
</tr>
<tr>
<td>750</td>
<td>310.95 (24.20)</td>
<td>423.08 (38.10)</td>
</tr>
<tr>
<td>4 Knapsacks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>30.24 (4.85)</td>
<td>43.68 (3.87)</td>
</tr>
<tr>
<td>250</td>
<td>55.08 (8.16)</td>
<td>136.40 (7.54)</td>
</tr>
<tr>
<td>500</td>
<td>113.88 (22.64)</td>
<td>370.5 (22.65)</td>
</tr>
<tr>
<td>750</td>
<td>171.20 (27.68)</td>
<td>555.80 (32.79)</td>
</tr>
</tbody>
</table>

The worst performer. For the two and three objective problems with 100 items, the $\varepsilon$-NSGAII outperforms $\varepsilon$-hBOA and SPEA2 on both the metrics. The $\varepsilon$-NSGAII’s dynamic population sizing generates diverse approximation sets within close proximity of the reference tradeoffs yielding its statistically superior performance on the smaller problem instances. SPEA2 failed to converge with respect to $\varepsilon$-hBOA and $\varepsilon$-NSGAII.

Correlated instances of the MOd-KP add an inherent structure to the problem and provide a difficult suite of combinatorial problems to benchmark algorithm performance. For the correlated instances shown in Table 3.6, the $\varepsilon$-hBOA is shown to be the statistically superior performing algorithm across most of the MOd-KP instances tested. The $\varepsilon$-NSGAII demonstrates superior performance on the two objective instances of the problem however $\varepsilon$-hBOA outperforms $\varepsilon$-NSGAII on the three and four objective instances. The superior performance of the $\varepsilon$-hBOA indicates of increased difficulty in the problem and implies a strongly interdependent decision structure. Overall the $\varepsilon$-NSGAII is the second best performing algorithm and the SPEA2 is the worst performing algorithm. There are a few cases for the
hypervolume metric where the $\varepsilon$-NSGAII and the $\varepsilon$-hBOA have statistically similar performance (and thus have not been highlighted). The superior performance of the $\varepsilon$-hBOA on correlated instances demonstrates the PMBGA's ability to discover and exploit problems' dependency structures to generate promising solutions.

Figure 3.5 shows $\varepsilon$-indicator success rate plots for each algorithm on each test instance. The success rate plots show the cumulative distribution functions that define the percentage of random seed trials for a given algorithm that were able to meet or exceed a threshold value of the $\varepsilon$-indicator metric as a function of the number of function evaluations (NFE) utilized. The success rate plots provide valuable insights into the probabilistic search dynamics of the algorithms. The $\varepsilon$-indicator threshold values were selected to be the 85% quantile values attained from the random trials of the best performing algorithm for each MOd-KP instance. The slope or steepness of each distribution is a measure of the reliability of the algorithm for a given level of computational effort. A vertical distribution signifies a highly reliable performance where all of the algorithm’s random seed trials exceeded the threshold performance with a minimum variance in computational effort (i.e., nearly the same NFE in all trials). Similarly, small slopes or horizontal distributions signify low or zero reliability, respectively. Figure 3.5A represents the success plots of 12 instances of the uncorrelated problem and Figure 3.5B represents the success plots of the 12 instances of the correlated problem. Each row (moving left to right) indicates an increase in the number of objectives (2, 3, 4) for a given number of decision variables. Columns of the sub-plots (moving from the top to the bottom) represent an increase in the number of decision variables (100, 250, 500, 750) for a given number of objectives. Blue solid lines indicate the success rates of $\varepsilon$-hBOA and the green dotted lines highlight the success rates of $\varepsilon$-NSGAII. Red dotted lines are used to highlight the success rate of SPEA2 but since SPEA2 could never achieve the 85% quantile of the best performing algorithm across all the algorithm runs of the 24 problems instances, its success rate stays at zero.

For the small instances of the uncorrelated problem (2 knapsacks 100 items, 250 items and 3 knapsacks 100 items) both the $\varepsilon$-hBOA and $\varepsilon$-NSGAII seem to reliably generate approximate sets close to the best performing algorithm. For the slightly larger problems instances (2 knapsack 500 items, 750 items and 3 knapsack
Figure 3.5. Success plots for the 24 problem instances. The plots indicate the probability of each algorithm to cross the 85% quantile on $\varepsilon$-Indicator additive metrics for each problem instance.
250 items), the $\varepsilon$-NSGAII has a decrease in reliability as compared to the $\varepsilon$-hBOA. The $\varepsilon$-hBOA achieves the specified threshold within 150,000 function evaluations whereas $\varepsilon$-NSGAII takes close to 500,000 function evaluations to reliably achieve the specified threshold. The differences in performance are more apparent in larger problems with the $\varepsilon$-NSGAII showing significant decreases in its efficiency and effectiveness. The $\varepsilon$-NSGAII fails completely on the larger instances of the three and four objective problems. The $\varepsilon$-hBOA shows robust performance by achieving the specified threshold within 300,000 function evaluations across almost all of the uncorrelated test instances.

The correlated instances of the problem show similar differences in the performances of the three algorithms. For the two objective problems, the $\varepsilon$-NSGAII and the $\varepsilon$-hBOA are fairly reliable at achieving the specified $\varepsilon$-indicator thresholds within 200,000 function evaluations. It is interesting to note that even when the $\varepsilon$-hBOA is not the best performing algorithm, it still gets reliably close to the best performing algorithm at a fairly early stage of evolution. For the three objective problems, the $\varepsilon$-NSGAII shows a substantial decrease in the reliability, the only exception being the 3 knapsack 500 items test case where it is the best performing algorithm and achieves the threshold at almost the same rate as $\varepsilon$-hBOA. The four objective problems highlight the dominance of the $\varepsilon$-hBOA. The $\varepsilon$-hBOA achieves the threshold the fastest whereas the $\varepsilon$-NSGAII seems to struggle to achieve the threshold at all. For the 4 knapsack 100 items and 250 items test case, the probability of the $\varepsilon$-NSGAII attaining the 85% quantile thresholds is 0.6 and 0.3, respectively. Success rate plots indicate that the $\varepsilon$-NSGAII is not reliable and tends to fail completely on the larger instances. On the other hand the $\varepsilon$-hBOA is observed to achieve the success thresholds fairly consistently independent of problem size. This implies that the success of the $\varepsilon$-hBOA is influenced by the quality of its models and the amount of statistical information available and less so by the MOd-KP’s size and its constraints. This can be an important advantage while solving large application problems. The $\varepsilon$-hBOA’s reliability may also lead to the reduction or complete elimination of the random seed trials yielding a dramatic reduction in the computational burden of solving MOd-KP problems.

The success plots also give insights on how increasing the number of objectives for a given number of decision variables or vice-versa impacts search. As the
Table 3.7. The best performing algorithm in each of the test instance. The decision is based on the statistical analysis of the performance metrics and the run-time success dynamics of the three algorithms.

<table>
<thead>
<tr>
<th></th>
<th>Uncorrelated Instance</th>
<th>Correlated Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε-hBOA</td>
<td>ε-NSGAII</td>
<td>SPEA2</td>
</tr>
<tr>
<td>2 Knapsacks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>250</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>500</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>750</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>3 Knapsacks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>Tie between ε-hBOA and ε-NSGAII</td>
<td>No</td>
</tr>
<tr>
<td>250</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>500</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>750</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>4 Knapsacks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>250</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>500</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>750</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

number of objectives increases (moving left to right in subplots) the time taken for each algorithm to achieve the threshold increases. The ε-hBOA shows only a slight increase in the required number of the function evaluations but the ε-NSGAII requires a substantial increase in the number of function evaluations. The reliability of the ε-hBOA stays fairly constant as it achieves the threshold at almost the same NFE across all of its random seed trial runs. The runtime variance across the random seeds for the ε-NSGAII increases significantly as the number of objectives increases. Similar trends are observed as the number of decision variables are increased (moving from the top to the bottom) for a given number of objectives. This comparison indicates that the ε-hBOA is robust in its performance independent of the number of objectives or decision variables.

Table 3.7 summarizes the performance of the three algorithms for each problem instance. Based on the statistical analysis of the performance metrics and the run-time success plots it highlights the best performing algorithm for each problem instance. It shows that SPEA2 does not outperform ε-hBOA and ε-NSGAII in any of the problem instances. The performance of the ε-NSGAII is critically dependent on the size of the problem instance. It is expected to perform fairly well for two objective instances of the problem and struggles to maintain its performance for
larger instances (3 and 4 objective instances). The ε-hBOA is outperformed by ε-NSGAII on the two-objective instances of the problem but is the best performing algorithm for the larger instances of the problem. Thus based on the problem size the reader can pick the best performing algorithm for his/her application.
CHAPTER 4

THE LONG-TERM GROUNDWATER MONITORING STUDY

4.1 Introduction

Real-world budgetary constraints within observation network design problems yield resource allocation conflicts across space, time, and competing foci that are equivalent in form to the multiobjective d-dimensional knapsack problem (MOd-KP). One such example of a real-world network design problem is the groundwater monitoring problem. In general, groundwater monitoring design has been shown to be a challenging optimization problem with multiple conflicting objectives and very large discrete decision spaces (ASCE, 1990; Knopman and Voss, 1989). With the financial constraints associated with the costs of sampling at various well locations there is a distinct similarity between the groundwater monitoring problems and MOd-KP (Knopman and Voss, 1989; Knopman et al., 1991).

The knapsack study showed that while the ε-hBOA can provide a high quality approximations, it also generates an enormous amount of information within its probabilistic models. The statistical information is inherently used by the algorithm to generate promising solutions, but the same information can again be used to visually interpret the interdependencies in the problem structure. An understanding of the existing dependencies gives the user valuable insights about the problem and their applications. Efficient methods of exploiting and presenting the
information captured in the $\varepsilon$-hBOA’s Bayesian network models has significant potential to enhance the formulation and solution of environmental monitoring problems. In this chapter we carefully test and verify the ability of $\varepsilon$-hBOA to capture the dependency structure for a groundwater monitoring test where it is feasible to fully enumerate the true Pareto optimal set.

4.2 Test Case

The long term groundwater monitoring (LTM) test case explored in this chapter has been used extensively to represent a complex, highly characterized groundwater monitoring site (Fogg et al., 1998; Maxwell et al., 2000, 2008) and is based on 50 million node flow and transport simulation representing the migration of a hypothetical perchloroethylene (PCE) plume originating from an underground storage tank. The site’s hydrogeology has been extensively characterized and concentration data are available for 25 pre-determined sampling locations with no more than three sampling ports available along its vertical axis. Each sampling location (i.e., well) is treated as a decision such that if a sampling location is selected, then all the available sampling ports present at that location are utilized. Thus the problem has 25 decision variables and can be enumerated to establish the true Pareto set. Knowledge of the true Pareto set provides rigorous a measure to assess the quality of the probabilistic sampling rules generated by the $\varepsilon$-hBOA.

The LTM test case seeks to balance 4-objectives simultaneously, each of which are to be minimized. The design objectives included: (1) the normalized cost of sampling the contaminant plume ($COST$), (2) the concentration estimation error relative to using all available data points to create spatial plume maps ($ERROR$), (3) the spatial uncertainty associated with characterizing the plume ($UNCERT$), and (4) the error associated with estimating the mass of contaminant in the plume ($MASS$). Readers interested in the actual equations used to quantify each of these objectives can refer Kollat and Reed (2006, 2007a). Figure 4.1 provides a visualization of the 4-objective enumerated true Pareto front for the 25 well test case. The $COST$, $ERROR$, and $UNCERT$ objectives are plotted on the X-, Y-, and Z-axes, respectively. Color is used to represent $MASS$ objective where red solutions indicate high mass error and blue solutions indicate low mass error. There are $n =$
25 binary spatial sampling decisions yielding a total of $2^{25}$ or 33.6 million sampling possibilities, all of which were evaluated to attain Figure 4.1. It should be noted that prior work has rigorously demonstrated $\varepsilon$-hBOA's successful solution of this test case and others of much higher difficulty Kollat et al. (2008a).

4.3 Methodology

4.3.1 Discovering and Exploiting Hierarchical Dependency Structure

As explained earlier in Chapter 2, $\varepsilon$-hBOA is a probabilistic model building genetic algorithm. It uses a set of directed acyclic graphs to model the conditional
probabilities between decision variables for generating non-dominated solutions. The statistical models seek to exploit the structure of the problems being solved and use this information for proper mixing and exploration of their search space. As the evolution proceeds the graphs and associated conditional probabilities are refined based on the search results. Changes in the model quality are measured using the Minimum Description Length (MDL) metric. Network operations like edge additions, edge removals and edge reversals are performed to improve the predicative ability of the model. Once the elementary network operations do not provide significant improvements to the model the Bayesian network model building is terminated. When the structure and parameters of the Bayesian network have been estimated in a given generation, new candidate solutions are then proposed according to the distribution encoded in the learned network (Pelikan, 2002). After the child population is generated the model building process starts again to evolve the next generation.

The information within the Bayesian network models is represented as the decision trees for each associated decision variable (termed a node). The decision tree represents the hierarchical “if and only if” structure for the corresponding decisions. Figure 8 shows an example of the decision tree. A decision tree is defined by three parameters: root, parent, and leaf. All decisions in the tree are initiated from the root. The root node has no parents and each node in the tree has exactly one parent. Successive decisions progress from the root to the nodes along the edges. The nodes are split to reveal a leaf that stores the information of interest (in this case the conditional probabilities). Since there is only one possible way of traversing through a tree to a leaf, each leaf stores the conditional probabilities associated with all the instances that end up the traversal of the tree in that leaf. In other words each leaf stores the conditional probabilities of the root given that the nodes contained in the path from root to leaf are fixed according to the path. For example in Figure 4.2 the decision variable for well 9 acts as the root for the decision tree. Decision variable for wells 10, 8 and 18 acts as the nodes for the decision tree and variables 0.62, 0.39, 0.58 and 0.46 refer to the leaves of the decision tree. The “0” or “1” values along the edges represents that a given well is either not sampled or sampled. The leaf 0.62 refers to the conditional probability associated with the root (well 9) given that the well 10 and 8 are fixed to the
values of 1 (are sampled). In simple terms the hierarchical rule states that well 9 (the root) will be sampled in nondominated solutions with a probability of 0.62 (the leaf) if and only if wells 8 and 10 are also sampled. Similar rules represent alternative sampling strategies designated by the other leaves (0.39, 0.58 and 0.46). The number of nodes between the root node and the leaf reflect the hierarchy of the rule associated with the corresponding leaf. In the above stated example there are two nodes (wells 8 and 10) between the root node (well 9) and leaf (0.62). Thus it is a second level hierarchical rule because for the rule to be true two conditionally dependent decisions (sampling of wells 8 and 10) must occur. The rule associated with leaf 0.46 is a first order hierarchical rule and the 0.58 and 0.39 leaves represent third order hierarchical rules.

Figure 4.2 shows a simple decision tree example. However as the evolution proceeds the models can become more complicated and consequently so do their component decision trees. Elementary network operations like edge additions and edge reversals add multiple levels of hierarchies to the decision trees. The MDL metric balances predictive skill and model complexity by explicitly penalizing any growth Bayesian network model complexity that does yield a significantly better rules for generating nondominated children. Thus it restricts the size of the decision graphs within the Bayesian network models.

The dependency structure identified by ε-hBOA for the ground water monitor-
ing problem were far more complex than anticipated given the simplicity of the test case. There is an enormous amount of data captured within these models which can be analyzed and presented to a decision-maker. In this analysis we focus on identifying the probabilistic rules that correctly discover and exploit the conditional dependencies between the Pareto optimal sampling decisions for the four objective groundwater monitoring application. To perform in-depth analysis of the probabilistic rules associated with the problem the 25 algorithms runs (200000 function evaluations per run) are performed and all the models are jointly analyzed.

4.3.2 Visualization Tool for the Probabilistic Model

The probabilistic model visualization framework has been developed as a plugin to an interactive visualization tool called Aerovis, developed to explore the objective and decision space for a problem. It is an extremely useful tool to understand and explore the objective tradeoffs of various many objective problems (Kollat and Reed, 2007b). Aerovis was developed using the Python programming language. Python is a dynamic, object-oriented scripting language, which integrates easily with other languages, and is virtually platform independent (Chun, 2006). Additionally, Python has a large number of graphical user interface (GUI) frameworks, including Tkinter, PyQt, and wxPython. Aerovis was developed using the Tkinter (Python’s standard GUI package), which is built on top of Tcl/Tk and is portable across Windows, Unix, and Mac platforms.

Python has a large standard library, commonly cited as one of Python’s greatest strengths, providing pre-written tools suited for many tasks. Besides being easy-to-use and excellent GUI development capabilities, it has an extremely active developer community. With a wide variety of tools provided by the standard library, combined with the ability to use a lower-level language such as C and C++, which is already capable of interfacing between other libraries, Python is a powerful glue language between languages and tools. The user developed Python modules/packages enhances its power and versatility by providing 9178 packages across all disciplines to interact with various libraries and applications.

The Aerovis plugin interacts with Graphviz, an open source graph visualiza-
tion software to display the Bayesian network models. Graphviz (short for Graph Visualization Software) (Ellson et al., 2003; Gansner and North, 2000) is a package of open source tools initiated by AT&T Research Labs for drawing graphs specified in DOT language scripts (Gansner et al., 1993). Graphviz is free software licensed under the Common Public License and also provides libraries for software applications to use the tools. It has various pre-built graph layout programs which represent relationships between data and can be utilized to effectively visualize Bayesian network models. The combination of python packages like Networkx, PyGraphviz, Pydot, and Matplotlib act as an interface between Aerovis and Graphviz. Networkx and Matplotlib create and manipulate the structure and dynamics of complex Bayesian networks. They are used to generate the directed graphs based on the networks generated by the $\varepsilon$-hBOA. Manipulations like checking the directional consistencies in the graph, maintaining equal edge lengths, node locations and so on are performed using the Networkx package. PyGraphviz and Pydot act as a Python interface to Graphviz's Dot language. PyGraphviz and Pydot create, edit, read, write, and draw graphs using the Graphviz graph data structure and layout algorithms. Various layout algorithms provided by Graphviz are dot, neato, fdp, sfdp, twopi and circo. The algorithms useful from the perspective of visualizing Bayesian model are dot and twopi. Dot is a hierarchical or layered drawing of directed graph. The layout algorithm aims edges in the same direction (top to bottom, or left to right) and thus provides graphs similar to the illustration shown in Figure 4.2. Twopi is a hierarchical directed graph arranged in a radial fashion. The nodes are placed on concentric circles with farther nodes representing higher orders of hierarchy. Given the high complexities and large number of networks generated by the $\varepsilon$-hBOA the twopi algorithm offers better utilization of space while conveying the information and is thus used to develop the framework. Figure 4.3 shows an example of Bayesian network visualized through the twopi algorithm of Graphviz. The afore mentioned packages are all freely available for download as Python package index libraries.
Figure 4.3. Strong rules derived from a Bayesian network. The numbered circles represent potential decisions for sampling one of 25 wells. Green designates wells that are sampled and red represents locations that are not sampled. The interior most circles represent the dependent variables in the rules subject to the decisions made in the outer circles. If the conditions on the outer circles are met there is a probability of 98% or more that the dependent well represented by the innermost circle will be sampled.

4.4 Do Our Observation Choices Have Strong Dependency Structure?

Although our prior studies have demonstrated that the $\varepsilon$-hBOA can solve the simple 25-well monitoring test case, they do not provide detailed insights on the value of the algorithm’s hierarchical Bayesian network models. Exploring the hierarchical Bayesian network models is an interesting experiment because the overall dependency structure is arguably minimally complex given the use of non-economic cost functions and Kriging-based design evaluations. Kriging has fewer space-time system dependencies in its linear estimation of the PCE concentration state versus physical flow-and-transport modeling. For this example, we would expect more complex socio-physical objectives and evaluation schemes to increase linkages in observation decisions across space-time.
To elucidate this issue, in this thesis we explore the type and quality of probabilistic dependency networks proposed by the \( \varepsilon \)-hBOA to model what constitutes a nondominated sampling strategy for the test case's 4-objectives. Figures 4.3 and 4.4 provide network visualizations of hierarchical rules proposed by the \( \varepsilon \)-hBOA to be satisfied by greater than 98% of all of the nondominated solutions for the groundwater monitoring test case. In reality, the algorithm provides a joint pdf model of rules for a full range of probabilities. We focus only on the strong rules (i.e., rules suggested at probabilities greater than 98%) because given the high number of rules identified by the algorithm a decision-maker would only be interested in analyzing the rules that lead to a high probability of selecting or not selecting a well. For example a rule which states that when wells 11 and 13 are selected there is a 99% chance that well 9 will be selected supersedes in importance (priority of analysis) to a rule that states when wells 10 and 8 are selected there is 62% chance that well 9 will be selected.

Figures 4.3 and 4.4 are drawn from the probabilistic model generated from a single algorithm run. In Figures 4.3 and 4.4 the numbered circles represent potential decisions for sampling one of 25 wells. Green designates wells that are sampled and red represents locations that are not sampled. Figure 4.3 shows all of the rules that lead to a well being sampled, and Figure 4.4 represents all of the rules that lead to a well not being sampled. The interior most circles represent the dependent variables in the rules subject to the decisions made in the outer circles. The order of hierarchy for each rule is defined by the number of independent decisions that influence the interior dependent sampling rules. For example (as shown in Figure 4.3, Rule A), well 22 has a zeroth order hierarchy rule which indicates that it should be sampled by greater than 98-percent of nondominated solutions independent of all other wells. In Figure 4.3 an example of first order hierarchy rule (Rule B, in Figure 4.3) proposes, if well 1 is selected than well 24 should also be selected in more than 98-percent of the solutions that compose the Pareto optimal set. An illustration of a second order rule in Figure 4.3 (Rule C) is, if well 1 and well 18 are selected then well 20 is also selected. The \( \varepsilon \)-hBOA proposes rules of up to the 7th order hierarchy for sampling well 2 in figure 4.3, which is striking given the simplicity of the test case. Note that well 2 has multiple rule instances which represent alternative independent decisions that would motivate
Figure 4.4. Strong rules derived from a Bayesian network. The numbered circles represent potential decisions for sampling one of 25 wells. Green designates wells that are sampled and red represents locations that are not sampled. The interior most circles represent the dependent variables in the rules subject to the decisions made in the outer circles. If the conditions on the outer circle are met there is a probability of 98% or more that the dependent well represented by the innermost circle will be sampled.

that the location be sampled. Some of these rules can be converted into heuristic rules to identify nondominated solutions. For example (Rule D) when decision-makers sample well 11 they should also sample well 21. The rules which dictate elimination of the wells are equally important as rules that dictate the sampling of wells. Figure 4.4 shows an assortment of rules which dictate the elimination of wells. It should be remembered that only strong rules are displayed in Figures 4.3 and 4.4. Various combinations of these rules enable the algorithm to generate the nondominated set. Prior studies (Kollat et al., 2008a) have shown the superiority of $\varepsilon$-hBOA in solving the 25 well test problem and thus add credibility to the models created by the algorithm.

While Figures 4.3 and 4.4 show the hypothesized probabilistic linkages and controls for observations to be nondominated across a single algorithm run, Figures 4.5, 4.6 and 4.7 shows probabilistic linkages identified across 25 runs of the
Figure 4.5.  The first-order hierarchy rules generated across 25 Bayesian networks. Green and red represents wells being sampled and not sampled respectively. Rules are to be read from from the outermost edge towards center with the interior most circles represent the dependent variables in the rules subject to the decisions made in the outer circles. Rules A,B,C are displayed in their spatial representation in the figure 4.8.

algorithm sorted based on orders of hierarchies. This is a significant departure and advancement from classical sensitivity analyses (e.g., see the methods reviewed by Tang et al. (2007)) in that we are seeking to determine a joint pdf model of all of the dependencies controlling non-domination for many-objectives simultaneously (versus the standard exploration of each metric independently).

Figure 4.5 shows all the distinct first order hierarchical rules identified across 25 algorithm runs. It should be noted that many of these rules were identified by multiple algorithms runs and this figure highlights only the distinct rules identified. First-order hierarchy rules are easy to understand and can be easily used to create simple heuristics for the problem. However the challenges posed in the use and interpretation of rules grow as the order of hierarchy increases. Figure 4.6 highlights the distinct second-order hierarchical rules identified across 25 algorithm runs. A
Figure 4.6. The second-order hierarchy rules generated across 25 Bayesian networks. Green and red represent wells being sampled and not sampled respectively. Rules are to be read from the outermost edge towards center with the interior most circles representing the dependent variables in the rules subject to the decisions made in the outer circles.
Figure 4.7. The tenth-order hierarchy rules generated across 25 Bayesian networks. Green and red represents wells being sampled and not sampled respectively. Rules are to be read from from the outermost edge towards center with the interior most circles represent the dependent variables in the rules subject to the decisions made in the outer circles.

A total of 98 distinct second-order hierarchy rules were identified by the $\varepsilon$-hBOA. Similar characteristics were seen for the third and higher order hierarchies. The number of rules increases from 98 for second-order to 260 corresponding to the fifth-order of hierarchy. The large number of higher order probabilistic linkages highlight the capability of the algorithm to model the complex decision space and generate the nondominated solutions. Figures 4.7 highlights the distinct tenth order hierarchy rules.

Figure 4.8 provides a spatial visualization of the three first-order hierarchy rules highlighted in Figure 4.5. The $\varepsilon$-hBOA's proposed rules are sources of potential hypotheses on system behavior when you place them within the natural space-time contexts of environmental systems. Figure 4.8 provides contextual meaning
for some of the rules (Rules A, B, C) discussed in Figure 4.5. For example, wells 1 and 11 jointly sample the defining boundaries of the plume (Rule A). Similarly, if well 18 is not sampled along the longitudinal axis of plume than the $\varepsilon$-hBOA’s rule proposes that well 20 should be, which makes intuitive since it is the next closest well near the mid-line of the PCE plume (Rule B). Likewise, well 15 is a suggested substitute for well 14 (Rule C). Broadly, the network graphics in Figure 4.8 provide a classification of how sensitive nondominated sampling strategies are to a range of interdependent sampling decisions. The rules provided by the $\varepsilon$-hBOA provide decision-makers with a means of discerning which sampling decisions have broad impacts over the full plume (e.g., well 1 in source area) versus those that have more localized effects (well 22 on the edge of the domain). There are two important issues to note across the Figures 4.3 - 4.8. First, although some of the simpler rules may seem intellectually trivial, they represent problem knowledge that traditional solution tools are incapable of capturing, thus enabling the $\varepsilon$-hBOA to deal with severely interdependent problems (a form of severe nonlinearity termed epistasis). Secondly, the more complex high order hierarchical rules (as highlighted in Figures 4.6 and 4.7) pose interesting hypotheses on the relationships and controls impacting how alternative sampling strategies can attain near optimal tradeoffs between the test case’s four objectives.

The Bayesian network generated by $\varepsilon$-hBOA forms the basis of the nondominated set generated by the algorithm. The solutions are sampled from the model’s joint probability distribution. The successful search results of Kollat et al. (2008a) would not be possible if the quality of the $\varepsilon$-hBOA’s models were not sufficient. To further this point and test the quality of the $\varepsilon$-hBOA’s probabilistic models, we can use an analysis of the sampling decisions within the true Pareto set in Figure
Figure 4.9. Verification of strong rules with respect to the Pareto set. The strong rules correspond to the rules with condition probabilities greater than 0.98. Each subplot represents deviations observed with respect to the Pareto set for each level of hierarchies.

4.1 to determine if the rules suggested by the $\varepsilon$-hBOA are approximately correct given that in any given trial run, the algorithm sampled less than 0.5% of the $2^{35}$ possible solutions. Figure 4.9 provides error histograms for the $\varepsilon$-hBOA’s proposed strong probabilistic rules relative to their true values within the Pareto set. Again the term strong rule refers to rules that predicted to be satisfied in the Pareto optimal set in greater than 98% of their occurrences. In each subplot of Figure 4.9, the horizontal axis gives the absolute magnitude of the percentage deviation of the probabilities of the true rules in the Pareto set and the probabilities proposed by the $\varepsilon$-hBOA. The vertical axis indicates the percentage of rules within each error domain. Subplots are divided based on hierarchy of the rules with the first subplot indicating the performance of first order hierarchy rules. The first subplot indicates that close to 70% of the first order hierarchy rules had errors of less than 10% of their true values and 20% of the rules had errors in the range 10-20% of their true values. It should be noted that the presence of errors implies that the $\varepsilon$-hBOA has successfully learned rules that exist in the true solution set, which represents an important contribution.

Across all of the orders of hierarchy that the $\varepsilon$-hBOA proposed, 80% of the
Figure 4.10. Verification of all the rules with respect to the Pareto set. Each rule irrespective of the conditional probabilities associated with it is verified against the Pareto set and percentage deviations are plotted. Each subplot represents deviations observed with respect to the Pareto set for each level of hierarchies.

predicted probabilities in the algorithm’s proposed strong rules were within 20% of their true values. Sixty percent of all of the $\varepsilon$-hBOA’s proposed rules actually had errors no greater than 10% of their true values. If the algorithm fails to correctly model problems’ hierarchical dependencies, it is fairly easy to diagnose because its search will fail to make continued progress when solving challenging multiobjective problems. Interestingly, the algorithm proposed 21 strong rules at the ninth-order of hierarchy. This was an unexpected result; we did not anticipate that this level of interdependency would exist for the small monitoring test case.

Expanding focus beyond strong rules enables a more comprehensive assessment of all the rules proposed by the $\varepsilon$-hBOA irrespective of their associated probabilities. Figure 4.10 reflects the performance of the models across all orders of hierarchy and associated probabilities. Across all the orders of hierarchy, 90% of the predicted probabilities in the algorithm’s proposed rules were within 20-percent of their true values. To put 90% into perspective let us consider the number of rules identified. With the number of rules ranging as high as 687 for the fourth order of hierarchy the bar plot indicates that 618 (90% of 687) rules were within 20% of
their true values within enumerated Pareto optimal set.
CHAPTER 5

CONCLUSIONS AND FUTURE WORK

This thesis compares the performance of two state-of-the-art MOEAs (\(\varepsilon\)-NSGAII and SPEA2) to a PMBGA (\(\varepsilon\)-hBOA) on various instances of multiobjective d-dimensional knapsack problems (MOd-KP). The MOd-KP is a combinatorial optimization problem with applications to capital budgeting, resource allocation and various other fields. The uncorrelated instances of the MOd-KP introduced by Zitzler and Thiele have been used extensively to benchmark algorithm performances on combinatorial problems. This thesis contributes correlated instances of the MOd-KP which are relatively harder to solve and better capture the problem dependency structure likely to occur in the real-world applications. Overall the three algorithms were tested across 24 test instances of MOd-KP with the number of objectives ranging from 2 to 4 and the number of decision variables ranging from 100 to 750. The reference sets were generated across the best known solutions from all the runs of the three algorithms and a comprehensive metrics-based analysis was performed to measure the efficiency and reliability of the three algorithms.

The results of the thesis demonstrate the superior performance of the \(\varepsilon\)-hBOA at finding the reference sets. The success of the \(\varepsilon\)-hBOA in this thesis demonstrates its effectiveness at accounting for the complex interdependency structure within the knapsack problems. The key difference between the \(\varepsilon\)-hBOA and other MOEAs is its Bayesian network model building capability. It generates probabilistic statistical models from which it is able to learn and exploit the dependency structure present in the problem. The relatively poor performance of the \(\varepsilon\)-NSGAII
and the SPEA2 on larger instances highlights scaling limitations present in traditional MOEAs. The high reliability of the $\varepsilon$-hBOA can serve to eliminate or reduce the random seed trials necessary for attaining high quality approximations to the MOd-KP, thereby dramatically reducing the computational burden of the applications.

An important contribution of the $\varepsilon$-hBOA is that while solving many-objective problems, the algorithm is explicitly building a joint probabilistic density function (pdf) model of what makes observation decisions likely to be non-dominated with respect to many-objectives, simultaneously. The joint pdf provides the potential for discovering and visualizing the hierarchical dependency structure of environmental observation problems. Kollat et al. (2008a) used a many-objective groundwater monitoring application to show that the $\varepsilon$-hBOA can dramatically enhance our approximately evaluation of Pareto efficiency for increasingly larger networks. An understanding of the existing dependencies on the observation network design problems gives the user valuable insights about the problem, its complexities and the inherent structure. A major benefit of the $\varepsilon$-hBOA on such applications is that while solving problems it provides users with an interesting hypotheses on the controls and dependencies impacting their systems performance across many-objectives, simultaneously.

Efficient methods of exploiting and presenting the information captured in the $\varepsilon$-hBOA’s Bayesian network models has significant potential to enhance the formulation and solution of environmental monitoring problems. This thesis highlights one of the ways to visually interpret the interdependency structure identified by the algorithm. It presents a visualization framework using Python packages and Graphviz to provide an illustration of the Bayesian networks generated by $\varepsilon$-hBOA. The visualization framework effectively displays the probabilistic linkages based on the user defined filtering criteria (the possible criterions being conditional probabilities and the order of hierarchies). The rules generated by the $\varepsilon$-hBOA are tested against the Pareto set for the problem to demonstrate the quality of the Bayesian network models. The results indicate the effectiveness of the models with close to 90% of the rules showing less than 20% deviation from their actual values on the Pareto set. The probabilistic linkages can be further analyzed to create simple heuristics for the problem. Though some of the rules identified by the algorithm
may be intuitive and intellectually trivial, they represent problem knowledge that traditional solution tools are incapable of capturing, thus enabling the $\varepsilon$-hBOA to deal with severely interdependent problems (a form of severe nonlinearity termed epistasis).

The visualization tool and the corresponding verification procedure represent a significant contribution towards ambitions in the fields of machine learning, optimization, and artificial intelligence that date back to the seminal work of Simon (1968). The discovery of approximate hierarchical dependency structures for our observation decisions within complex environmental systems is a methodological advance that has significant implications. These rules themselves provide the potential for advancing our observational hypotheses while simultaneously helping to overcome the computational barriers limiting our use of Pareto efficiency as an integrated measure of the value of information.

This thesis focuses on the end of the run analysis of the hierarchical Bayesian network. It provides a detailed study of the interdependency structures for the problem and its implications. However the study of the dynamic evolution of the hierarchical Bayesian network can provide valuable insights about the problem and the algorithm. How early a probabilistic linkage is identified in an evolution process and how it grows across the generations can give rewarding insights about the dynamic performance of the algorithm. This can be the focus for the future work in the study. One drawback of the algorithm is the time-consuming process of generating the Bayesian network. The future work can also focus on the development of a parallel extension of the algorithm that is capable of running on massively parallel computing resources to address the computational scaling issues of the algorithm.
APPENDIX A

RESULTS FOR THE MAXIMUM RATIO REPAIR APPROACH

Table A.1 shows the percentage contribution of the three algorithms towards the generation of the new reference sets for the MOd-KP correlated and uncorrelated instances that did not have pre-existing benchmark solution sets. The values indicate that $\varepsilon$-hBOA contributed the most nondominated solutions for the larger problem instances (in terms of objective count and the number of decision variables). The percentage contributions of $\varepsilon$-hBOA varies from 84% to 100% while the contributions of $\varepsilon$-NSGAII and SPEA2 were limited to 0% - 15% and 0% - 0.28%, respectively. For the small correlated instances of the 2 knapsack test case containing 100 items and 250 items, $\varepsilon$-NSGAII is the major contributor with 74% and 70% solutions respectively. The entire reference sets for the 3 knapsack 750 items, 4 knapsack 500 items and 750 items problems were contributed by the $\varepsilon$-hBOA.

Figure A.1 visually illustrates the best known approximation set for the 4 knapsack 250 item problem instance. Figure A.1 shows the predominant contribution of the $\varepsilon$-hBOA (indicated by the blue points) to the profits for knapsack 1, knapsack 2 and knapsack 3 plotted on the X, Y and Z axes, respectively. The fourth objective is represented by the orientation of the solution cones. Cones with an upward orientation represent high profits on knapsack 4 and conversely the cones with a downward orientation represent low profits. The color of the cones reflects
Table A.1. Percentage contributions of the $\varepsilon$-hBOA, the $\varepsilon$-NSGAII and the SPEA2 towards the generation of reference sets. Uncorrelated instance reference sets for 2 Knapsack 100 items, 250 items, 500 items and 3 Knapsack 100 items were made available by Zitzler and Thiele.

<table>
<thead>
<tr>
<th>Items</th>
<th>Uncorrelated Instances</th>
<th>Correlated Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\varepsilon$-hBOA</td>
<td>$\varepsilon$-NSGAII</td>
</tr>
<tr>
<td>100</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>250</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>500</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>750</td>
<td>84.20</td>
<td>15.80</td>
</tr>
<tr>
<td>2 Knapsacks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>250</td>
<td>95.84</td>
<td>4.16</td>
</tr>
<tr>
<td>500</td>
<td>98.17</td>
<td>1.81</td>
</tr>
<tr>
<td>750</td>
<td>98.98</td>
<td>0.90</td>
</tr>
<tr>
<td>3 Knapsacks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>90.47</td>
<td>9.24</td>
</tr>
<tr>
<td>250</td>
<td>92.87</td>
<td>7.13</td>
</tr>
<tr>
<td>500</td>
<td>99.66</td>
<td>0.34</td>
</tr>
<tr>
<td>750</td>
<td>100.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

which algorithm contributed each particular solution (blue represents $\varepsilon$-hBOA and red represents $\varepsilon$-NSGAII). SPEA2 did not contribute any solution for this problem instance. The high density region of the blue points illustrates the superior performance of the $\varepsilon$-hBOA at finding the reference set (92.87%). An interesting aspect of the contributions of the $\varepsilon$-NSGAII and the SPEA2 is that they contribute solutions that are at the farthest extremes of the objective space. This trend is observed across all of the MOd-KP instances.

Figure A.2 shows the nondominated front generated across 25 seeds for $\varepsilon$-hBOA, $\varepsilon$-NSGAII and SPEA2 on 2 objective test instances (Please note the objectives are to be maximized). It can be seen that for lower instance (250 items) the fronts of all the three algorithms overlap indicating similar efficiencies. However the distance between the fronts does increase on the instances with more decision variables (500 items and 750 items). The $\varepsilon$-hBOA generates more unique solutions and tends to move further away in the objective space. The front generated by $\varepsilon$-hBOA dominates a major portion of the fronts generated by $\varepsilon$-NSGAII and SPEA2. The
Figure A.1. Reference set for the uncorrelated instance of the 4 knapsack test case containing 250 items. Blue cones show the solutions contributed by $\varepsilon$-hBOA and red cones show the solutions contributed by $\varepsilon$-NSGAII. SPEA2 did not contribute any solutions towards the generation of this reference set.

Domination is even more pronounced in the higher objectives. Figure A.3 shows detailed views of the nondominated fronts generated across 25 random seed trials for each of the algorithms ($\varepsilon$-hBOA in blue, $\varepsilon$-NSGAII in green and SPEA2 in red) on the correlated problems instance of 3 knapsack 750 items. The plot clearly indicates the differences in the performances of the three algorithms. It can be seen that the $\varepsilon$-NSGAII and the SPEA2 fail to converge relative to the $\varepsilon$-hBOA and the front generated by the $\varepsilon$-hBOA dominates a major portion of the fronts generated by the $\varepsilon$-NSGAII and SPEA2. The $\varepsilon$-hBOA generates more unique solutions and tends to push toward more favorable regions of the objective space which explains its higher percentages in capturing the reference set. However the $\varepsilon$-hBOA fails to explore the extremes of the objective space where there is a low solution density in the decision space. The SPEA2 and the $\varepsilon$-NSGAII explore the extremes of the reference sets much more effectively.
Figure A.2. Nondominated fronts generated by the $\varepsilon$-hBOA, the $\varepsilon$-NSGAII and the SPEA2 for the correlated problem instance of 2 knapsack test cases (250,500,750 items).

It should be noted that this behavior is expected since there are far fewer combinations of solutions for the $\varepsilon$-hBOA to model with its Bayesian network in the extreme regions of the MOd-KP instances analyzed. This behavior was also exhibited in Kollat et al. (2008a). Bayesian network building is based on conditional probabilities and generates the solutions which have the higher joint probability distribution of being in a nondominated region. Given the problem constraints, there are significantly more options to model when selecting 200 items out of 500 items than when selecting 400 items. This combinatorial difference affects the spread of the solutions captured by the $\varepsilon$-hBOA. It concentrates on the highest solution densities in the MOd-KP’s decision space and makes it difficult to locate solutions in sparse regions. Thus, it is biased towards the center compromise regions of the MOd-KP’s tradeoffs. However, the $\varepsilon$-hBOA shows statistical superiority on two of the largest uncorrelated problem instances (4 knapsack 500 items, 750 items) for both the $\varepsilon$-indicator and hypervolume metrics.

Tables A.2 and A.3 show the end-of-run performance values attained by each algorithm for the unary $\varepsilon$-indicator-additive and hypervolume metrics. Readers are reminded that hypervolume metric measures the volume between the approximation set and the best known reference set with respect to a nadir point. Values
in the table show the average metric value for 25 random seed trials and standard deviations are included in the parentheses. In addition, a Kruskal-Wallis nonparametric statistical test was used to test the statistical significance of performance differences at the 95% confidence level for the metric distributions attained by each algorithm. The results of the statistical tests were used to rank the relative performances of each algorithm.

For the uncorrelated instances (see Table A.2) the $\varepsilon$-hBOA seems to be the worst performer with respect to the $\varepsilon$-indicator metrics for most of the test instances. For the two and three objective problems, the SPEA2 has superior $\varepsilon$-indicator performance followed by the $\varepsilon$-NSGAII and the $\varepsilon$-hBOA. It is interesting to note that the hypervolume metric shows the $\varepsilon$-NSGAII to be the best performing algorithm across most of the test instances. The SPEA2’s nearest neighbor den-
Table A.2. Performance of the \(\varepsilon\)-hBOA, the \(\varepsilon\)-NSGAII and SPEA2 with respect to the \(\varepsilon\)-indicator and hypervolume metrics for the uncorrelated problem instances. Highlighted values designate the statistically significant (to the 95-percent confidence level) best performing algorithm for the problem instance.

<table>
<thead>
<tr>
<th>Items</th>
<th>(\varepsilon)-hBOA</th>
<th>(\varepsilon)-NSGAII</th>
<th>SPEA2</th>
<th>(\varepsilon)-hBOA</th>
<th>(\varepsilon)-NSGAII</th>
<th>SPEA2</th>
<th>Multiplication factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mul/g415plica/g415on 100</td>
<td>31.19 (12.29)</td>
<td>17.6 (5.57)</td>
<td>30.82 (14.2)</td>
<td>1.67 (0.66)</td>
<td>0.78 (0.26)</td>
<td>1.52 (0.52)</td>
<td>1.00E+04</td>
</tr>
<tr>
<td>250</td>
<td>283.4 (51.72)</td>
<td>188.76 (29.21)</td>
<td>166 (35.85)</td>
<td>3.8 (0.6)</td>
<td>2.7 (0.35)</td>
<td>3.31 (0.54)</td>
<td>1.00E+06</td>
</tr>
<tr>
<td>500</td>
<td>565.56 (56.5)</td>
<td>491.24 (53.58)</td>
<td>470.88 (58.77)</td>
<td>1.81 (0.14)</td>
<td>1.45 (0.14)</td>
<td>1.9 (0.14)</td>
<td>1.00E+07</td>
</tr>
<tr>
<td>750</td>
<td>511 (91.58)</td>
<td>246.92 (73.71)</td>
<td>164.56 (92.97)</td>
<td>2.38 (0.41)</td>
<td>0.92 (0.28)</td>
<td>1.81 (0.35)</td>
<td>1.00E+07</td>
</tr>
<tr>
<td>100</td>
<td>62.95 (18.44)</td>
<td>73.44 (15.44)</td>
<td>75.71 (16.07)</td>
<td>2.7 (0.46)</td>
<td>2.4 (0.31)</td>
<td>5.01 (2)</td>
<td>1.00E+08</td>
</tr>
<tr>
<td>250</td>
<td>242.05 (47.36)</td>
<td>177.08 (47.19)</td>
<td>143.76 (25.85)</td>
<td>3.93 (0.66)</td>
<td>2.62 (0.82)</td>
<td>2.88 (0.45)</td>
<td>1.00E+10</td>
</tr>
<tr>
<td>500</td>
<td>451.96 (79.43)</td>
<td>362.76 (87.94)</td>
<td>330.8 (67.61)</td>
<td>3.2 (0.49)</td>
<td>2.12 (0.58)</td>
<td>2.58 (0.29)</td>
<td>1.00E+11</td>
</tr>
<tr>
<td>750</td>
<td>615.14 (138.26)</td>
<td>517.96 (76.79)</td>
<td>441.44 (58.23)</td>
<td>9.91 (2)</td>
<td>7.27 (1.41)</td>
<td>7.97 (0.91)</td>
<td>1.00E+11</td>
</tr>
<tr>
<td>100</td>
<td>81.44 (18.05)</td>
<td>80.92 (8.96)</td>
<td>82.8 (4.52)</td>
<td>4.59 (1.37)</td>
<td>4.88 (0.78)</td>
<td>16.26 (1.92)</td>
<td>1.00E+12</td>
</tr>
<tr>
<td>250</td>
<td>278.04 (41.34)</td>
<td>203.84 (34.36)</td>
<td>246.92 (28.8)</td>
<td>4.52 (0.66)</td>
<td>2.45 (0.72)</td>
<td>7.08 (0.72)</td>
<td>1.00E+14</td>
</tr>
<tr>
<td>500</td>
<td>536.52 (82.8)</td>
<td>555.08 (39.61)</td>
<td>569.13 (58.06)</td>
<td>6.38 (1.02)</td>
<td>7.42 (0.79)</td>
<td>13.04 (1.02)</td>
<td>1.00E+14</td>
</tr>
<tr>
<td>750</td>
<td>398.04 (65.27)</td>
<td>917.4 (55.94)</td>
<td>703.16 (34.1)</td>
<td>1.78 (0.5)</td>
<td>3.24 (0.48)</td>
<td>4.32 (0.38)</td>
<td>1.00E+16</td>
</tr>
</tbody>
</table>

Density estimation technique helps it preserve boundary solutions and thereby results in better \(\varepsilon\)-indicator metric values. The \(\varepsilon\)-NSGAII’s dynamic population sizing generates diverse approximation sets within close proximity of the reference trade-offs yielding its statistically superior performance with respect to the hypervolume metric. The poor performance of the highest contributor to the reference set (\(\varepsilon\)-hBOA) on both of the metrics indicates that the geometric distribution and spread of the approximation set has a significant impact on the performance measures.

This thesis also tests a new correlated instances of the MOd-KP. Correlated instances add an inherent structure to the MOd-KP and provides a difficult suite of combinatorial problems to benchmark algorithm performances. For the correlated instances shown in Table A.3, the \(\varepsilon\)-hBOA is shown to be the statistically superior performing algorithm across most of the MOd-KP instances tested. The superior performance of the \(\varepsilon\)-hBOA indicates the presence of increased difficulty in the problem and implies a strongly inter related decision structure. The \(\varepsilon\)-NSGAII is the second best performing algorithm and the SPEA2 is the worst performing algorithm. There are a few cases for the hypervolume metric where the \(\varepsilon\)-NSGAII and the \(\varepsilon\)-hBOA have statistically similar performance (and thus...
Table A.3. Performance of the $\varepsilon$-hBOA, the $\varepsilon$-NSGAII and the SPEA2 with respect to the $\varepsilon$-indicator and the hypervolume metrics for the correlated problem instances. Highlighted values designate the statistically significant (to the 95-percent confidence level) best performing algorithm for the problem instance.

| Items | Epsilon Indicator - Additive | | | | | | | | | | | | | | | | | | | | | | | |
|-------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|       | $\varepsilon$ - hBOA | $\varepsilon$ - NSGAII | SPEA2 | $\varepsilon$ - hBOA | $\varepsilon$ - NSGAII | SPEA2 | Multiplication factor |
| 100   | 16.44 (2.36) | 7.68 (2.82) | 8.56 (2.9) | 6.97 (1.6) | 2.53 (1.35) | 5.33 (1.58) | 1.00E+04 |
| 250   | 39.22 (12.51) | 22.92 (12.52) | 30.4 (19.09) | 3.93 (1.08) | 2.11 (0.99) | 5.18 (0.68) | 1.00E+05 |
| 500   | 33.92 (11.03) | 36.64 (8.08) | 82.56 (17.27) | 6.98 (2.29) | 8.4 (2.88) | 23.93 (3.62) | 1.00E+05 |
| 750   | 103.8 (22.52) | 66.88 (23.65) | 176.52 (33.44) | 3.75 (0.82) | 2.03 (0.67) | 6.55 (1) | 1.00E+06 |
| 100   | 18.65 (2.82) | 21.32 (4.1) | 22.48 (2.83) | 2.08 (0.7) | 2.15 (0.78) | 2.56 (0.61) | 1.00E+08 |
| 250   | 60.52 (13.96) | 52.72 (9.89) | 54.32 (8.85) | 5.83 (0.83) | 5.84 (0.82) | 6.3 (0.99) | 1.00E+09 |
| 500   | 85.2 (20.11) | 108.24 (15.34) | 134.36 (17.11) | 3.76 (1.08) | 4.13 (0.65) | 6.55 (0.61) | 1.00E+10 |
| 750   | 108.96 (33.06) | 206.2 (17.74) | 226 (22.06) | 9.65 (3.13) | 20.81 (1.86) | 27.37 (1.35) | 1.00E+10 |
| 100   | 25.84 (2.67) | 35.04 (3.75) | 40.2 (1.89) | 1.14 (0.28) | 1.17 (0.33) | 2.45 (0.32) | 1.00E+12 |
| 250   | 67.83 (16.93) | 76.5 (5.23) | 88.28 (17.9) | 0.55 (0.12) | 0.51 (0.11) | 1.28 (0.09) | 1.00E+14 |
| 500   | 76.96 (13.2) | 196.48 (15.63) | 202.72 (10.98) | 2.17 (1.97) | 11.7 (1.67) | 20.3 (1.6) | 1.00E+14 |
| 750   | 128.32 (22.64) | 331.04 (18.13) | 324.2 (18.27) | 2.56 (0.75) | 9.64 (0.77) | 11.23 (0.69) | 1.00E+15 |

have not been highlighted). Overall the superior performance of the $\varepsilon$-hBOA on correlated instances demonstrates the PMBGA’s ability to exploit problem structure and generate promising solutions.

Figure A.4 shows $\varepsilon$-indicator success rate plots for each algorithm on each test instance. The success rate plots show the cumulative distribution functions that define the percentage of random seed trials for a given algorithm that were able to meet or exceed a threshold value of the $\varepsilon$-indicator metric as a function of the number of function evaluations (NFE) utilized. The success rate plots provide valuable insights into the probabilistic search dynamics of the algorithms. The $\varepsilon$-indicator threshold values were selected at 85% quantile values from the random trial distributions of the best performing algorithm for each MOd-KP instance. The slope or steepness of each distribution is a measure of the reliability of the algorithm for a given level of computational effort. A vertical distribution signifies a highly reliable performance where all of the algorithm’s random seed trials exceeded the threshold performance with a minimum variance on computational effort (i.e., nearly the same NFE in all trials). Similarly, small slopes or horizontal distributions signify low or zero reliability, respectively. Figure A.4A represents the success plots of 12 instances of the uncorrelated problem and Figure A.4B...
Figure A.4. Success plots for the 24 problem instances. The plots indicate the probability of each algorithm to cross the 85% quantile on ε-Indicator additive metrics for each problem instance.
represents the success plots of the 12 instances of the correlated problem. Each row (moving left to right) indicates an increase in the number of objectives (2, 3, 4) for a set number of decision variables. Columns of the sub-plots (moving from the top to the bottom) represent an increase in the number of decision variables (100, 250, 500, 750) for a given number of objectives.

For the small instances of the uncorrelated problem (2 knapsacks 100 items, 250 items, 500 items, 750 items) all of the algorithms seem to reliably generate approximate sets close to the best performing algorithm. For the three objective problems, the SPEA2 has a slight decrease in reliability as compared to the $\varepsilon$-NSGAII and the $\varepsilon$-hBOA. The $\varepsilon$-NSGAII and the $\varepsilon$-hBOA achieve the specified threshold within 200,000 function evaluations. The differences in performance are more apparent in four objective problems with the $\varepsilon$-NSGAII and the SPEA2 showing significant decreases in the efficiency and effectiveness of their search. The $\varepsilon$-hBOA shows robust performance by achieving the specified threshold within 200,000 function evaluations across almost all of the test instances. It is interesting to note that even when the $\varepsilon$-hBOA is not the best performing algorithm, it still gets reliably close to the best performing algorithm at a fairly early stage of evolution.

The correlated instances of the problem show significant differences in the performances of the three algorithms. For the two objective problems, the $\varepsilon$-NSGAII and the $\varepsilon$-hBOA are fairly reliable at achieving the specified $\varepsilon$-indicator thresholds within 100,000 function evaluations whereas the SPEA2 takes around 400,000 function evaluations. For the three objective problems, the $\varepsilon$-hBOA achieves the threshold fastest followed by the $\varepsilon$-NSGAII and then the SPEA2. The four objective problem highlights the dominance of the $\varepsilon$-hBOA. The $\varepsilon$-hBOA achieves the threshold the fastest whereas the other algorithms seem to struggle to achieve the threshold at all. For the 4 knapsack 500 items test case, the probability of the $\varepsilon$-NSGAII and the SPEA2 attaining the 85% quantile thresholds is 0.4 and 0.1, respectively. Success rate plots indicate that $\varepsilon$-NSGAII and SPEA2 are not reliable and tend to fail completely on the larger instances. On the other hand the $\varepsilon$-hBOA is observed to achieve the success thresholds fairly consistently independent of problem size. This implies that the success of the $\varepsilon$-hBOA is influenced by the quality of its models and the amount of statistical information available and less so by the MOd-KP’s size and its constraints. This can be an important advan-
tage while solving large application problems. The $\varepsilon$-hBOA’s reliability may also lead to the reduction or complete elimination of the random seed trials yielding a dramatic reduction in the computational burden of solving MOd-KP problems.

The success rate plots indicate that the mean of the end of the run performance measure can be a poor indicator of the actual performance of the algorithm. The end of run metric values in Table 3.5 show the SPEA2 to be statistically superior and the $\varepsilon$-hBOA to be the worst performing algorithm on the $\varepsilon$-indicator metrics. However the success plots indicate a relatively poor reliability of the SPEA2 with respect to the $\varepsilon$-hBOA at achieving our success thresholds.

The success plots also give insights on how increasing the number of objectives for a given number of decision variables or vice-versa impacts search. As the number of objectives increases (moving left to right in subplots) the time taken for each algorithm to achieve the threshold increases. The the $\varepsilon$-hBOA shows only a slight increase in the required number of the function evaluations but the $\varepsilon$-NSGAII and the SPEA2 show substantial increases in the number of function evaluations required. The reliability of the $\varepsilon$-hBOA stays fairly constant as it achieves the threshold at almost the same NFE across all of its random seed trial runs. The runtime variance across the random seeds for the SPEA2 and the $\varepsilon$-NSGAII increases significantly as the number of objectives increases. Similar trends are observed as the number of decision variables are increased (moving from the top to the bottom) for a given number of objectives. This comparison indicates that the $\varepsilon$-hBOA is fairly robust in its performance independent of the number of objectives or decision variables. The performance of the $\varepsilon$-NSGAII and the SPEA2 are critically dependent on the size of the problem instance. They are expected to perform fairly well for small instances of the problem (2 knapsack 100 items, 250 items) and struggle to maintain their performance for larger instances (4 knapsack 500 items, 750 items). Moreover, the introduction of correlations (or linkages) for successively larger MOd-KP instances causes the $\varepsilon$-NSGAII and the SPEA2 to completely fail relative to the $\varepsilon$-hBOA.


Tang, Y., Reed, P., Wagener, T., 2006. How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration? Hydrology and earth system sciences 10 (2).


