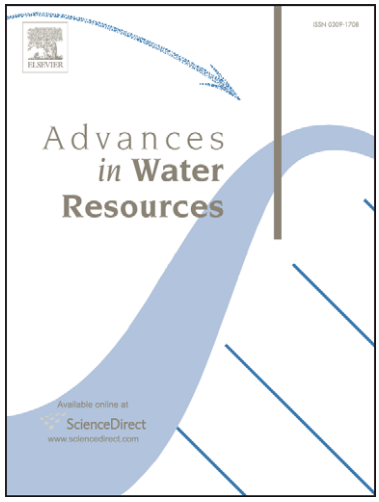

A New Epsilon-Dominance Hierarchical Bayesian Optimization Algorithm for Large Multi-Objective Monitoring Network Design Problems

J.B. Kollat, P.M. Reed, J.R. Kasprzyk

PII: S0309-1708(08)00029-8
DOI: 10.1016/j.advwatres.2008.01.017
Reference: ADWR 1246

To appear in: *Advances in Water Resources*

Received Date: 17 October 2007
Revised Date: 29 January 2008
Accepted Date: 31 January 2008

Please cite this article as: Kollat, J.B., Reed, P.M., Kasprzyk, J.R., A New Epsilon-Dominance Hierarchical Bayesian Optimization Algorithm for Large Multi-Objective Monitoring Network Design Problems, *Advances in Water Resources* (2008), doi: 10.1016/j.advwatres.2008.01.017

A New Epsilon-Dominance Hierarchical Bayesian Optimization Algorithm for Large Multi-Objective Monitoring Network Design Problems

J.B. Kollat, P.M. Reed *, J.R. Kasprzyk

Department of Civil and Environmental Engineering, The Pennsylvania State University, 212 Sackett Building, University Park, PA 16802-1408

Abstract

This study focuses on the development of a next generation Multi-Objective Evolutionary Algorithm (MOEA) that can learn and exploit complex interdependencies and/or correlations between decision variables in monitoring design applications to provide more robust performance for large problems (defined in terms of both the number of objectives and decision variables). The proposed MOEA is termed the Epsilon-Dominance Hierarchical Bayesian Optimization Algorithm (ε-hBOA), which is representative of a new class of probabilistic model building evolutionary algorithms. The ε-hBOA has been tested relative to a top performing traditional MOEA, the Epsilon-Dominance Nondominated Sorted Genetic Algorithm II (ε-NSGAII) for solving a four-objective LTM design problem. A comprehensive performance assessment of the ε-NSGAII and various configurations of the ε-hBOA have been performed for both a 25 well LTM design test case (representing a relatively small problem with over 33-million possible designs), and a 58 point LTM design test case (with over 2.88×10^{17} possible designs). The results from this comparison indicate that the model building capability of the ε-hBOA greatly enhances its performance relative to the ε-NSGAII, especially for large monitoring design problems. This work also indicates that decision variable interdependencies appear to have a significant impact on the overall mathematical difficulty of the monitoring network design problem.

Key words: Long-term groundwater monitoring, evolutionary algorithms, multi-objective optimization, Bayesian networks, Probabilistic models

* Corresponding author.

Email addresses: juk124@psu.edu (J.B. Kollat), preed@engr.psu.edu (P.M. Reed), jrk301@psu.edu (J.R. Kasprzyk).

1 Introduction

This study contributes a new Multi-Objective Evolutionary Algorithm (MOEA) termed the Epsilon-Dominance Hierarchical Bayesian Optimization Algorithm (ε-hBOA), which has been developed to solve large, long-term groundwater monitoring (LTM) design problems (defined in terms of both the number of design objectives and decision variables). LTM networks use spatially distributed wells to characterize groundwater contamination over long time scales (ranging from years to decades) and ensure that the contamination does not pose an unacceptable risk to humans and the environment [1]. Management strategies for the health risks associated with contaminated groundwater must also consider the long-term economic costs associated with site monitoring. Federal expenditures on LTM for the years 2000 through 2010 are estimated at 5-billion US dollars [1], motivating the need for cost effective LTM design strategies that are protective of human and ecological systems. However, designing LTM networks for contaminated groundwater is a challenging problem that has long been recognized to suffer from a "curse of dimensionality" [2]. This is largely due to their discrete decision spaces that grow exponentially as the number of contaminant measurements, their locations, and sampling rates are considered.

For example, the decision space (or total number of possible designs) of a LTM network design problem grows according to the equation 2^{snt} where s represents the number of potential observations at a sampling location, n is the number of possible sampling locations, and t is the number of sampling periods. The 2 in this equation assumes a binary decision where there are two options, sample (1) or do not sample (0). If there are 25 potential sampling locations ($n = 25$) to characterize a single contaminant ($s = 1$) at a snapshot in time ($t = 1$), this particular network design problem contains 2^{25} (or over 33-million) possible designs. Adding just one additional sampling period results in a significant increase in the size of the decision space (over 1.12×10^{15} possible designs for $n = 25$, $s = 1$ and $t = 2$).

The exponential scaling of the LTM problem as well as its discrete decisions have motivated several researchers to utilize evolutionary algorithms (EAs) in its solution since the early 1990s [3–9]. EAs evolve populations of designs toward near optimal sampling schemes using processes that are analogous to selection, mating, and mutation. While previous LTM design applications using EAs have generally been deemed successful in their attempts to generate approximately optimal solutions, they have typically either been applied to problems of limited size and complexity or limited in their exploration of designs. In general, the LTM design problem is inherently suited to formulations that take into account multiple design objectives simultaneously (e.g., cost, uncertainty, reliability, etc. [2]). The use of multiple design objectives typically

results in tradeoffs where the performance in one objective cannot be improved without degrading the performance in another objective [10]. The goal of multiobjective search is to identify optimal solutions that compose tradeoff surfaces of maximum dimension $M-1$ given M design objectives. For the past 20 years, a variety of MOEAs have been developed and shown to be capable of optimizing highly nonlinear, discrete, and non-convex objective space landscapes without differentiation [11–13]. In addition, MOEAs' population-based search enables them to evolve approximations for entire tradeoff (or Pareto [14]) surfaces within a single optimization run.

Building on prior multiobjective LTM design applications [1, 3, 4, 7, 15–25], recent studies [7, 26] have shown that MOEAs can be combined with visualization tools to solve LTM problems with three or more objectives (termed high-order Pareto optimization problems). Multiobjective formulations add to the complexity of LTM design problems by increasing the number of solutions that must be evolved. It has also been shown that the required population size for an MOEA to maintain a diverse representation of the LTM design tradeoffs is directly related to the complexity of the objective space, and that adding objectives generally increases this complexity [27, 28]. Moreover, functional interdependencies between decision variables also increases the difficulty and computational search requirements for optimization problems [29]. In short, this work posits that a new class of combinatorial search algorithm is necessary to resolve the LTM problem's potential for large numbers of design objectives and sampling decisions that can have complex spatial and/or temporal interdependencies depending on the contaminant plume.

Kollat and Reed [30] recently showed that as the number of LTM sampling decisions increases linearly, the computational complexity of using the Epsilon-Dominance Nondominated Sorted Genetic Algorithm II (ε-NSGAII) grows quadratically. Computational scaling defines how problem size (i.e., number of wells, contaminants sampled, sampling times) increases the average number of design evaluations required by an MOEA to evolve a high quality solution for a given LTM design problem. The ε-NSGAII has been shown previously to be highly robust at solving LTM design problems relative to other state-of-the-art MOEAs [9]. Kollat and Reed [32] conclude that traditional MOEAs such as the ε-NSGAII are limited to increasingly lower quality approximations for large LTM problems given their large solution set sizes and the limits posed by their quadratic computational complexities. This study builds on Kollat and Reed [30] by introducing a new MOEA termed the Epsilon Dominance Hierarchical Bayesian Optimization Algorithm (ε-hBOA), which uses Bayesian networks (a form of probabilistic model building) to optimize large LTM design problems. This study seeks to determine if the incorporation of probabilistic modeling building improves the ability of an MOEA to solve LTM design applications, and if so, what is the best implementation of model building to maximize search success?

The remainder of this study proceeds as follows. Section 2 details the four-objective LTM test case problem formulation. Section 3 discusses the ε-NSGAII and the ε-hBOA, as well as the various configurations of the ε-hBOA tested in this study. Section 4 describes the computational experiment used to demonstrate the performance of the new ε-hBOA algorithm. Sections 5 and 6 present and discuss the results of the study. Section 7 concludes with our key findings and recommendations for future work.

2 Monitoring Case Studies

2.1 Test Cases

The LTM network design test case used in this study is based on a 50-million node flow and transport simulation representing the migration of a hypothetical perchloroethylene (PCE) plume originating from an underground storage tank [31]. Although the contamination is simulated, the hydrogeology of the site is real and has been extensively characterized [32–35]. For this test case, there are 29 sampling wells located throughout the contaminant plume, each with no more than three sampling ports available along its vertical axis. This results in a total of 58 sampling locations where PCE concentration data is available. Additional details on this test case can be found in Maxwell et al. [31].

In order to test how LTM problem size impacts the performance of a traditional MOEA versus that of the new ε-hBOA algorithm, two LTM network design test cases were developed - one small and one large. The small test case was developed by eliminating four wells from the 29 well test case for a total of 25 sampling wells. Each of the 25 well locations was then treated as a decision. This means that if a particular well is chosen for sampling and it has multiple sampling ports available, all sampling ports are utilized. The size of this test case was chosen in part because it can be enumerated (solved to completion) by evaluating all 2^{25} (or over 33.5-million) possible designs given current computational constraints. Knowing the true solution to the LTM test case ultimately provides the most rigorous means of assessing MOEA performance. In order to test MOEA performance for a large network design problem, the second larger test case treats each of the 58 sampling ports available at the 29 well locations as decision variables. This increases the size of the decision space to 2^{58} (or over 2.88×10^{17}) possible designs. Given that an LTM problem of this size cannot be enumerated, a "best known" reference set was developed by combining all of the solutions found by all algorithm runs. Hereafter, the small test case will be referred to as the 25 well test case, and the large test case will be referred to as the 58 point test case.

2.2 Design Objectives

Four design objectives were chosen for this study, each of which were minimized. The design objectives included: (1) the cost of sampling the contaminant plume, (2) the error relative to using all available data points to create spatial plume maps, (3) the spatial uncertainty associated with characterizing the plume, and (4) the error associated with estimating the mass of contaminant in the plume. Hereafter, each of the four design objectives are referred to as COST, ERROR, UNCERT, and MASS respectively. Equation 1 describes each of the four objectives as a function of the vector $\mathbf{x}_\kappa$ representing a particular sampling plan κ in the entire decision space Ω. Each sampling design vector is composed of the design decisions $x_{\kappa,i}$, ones and zeros indicating whether or not a sample is taken. Equation 1 is subject to the constraint that a particular sampling plan $\mathbf{x}_\kappa$ does not result in unestimated PCE concentrations, otherwise the design is penalized. The spatial interpolation scheme used to estimate PCE concentrations at unsampled locations throughout the domain, as well as the penalization scheme used for constraint violations is discussed later in this section.

Minimize:

$$\begin{aligned} &\mathbf{F}(\mathbf{x}_\kappa) = \Big(f_{\text{cost}}(\mathbf{x}_\kappa), f_{\text{error}}(\mathbf{x}_\kappa), f_{\text{uncert}}(\mathbf{x}_\kappa), f_{\text{mass}}(\mathbf{x}_\kappa)\Big), \quad \forall \kappa \in \mathbf{\Omega} \\ &\text{where } x_{\kappa,i} = \begin{cases} 1, & \text{if the ith well is sampled} \\ 0, & \text{otherwise} \end{cases}, \quad \forall \kappa, i \\ &\text{Subject to } U(\mathbf{x}_\kappa) = 0 \end{aligned} \tag{1}$$

COST. The cost of a particular sampling plan $\mathbf{x}_\kappa$ is described by Equation 2.

$$f_{\text{cost}}(\mathbf{x}_\kappa) = \sum_{i=1}^{n_{\text{samples}}} C_S(i) x_{\kappa,i} \tag{2}$$

The COST objective is simply the summation of the cost, $C_S(i)$, of each sample for a particular plan. In this study, normalized costs were used. For the 25 well test case, since all sampling ports are utilized when a well is sampled, $C_S(i)$ ranged from 1 to 3 depending on the well, while for the 58 point case, $C_S(i)$ was always 1. As a result, the range of the COST objective for the 25 well test case was 0 to 47 and for the 58 point test case was 0 to 58.

ERROR. The error associated with characterizing the plume is described by Equation 3.

$$f_{\text{error}}(\mathbf{x}_\kappa) = \sum_{j=1}^{n_{\text{estimates}}} \Big(c_{all}(\mathbf{u}_j) - c_\kappa(\mathbf{u}_j)\Big)^2 \tag{3}$$

Quantile Kriging (QK) described later in this section was used to obtain PCE concentration estimates throughout the sampling domain based on a grid containing points $\mathbf{u}_j$. The ERROR objective is quantified by summing the squares of the residuals between the Kriged PCE concentration map obtained by sampling from all locations, $c_{all}(\mathbf{u}_j)$, and the Kriged PCE concentration map, $c_\kappa(\mathbf{u}_j)$, obtained using sampling plan κ.

UNCERT. The uncertainty associated with characterizing the plume is described by Equation 4.

$$f_{\text{uncert}}(\mathbf{x}_\kappa) = \sum_{j=1}^{n_{\text{estimates}}} A_j \sigma(\mathbf{u}_j) \tag{4}$$

The UNCERT objective is quantified by summing the weighted standard deviations, $\sigma(\mathbf{u}_j)$, obtained from QK throughout the sampling domain. The weighting factor A_j can be used to assign importance to certain regions of the domain. For this study, all regions were assumed equally important. The weighting factor, A_j, used for all locations, $\mathbf{u}_j$, was based on the standard deviation of a uniform distribution, $2\sqrt{3}$.

MASS. The error in estimated dissolved mass of PCE contaminant as a result of characterizing the plume using QK is described by Equation 5.

$$f_{\text{mass}}(\mathbf{x}_\kappa) = \log_{10}\left(\left|\frac{\text{Mass}_{all} - \text{Mass}_\kappa}{\text{Mass}_{all}}\right| \cdot 100\%\right) \tag{5}$$

The MASS objective quantifies the relative error between the mass of PCE contamination estimated based on sampling from all locations, Mass_{all}, and the mass of PCE contamination estimated, Mass_κ, based on a particular sampling plan κ. Dissolved mass estimates were based on the concentration data provided by QK throughout the sampling domain. This objective is scaled logarithmically since its values can range over multiple orders of magnitude.

Spatial Interpolation. PCE concentration estimates were obtained at unsampled locations throughout the sampling domain using QK. Kriging provides a minimum error variance estimate of contaminant concentration at an unsampled location provided the data at the sampled locations [36,37]. Quantile Kriging was chosen in this study based on its effectiveness in providing high quality plume interpolations despite highly variable PCE concentrations and preferential sampling [38]. QK extends Ordinary Kriging by transforming contaminant concentrations to quantile space according to their rank ordering [39]. This is done to reduce the influence of extreme concentration values on the mean and variance of the data. A C version of KT3D, a three-dimensional Kriging library included as part of the GSLIB software package available in Fortran [36] was used to perform the Kriging required to quantify the ERROR, UNCERT, and MASS objectives.

Constraint Violations. A moving local neighborhood (i.e., a locally varying mean) was used to perform each spatial interpolation. If a particular sampling plan $\mathbf{x}_\kappa$ contained too few samples, or if the samples were poorly distributed in space, unestimated points result and the constraint of Equation 1 is violated resulting in a penalty. Penalizing the objective values of solutions that violate a constraint rather than eliminating them from consideration is a common practice in evolutionary optimization to ensure that these solutions are given the opportunity to evolve into better solutions. Solutions which result in a constraint violation may contain valuable information, making their continued contribution to the evolutionary process important.

$$\mathbf{F}_{\text{penalty}}(\mathbf{x}_\kappa) = \begin{cases} f_{\text{cost}}^{\text{penalty}} = f_{\text{cost}} + f_{\text{cost}}^{\max} \\ f_{\text{conc}}^{\text{penalty}} = f_{\text{conc}} + n_{\text{estimates}} + U(\mathbf{x}_\kappa) + f_{\text{cost}}^{\max} \\ f_{\text{uncert}}^{\text{penalty}} = f_{\text{uncert}} + n_{\text{estimates}} + U(\mathbf{x}_\kappa) + f_{\text{cost}}^{\max} \\ f_{\text{mass}}^{\text{penalty}} = f_{\text{mass}} + n_{\text{estimates}} + U(\mathbf{x}_\kappa) + f_{\text{cost}}^{\max} \end{cases} \tag{6}$$

Equation 6 penalizes the COST objective based on the maximum cost of sampling $f_{\text{cost}}^{\max}$, and penalizes the ERROR, UNCERT, and MASS objectives based on the number of QK estimates throughout the domain $n_{\text{estimates}}$, the number of unestimated points causing the constraint violation $U(\mathbf{x}_\kappa)$, and the maximum cost of sampling the system $f_{\text{cost}}^{\max}$. Taking into account the number of unestimated points causing the constraint violation provides a relative weighting to the penalized solutions (i.e., solutions with many unestimated points are penalized more heavily than those with few unestimated points).

3 Multiobjective Evolutionary Optimization

The goal of multiobjective optimization is to identify the Pareto-optimal tradeoffs between an application's objectives. These tradeoffs are composed of the set of solutions that are better than all other solutions in at least one objective and are termed non-dominated or Pareto-optimal solutions [14]. The Pareto-optimal front is obtained by plotting these solutions according to their objective values yielding maximally an $M-1$ dimensional surface where M is the total number of design objectives. MOEAs' population-based search enables them to evolve entire tradeoff (or Pareto [14]) surfaces within a single optimization run for problems with huge decision spaces. For additional resources on multiobjective optimization, the reader is encouraged to refer to the books by Deb [10] and Coello Coello [40] for comprehensive introductions to these topics.

3.1 The Epsilon-Nondominated Sorted Genetic Algorithm-II (ε-NSGAII)

The ε-NSGAII has been shown to perform very well relative to other state-of-the-art MOEAs at solving LTM network design problems [7,9,30]. In addition, the ε-NSGAII's performance has been validated extensively on a variety of test functions and applications [41–44]. The ε-NSGAII is based on the original NSGAII [45], which uses nondomination sorting and crowding distance to maintain solution diversity, simulated binary crossover (SBX) [46], polynomial mutation [10], and elitism [10]. The ε-NSGAII expands on the original NSGAII through the inclusion of dynamic population sizing [47] and ε-nondominance archiving [48, 49].

The ε-NSGAII evaluates potential LTM designs using Equation 1. Initially, a population of N random designs is generated and non-domination and crowding distance [45] are used to assign fitness values to each design based on their performance across the suite of objectives. SBX crossover and polynomial mutation are then performed to generate N *child* solutions created from the selected group of highly fit *parents*. The algorithm then combines N child solutions and N parent solutions into a temporary pool of $2N$ candidate solutions. Non-domination sorting is then used to rank each of the $2N$ designs based on the number of solutions that dominate them in all objectives. In addition, crowding distance is calculated based on the average Euclidean distance between a design and the remaining designs that have been assigned the same rank. At this point, a new population of N solutions is filled by gathering the highest ranked solutions. When solutions of equal rank exceed the size of the new population, two-step crowded binary tournament selection is performed to fill the available population slots while giving preference to solutions with higher crowding distance values (see "Front i" in Figure 2). Designs with higher crowding distance values add diversity to the population of designs, which helps to ensure that the ε-NSGAII will find solutions along the full extent of the Pareto surface. The N children that have been selected now become the parents of a subsequent generation from which the process is repeated. In addition these N children are eligible for entry into an offline archive that stores the best solutions found throughout the run. To achieve entry into the archive, child solutions must be ε-nondominated with respect to the currently stored solutions of the archive.

Two defining characteristics of the ε-NSGAII [9] are its ability to dynamically change its population size [47] based on search progress, and the inclusion of ε-nondominance archiving [48, 49] based on user specified precision requirements for each objective. Epsilon-nondominance archiving allows the user to control the computational cost of their application by eliminating the need to evolve unnecessarily precise solutions [9, 30, 48]. Depending on the preferences of the user and their willingness to accept an approximation for the full res-

olution Pareto set, ε-nondominance can greatly decrease the computational requirements of the algorithm as was recently shown by Kollat and Reed [30]. Dynamic population sizing [47] eliminates the need to specify the population size of the ε-NSGAII by using a series of *connected runs* where small populations are initially exploited to pre-condition search and subsequent populations are sized based on search progress. Dynamic population sizing allows the ε-NSGAII's population size to increase or decrease commensurate with problem difficulty. In addition, when the size of the ε-nondominance archive stabilizes, the ε-NSGAII's connected runs are equivalent to a diversity-based EA search enhancement recommended by Goldberg [50] termed *time continuation* where diverse search is promoted for as long as is necessary or feasible. The interested reader can refer to prior published work regarding these features of the ε-NSGAII [9, 30].

3.2 The Epsilon-Nondominated Hierarchical Bayesian Optimization Algorithm (ε-hBOA).

For the LTM design problem, decisions have complex interdependencies or correlations as a result of the underlying physics of flow-and-transport, the influence of domain geometry, and the underlying properties of the performance criteria considered. Recent literature in the field of evolutionary computation [50, 51] has shown that identifying these complex interdependencies and preserving them while generating new solutions is vital for EAs solving difficult engineering and science problems. The traditional MOEA operators of crossover and mutation assume that all of a problem's decisions are statistically independent, which may limit their performance for some challenging applications such as LTM design. A new class of EAs termed Probabilistic Model Building Genetic Algorithms (PMBGAs) "learn" the linkages between decision variables by building probabilistic models that express the interdependencies between decision variables to better preserve these links throughout the evolutionary process [51]. PMBGAs differ from traditional genetic algorithms by replacing the crossover and mutation operators with a probabilistic model [29]. Each generation, promising solutions are selected from the population and a probability distribution is estimated based on the conditional relationships of decision variables. New *child* solutions are then generated by sampling the estimated probability distribution. PMBGAs can be useful on any class of problem where decision variables are correlated (or linked) with one another. For example, in LTM design, the decision to sample from a certain location in space and time affects other sampling decisions because of the spatio-temporal structure of a contaminant plume. 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 decisions) and increasingly superior as problem difficulty increases (i.e.,

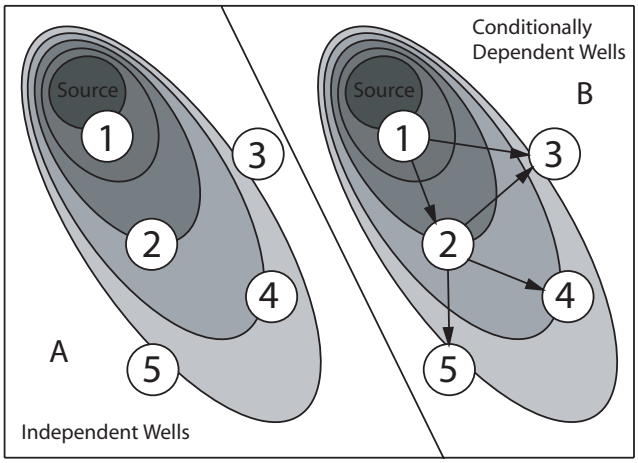

Fig. 1. Potential Bayesian network relationship of sampling wells for a contamination plume. Figure A demonstrates the case when all wells are sampled independently and Figure B illustrates the case where conditional probabilities may exist in sampling various wells.

increasing interdependencies between decisions) [51].

The Hierarchical Bayesian Optimization Algorithm (hBOA) developed by Pelikan [29] attempts to overcome the difficulties posed by large sets of interrelated decision variables by building Bayesian network models of the decision space. Bayesian networks [52] use directed acyclic graphs (defined as graphs with edges directed between vertices where there are no cycles which center on only one vertex [53]), to model sets of conditional probabilities between variables. The network contains a structure showing which variables are independent and a set of conditional probabilities for each variable [52]. Each decision variable of the problem is represented as a node in the Bayesian network. Edges, which represent the conditional probabilities between each of the decision variables, are used to establish relationships between the nodes. For example, if two decisions (nodes) are completely independent of one another, no edges connect these nodes. Figure 1 presents a hypothetical illustration of the potential for conditional dependencies that might exist between monitoring wells for a contamination plume in a typical LTM application. Figure 1A illustrates the case where the probability of sampling from each of the five potential monitoring points are completely independent of one another (i.e., there are no edges connecting the decisions). This represents the assumption made by the ε-NSGAII. Figure 1B illustrates the case when sampling certain wells may be dependent on whether or not a sample has been taken at another well. For example, the decision to sample from well "3" may depend on whether or not contamination exists at wells "1" or "2". While the dependency structure of network design problems has the potential to be quite complex, the identification of at least some of these dependencies may greatly aid in improving the search efficiency of MOEAs.

The Bayesian network model is built iteratively using a greedy search algorithm [54] that performs elementary network operations such as edge additions, removals, and reversals, intended to maximally improve the quality of the model. Changes in model quality can be measured using a variety of information theory based metrics [29], some of which include the Bayesian Dirichlet metric (BD) [54], the K2 Metric [55], or the Bayes Information Criterion (BIC) metric [56]. These metrics enforce Occam's Razor by penalizing complex models that do not significantly improve their predictive skill. Regardless of the chosen metric, network operations which result in the greatest metric score increase are the basis for building the Bayesian network model [29]. Once elementary network operations no longer exhibit significant improvements in the model, the construction of the network is assumed adequate and the Bayesian network model building is terminated. In each generation, the Bayesian network models are built from the binary strings that compose the current population. Models are rewarded when they find binary variable combinations within the population that are associated with highly ranked solutions.

In complex problems such as LTM, there may be interactions within highly correlated groups of decisions as well as across those groups (e.g., well clusters in the source area or sampling points along a plume's boundaries). The hBOA was selected in this study because of its ability to solve hierarchically difficult problems. 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 [57]. For example in the context of the LTM design test case used in this study, combinations of wells are important for defining major plume zones such as the source area or leading edge. Within each area, each well can define concentrations at up to three vertical locations, and it would seem reasonable that a bottom level of problem decomposability would correspond to individual sampling point decisions. The goal in developing a hierarchical solver such as the hBOA is to extend its capabilities to problem structures which exhibit strong multivariate links between single decisions as well as clusters of decisions. The hBOA learns proper hierarchical problem decomposition using a technique referred to as *chunking* within the Bayesian network model building process. Chunking is used within the model building process to allow groups of decision variables which are related at lower hierarchical levels to be clustered, and subsequently used to model higher order interactions [29]. Decision graphs are used to store the conditional probabilities of each decision variable. The decision graphs contain various nodes, all of which are permitted to have multiple parents (except for the root node). The performance of the hBOA has been demonstrated on both test functions [29, 58–60] and various real-world applications such as the notably difficult Ising spin-glass systems [61,62]. However, there are some known limitations to the hBOA. The Bayesian network model building is challenging and adds complexity to the algorithm. In addition, population size while seen as important within conventional EAs [63], is very important

to the hBOA since the quality of its probabilistic models is directly correlated with the size of the population sampled.

The hBOA algorithm is fundamentally different from a traditional MOEA like the ε-NSGAII because it eliminates the crossover and mutation operators, and replaces these with Bayesian network model building. Also, the original hBOA algorithm was developed to handle only a single design objective. Since the ε-NSGAII had been shown to be highly effective at solving LTM design problems in the past [7, 9, 30, 42], it was decided that the new ε-hBOA should represent a combination of the strengths of both the ε-NSGAII and the original hBOA. To accomplish this, the Bayesian network model building of hBOA was used to replace the SBX crossover and polynomial mutation operators of the ε-NSGAII. This results in a multiobjective algorithm with dynamic population sizing and ε-nondominance archiving options. The new ε-hBOA algorithm proceeds as follows. Initially, a population of N random designs is generated and the concept of Pareto-dominance is used to assign fitness values to each design based on its performance in terms of each design objective (see Figure 2). A Bayesian network is then iteratively generated based on this *parent* population of designs using the BIC metric as a termination criteria. *Child* solutions are then generated by sampling the resulting Bayesian network's modeled joint probability distribution for sampling designs. The algorithm then proceeds similarly to the ε-NSGAII where Pareto ranking and crowded binary tournament selection are used to fill a new population of N superior designs. The N *children* of the new population are eligible for inclusion in the ε-nondominated archive and become the parents of a subsequent generation from which the process is repeated until some termination criteria is met. Dynamic population sizing can also be utilized by the new ε-hBOA developed in this study, resulting in a series of "connected runs" each with a unique population size based on search progress (ε-nondominated archive size).

4 Computational Experiment

The ε-NSGAII and the ε-hBOA algorithms were tested on both the 25 well and 58 point LTM network design test cases described in Section 2.1 to determine the effects of Bayesian network model building on algorithm performance. Several configurations of the new ε-hBOA algorithm were tested to determine the important factors affecting algorithm performance.

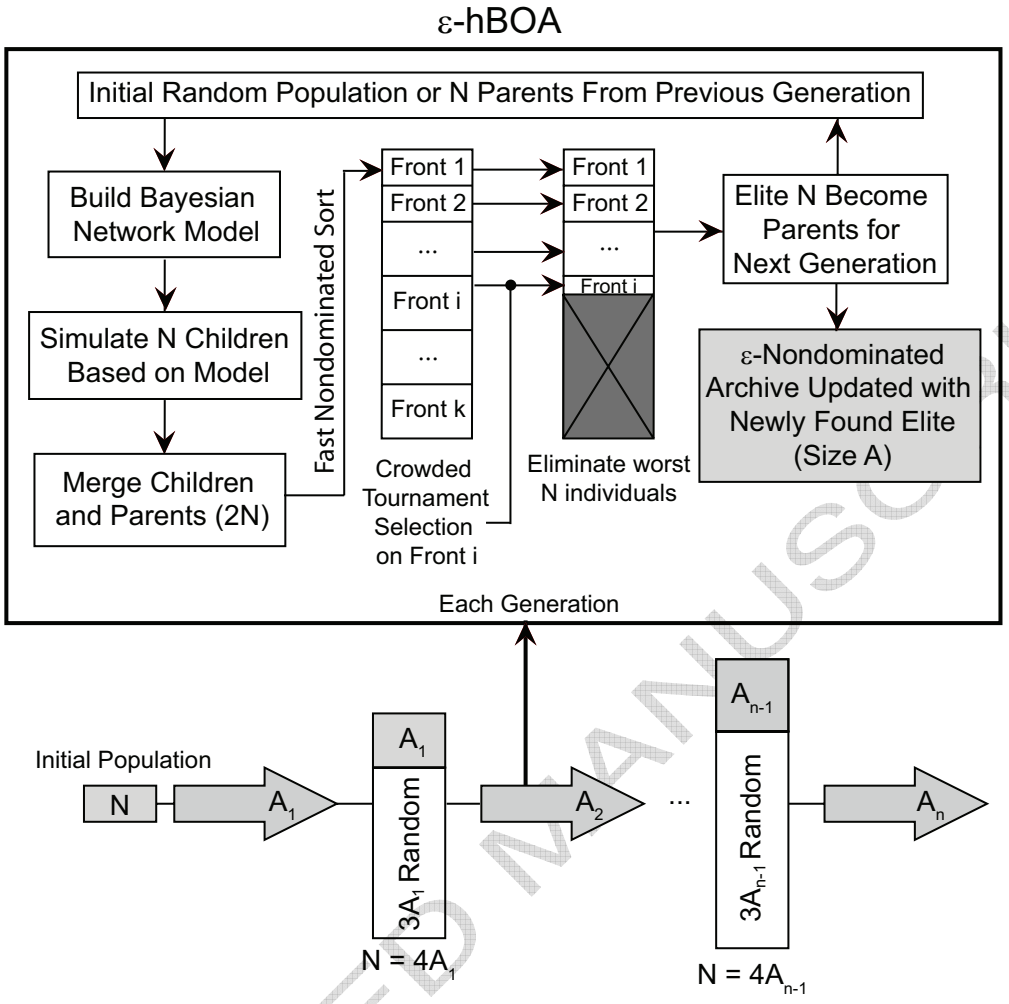

Fig. 2. Schematic diagram of the ε-hBOA illustrated using the notation of Deb et al. [45]. This figure shows the *connected runs* and dynamic population sizing concepts of the ε-NSGAII combined with Bayesian network model building and simulation (both features of the hBOA). In the figure, N represents population size and A represents ε-nondominance archive size.

4.1 Reference Set Generation

The true solution to the 25 well test case was fully enumerated, ultimately providing the most rigorous evaluation framework with regard to algorithm performance. The Pareto-optimal solution set for the 25 well test case is composed of 2,472 solutions and is shown in Figure 3A with the COST, ERROR, and UNCERT objectives plotted on the X, Y, and Z-axes respectively. Since this is a four objective problem, the color of the solution is used to represent the MASS objective where red solutions indicate high mass error and blue solutions indicate low mass error. Objectives values for this Pareto-set range from 8 to 47 in the COST objective, 0 to 43.7 in the ERROR objective, 1396 to 1672 in the UNCERT objective, and -7 to 1.72 in the MASS objective

(which is scaled logarithmically).

The 58 point test case is significantly more difficult and cannot be enumerated since the decision space is composed of over 2.9×10^{17} possible designs. However, a best known solution was generated by combining all solutions found by all trials of all algorithm configurations explored in this study. The union of all solution sets from all algorithm configurations was then reduced to a Pareto-approximation of the true Pareto set. The Pareto-approximation for the 58 point test case contains 22,333 solutions and is shown in Figure 3B, where again, the MASS objective is represented by the color of the solutions. Objectives for this Pareto-approximation range from 5 to 58 in the COST objective, 0 to 48.8 in the ERROR objective, 1376 to 1702 in the UNCERT objective, and -7 to 2.31 in the MASS objective. The most notable feature of the 58 point case is that it generally results in increased ranges of objective values as a result of being able to sample from individual locations along each well. Figure 3C shows a closer view of the central region of the Pareto-approximation that highlights the density of solutions. Here it can be seen that at each level of cost, a three objective tradeoff surface exists between ERROR, UNCERT, and MASS. Figure 3C shows that the reference set's solution density is greater for sampling schemes that yield larger numbers of combinations (e.g., selecting 57 of 58 total locations yields far fewer combinations than does selecting 30 locations as expected).

4.2 Performance Assessment of MOEAs

When evaluating the performance of MOEAs, metrics that measure the effectiveness, efficiency, and reliability of the algorithm are important. Effectiveness refers to the ultimate performance of the algorithm (i.e., how well did it solve the problem), and is evaluated by measuring end-of-run statistics for how well reference sets were captured. Measuring algorithm efficiency is important for the computational costs and overall search dynamics of the algorithm. Was it slow to get started initially? Did it reach its final solution quickly? Algorithm efficiency can be assessed using runtime performance metrics which record the progress of the algorithm throughout its entire run. Reliability provides a measure of the reproducibility of the results of the algorithm for different random seeds. Since MOEAs require random initial populations, the choice of this initial population may or may not influence the effectiveness and efficiency of the algorithm. Ideally, it is desired that the initial population have no bearing on algorithm performance. The reliability of the algorithm is assessed using random seed analysis where multiple random initializations (trials) are run.

Runtime convergence [64] is used in this study to measure the average Euclidean distance between an approximation set (i.e., the set of solutions found by

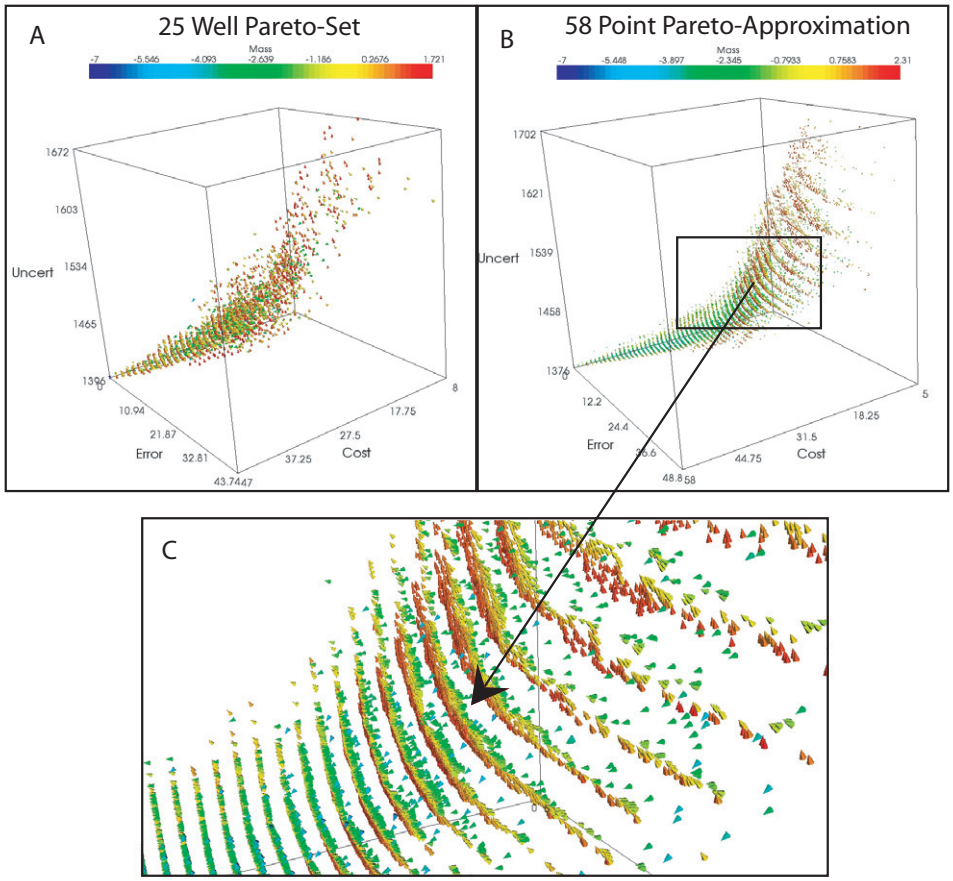

Fig. 3. Plots A and B show the 25 well Pareto-set and 58 point Pareto-approximation respectively. Plot C shows an enlarged view of a densely populated area of the 58 point Pareto-approximation. COST, ERROR, and UNCERT are plotted on the X, Y, and Z-axes, and MASS is plotted using color.

the algorithm [65]) and a reference set (i.e., the true Pareto set or best known solution). Small values of convergence are preferred indicating a small average distance to the reference set. The *runtime ε-performance* metric, recently proposed by Kollat and Reed [41] measures the proportion of ε-nondominated solutions which have been found within an acceptable ε-error for the reference set. This metric uses the concept of ε-nondominance where the desired precision for each objective is specified, and the approximation and reference sets are then sorted based on this precision. In this way, the "strictness" of the ε-performance metric can be strengthened or relaxed based on the precision requirements of the user. The ε-performance metric ranges from zero to one where a metric value of one indicates 100-percent convergence based on the specified ε-nondominance precision of the reference set. The *unary ε-indicator* metric [65] quantifies the smallest distance that an approximation set must be translated in order to completely dominate a reference set. Small values of this metric are desirable as this indicates a closer approximation to the reference set. For additional details on the unary ε-indicator metric, see Zitzler

and Thiele [66] and Zitzler et al. [65]. The *hypervolume* metric [67] quantifies the volume of the approximation set with respect to some reference or nadir point. When a reference set is available, this metric can be calculated with respect to the reference set. In other words, when minimizing objectives, the hypervolume can be calculated as the difference in volume between an approximation set and the reference set. When calculating the hypervolume as a difference, small values are optimal indicating small difference with respect to the reference set. The hypervolume metric is an excellent measure of solution set diversity or spread across the full range of tradeoffs. The statistical metrics-based evaluation framework used in this study provides a direct measure of performance differences for the tested MOEAs with respect to their ability to converge while maintaining a diverse representation of tradeoffs [65, 68].

4.3 Algorithm Configurations

Various algorithm configurations were tested on both the 25 well and 58 point LTM test cases. Preliminary analysis of a variety of configurations allowed us to focus on six major configurations which embody the major findings of the study. These six configurations are now described in detail.

ε-NSGAII. This configuration refers to the original ε-NSGAII algorithm, and was chosen as a current performance benchmark. The ε-NSGAII was parameterized to use an initial population size of 12 individuals and an ε-nondominated archive injection rate of 25-percent. This means that following each connected run, a new population is generated based on the archived solutions (at a rate of 25-percent), and 75-percent random solutions. This injection rate was previously found to produce the most efficient algorithm performance for the ε-NSGAII relative to other injection rates [41]. Additional relevant parameters include the probability of SBX crossover = 1.0 [46], the probability of polynomial mutation = 0.02 [10], the distribution index for SBX crossover = 15, and the distribution index for polynomial mutation = 20, all of which are based on prior literature recommendations.

ε-hBOA-Base. This notation refers to a base version of the ε-hBOA algorithm as described in Section 3.2 and Figure 2. The ε-hBOA-Base implementation replaces the ε-NSGAII's traditional crossover and mutation operators with Bayesian network model building and simulation of a new population based on this model. This configuration then uses binary crowded tournament selection available within the ε-NSGAII and originally utilized by the NSGAII [45] to determine whether newly simulated population members should replace their parents. In addition, ε-NSGAII's dynamic population sizing techniques are utilized within the ε-hBOA with an ε-nondominated archive injection rate of 25-percent. Pelikan [29] showed

previously that the population size required for optimal model building within hBOA is on the order of $O(2^k n^{1.05})$ where k is the building block order and n is the number of binary decision variables. Assuming a lower bound building block complexity of $k = 4$ decisions (i.e., no more than four sampling locations are jointly important), this would require a population of approximately 1000 individuals. This lower bound population size was chosen as the initial population size for the ε-hBOA-Base configuration. However, as stated, this reflects a lower bound complexity and the ε-nondominated archive injection will ensure that the population size increases as search progresses.

ε-hBOA-Archive. This notation refers to a version of the ε-hBOA which utilizes the current ε-nondominated archive in the Bayesian network model building. In other words, following each generation, the current best found solutions that are stored in the ε-nondominated archive are combined with the currently evolving population to build the Bayesian network model. This configuration was tested to confirm if the inclusion of additional (high quality) information in the model building would improve algorithm performance.

ε-hBOA-RTR. The original hBOA algorithm utilized restricted tournament recombination (RTR) as recommended by Pelikan [29] to determine which children should replace parent members of the population. Based on this recommendation, a configuration of the ε-hBOA algorithm which utilized RTR instead of binary crowded tournament recombination was tested.

ε-hBOA-Static. According to Pelikan's [29] theoretical population sizing requirements for the hBOA, the required population can become quite large, especially for problems with many decision variables. To test the importance of population size, a static population size variant of the ε-hBOA algorithm was tested that eliminated the dynamic population sizing available within the ε-NSGAII. Requiring the specification of a static population size adds an additional parameter to the algorithm that is difficult to estimate *a priori*. For the smaller 25 well test case, the static population size was chosen based on $2l$ (where l is the number of decisions) generations of evolution [69,70] (i.e, 50 for the 25 well case) and a maximum runtime of 200,000 evaluations for a total static population of 4,000 individuals. For the larger 58 point test case, the population size was based on the average archive size attained for individual runs of the ε-hBOA-Base. Several prior EA studies [25,28,71] have shown that MOEA population size is a direct function of the Pareto set size. The population size is very important to the hBOA since the quality of its probabilistic models is directly correlated with the size of the population sampled.

ε-hBOA-Hybrid. Since one of ε-NSGAII's main strengths is utilizing small populations initially to pre-condition search and since the hBOA requires large populations to optimally construct the Bayesian network, a hybrid version of

the two techniques is tested in this study. The ε-hBOA-Hybrid configuration utilizes the ε-NSGAII to pre-condition search for 10-percent of the total run duration and then switches to a static population size using the ε-hBOA. The static population size utilized by ε-hBOA is strictly based on what the ε-NSGAII found during the initial portion of the run at a 25-percent injection rate. If the ε-NSGAII found 1000 ε-nondominated archive solutions during the first 10-percent of the run, the static population size used by the ε-hBOA would be 4000 individuals, 1000 of which are from the archive and the remaining 3000 of which are initially generated at random. Once the ε-hBOA-Hybrid version switches to the ε-hBOA, the population size remains static.

Other Algorithm Parameters. A number of algorithm parameters were specified identically across all configurations. The number of generations of evolution per connected run was specified as $2l$ [69, 70] where l is the number of decisions. This means that for the 25 well test case, each run contained 50 generations and for the 58 point case, each run contained 116 generations. The maximum runtime was chosen based on computational feasibility and was expressed as a total number of function evaluations. For the 25 well test case, the maximum number of evaluations was set at 200,000 and for the 58 point test case, the maximum number of evaluations was set at 2-million. The ε-dominance precision settings for each of the four design objectives were chosen to result in very high precision Pareto-sets for each test case and were specified as $\varepsilon = (1.0, 0.01, 0.01, 0.01)$ for each the COST, ERROR, UNCERT, and MASS objectives respectively. Finally, 50 random seed trials were conducted for each algorithm configuration to provide a means of assessing reliability. Overall, 900 algorithm runs were conducted for this study, 300 for the 25 well test case, 300 to generate the 58 point Pareto-approximation, and another 300 to obtain metric results for the 58 point test case relative to the generated reference set. This represents well over 1-year of continuous computing on a serial machine. However, individual runs were distributed across the LION-XO high performance computing cluster available at The Pennsylvania State University in order to be completed within a couple of weeks.

5 Results

5.1 Effectiveness of Search

Figure 4 displays a histogram showing the relative effectiveness of each algorithm configuration at identifying reference set solutions for the 58 point test case. Light grey bars signify the total percentage of the reference set's solutions identified by a particular configuration across all random seed trials. The

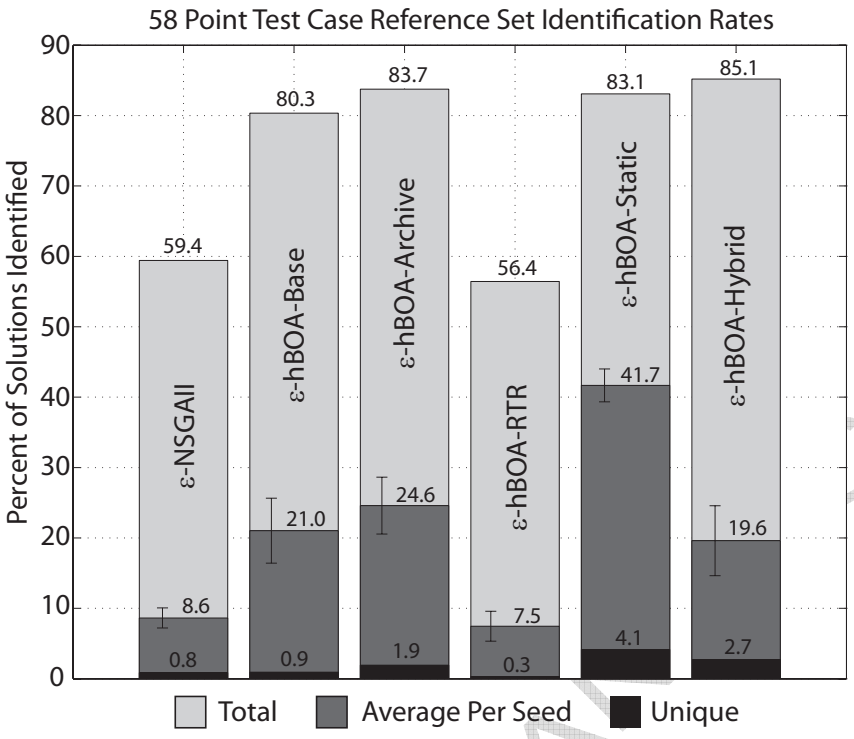

Fig. 4. Histogram displaying the relative contributions of each algorithm configuration to the 58 point test case reference set. Light gray bars indicate total contribution across all random seed trials, medium gray bars, the average per seed contribution, and the black bars, unique contribution across all seeds.

medium grey bars signify the average per seed effectiveness for each configuration and the error bars indicate one standard deviation. Finally, the black bars indicate the total unique contribution of each configuration, which is defined as the number of solutions that a particular configuration contributed across all random seed trials that no other configuration found. From the histogram, it is clear that the ε-hBOA-RTR configuration contributed the least to the reference set in all categories. The ε-hBOA-Base, ε-hBOA-Archive, ε-hBOA-Static, and ε-hBOA-Hybrid configurations contributed the largest percentages overall (between 80 and 85-percent), while the average per seed contribution of the ε-hBOA-Static configuration was the highest at 41.7-percent, more than twice that of all other configurations except the ε-hBOA-Archive. In addition, the ε-hBOA-Static configuration provided the highest contribution of unique solutions at 4.1-percent, more than twice that of any other configuration except the ε-hBOA-Hybrid. These results indicate that on a per seed basis, the ε-hBOA-Static configuration is capturing the highest percentage of the 58 point test case's reference set.

Table 1 displays metric values for each algorithm configuration for the smaller 25 well test case. Metrics were evaluated when the number of function evalua-

tions (NFE) reached 25,000 and again at the end of the run (NFE = 200,000). The reader is reminded that minimal values of convergence, ε-indicator, and hypervolume are preferred while ε-performance expresses the proportion of the reference set found, meaning that 1.0 is optimal for this metric. Hypervolume metric results are not available at NFE = 25,000 because the non-dominated sets which are required to calculate the metric were not output throughout the run, only at the end of the run. This table shows the average metric value for all 50 random seed trials as well as the standard deviation in parentheses. In addition, a Kruskal-Wallis [72] non-parametric statistical test was used to test the statistical significance of performance differences at the 95-percent confidence level for the metric distributions attained for each algorithm configuration. The results of the statistical tests were used to rank the relative performances of each algorithm configuration and the top performing algorithm configurations for each metric are highlighted in bold in Table 1. In Table 1, the ε-hBOA-Static configuration is the worst performer early in the run in terms of all metrics, but subsequently becomes the best performer by the end of the run. In terms of early run results, it appears that the ε-hBOA-Archive and ε-hBOA-Base configurations are the top performers. The ε-hBOA-Base configuration maintains good overall performance throughout the run, and generally ends the run in second place. It seems that the ε-hBOA-Archive and ε-hBOA-Hybrid configurations are comparable to one another by the end of the run and the ε-hBOA-RTR configuration is the worst performing configuration overall. It is interesting to note that the ε-NSGAII performs well in terms of both the ε-indicator and hypervolume metrics. The performance of the ε-NSGAII in terms of the ε-indicator and hypervolume metrics is likely due to early rapid approximation of the 25 well test case reference set as was shown by Kollat and Reed [30].

Table 2 displays early run (NFE = 250,000) and end-of-run (NFE = 2,000,000) metric results for the 58 point test case. Statistically significant rankings are displayed similarly to Table 1. Early in the run, the ε-hBOA-Base configuration is the top performer while the ε-hBOA-Archive configuration trails in second place. The computational scaling limitations highlighted by Kollat and Reed [30] are caused the ε-NSGAII to be the worst performer overall. At the end of the run, the ε-hBOA-Static configuration is again the top performer, similarly to the 25 well test case. Although the ε-hBOA-Static configuration has the drawback of requiring a careful specification of the population size and exhibits an early run lag in terms of finding reference set solutions, it ultimately produces the best performance over all other configurations. Again, only top ranked configurations with statistically significant performance differences are highlighted in bold in Table 2. The reader should also note that there are many cases for the ε-indicator and hypervolume metrics where multiple configurations share statistically similar performance. To further elucidate performance differences, the next section provides a detailed analysis of search performance dynamics.

Table 1
NFE = 25,000 and end-of-run mean and (standard deviation) metric results for the 25 well test case. Metric values in bold text indicate statistical superiority at a 95-percent confidence level.

NFE=25,000	Conv.$\times 10^3$	ε-Perf.	ε-Ind.	Hyper.$\times 10^{-5}$
ε-NSGAII	14.210 (2.876)	0.127 (0.025)	**3.965 (0.313)**	NA
ε-hBOA-Base	**2.751 (0.507)**	0.420 (0.015)	4.297 (1.155)	NA
ε-hBOA-Archive	2.911 (0.665)	**0.427 (0.018)**	4.005 (0.857)	NA
ε-hBOA-RTR	8.880 (1.807)	0.205 (0.023)	4.518 (0.975)	NA
ε-hBOA-Static	36.910 (1.923)	0.008 (0.002)	6.668 (0.468)	NA
ε-hBOA-Hybrid	15.610 (2.940)	0.106 (0.022)	4.003 (0.344)	NA
NFE=200,000	Conv.$\times 10^3$	ε-Perf.	ε-Ind.	Hyper.$\times 10^{-5}$
ε-NSGAII	1.560 (0.270)	0.688 (0.015)	3.494 (0.308)	5.085 (0.619)
ε-hBOA-Base	0.866 (0.147)	0.698 (0.015)	3.460 (0.571)	5.381 (0.444)
ε-hBOA-Archive	0.994 (0.274)	0.683 (0.020)	3.539 (0.657)	5.658 (0.352)
ε-hBOA-RTR	1.740 (0.304)	0.557 (0.021)	3.478 (0.256)	5.709 (0.392)
ε-hBOA-Static	**0.787 (0.154)**	**0.752 (0.011)**	**3.133 (0.317)**	5.176 (0.617)
ε-hBOA-Hybrid	1.140 (0.253)	0.703 (0.018)	3.465 (0.174)	5.511 (0.337)

5.2 Search Efficiency and Reliability

Figure 5 plots ε-performance success rates for each algorithm configuration for the 25 well and 58 point test cases. The ε-performance success rate is defined as the percentage of random seed trials that exceed some threshold of metric performance for a given NFE to provide insight into algorithm dynamics. Cases with nearly vertical success rate cumulative distribution curves represent highly reliable performance where all of a configuration's random seed trials exceeded the threshold of performance at approximately the same NFE. Likewise, small slopes for success rate curves indicate low search reliability. For the 25 well test case, ε-performance success rates are plotted at thresholds of 0.25, 0.5, and 0.7 (corresponding with quantification of 25, 50, and 70-percent of the reference set) in Figures 5A through 5C respectively. Figure 5A indicates that all configurations are fairly reliable at quantifying 25-percent of the 25 well test case's reference set. However, it appears that the ε-hBOA-Base and ε-hBOA-Archive configurations exceed this threshold fastest, while the ε-NSGAII and ε-hBOA-Static configuration are the slowest at meeting this goal. A shift in relative configuration success rates can be noted in Figure 5B where the ε-hBOA-Static configuration moves from

Table 2
NFE = 250,000 and end-of-run mean and (standard deviation) metric results for the 28 point test case. Metric values in bold text indicate statistical superiority at a 95-percent confidence level.

NFE=250,000	Conv.$\times 10^3$	ε-Perf.	ε-Ind.	Hyper.$\times 10^{-6}$
ε-NSGAII	16.028 (1.403)	0.005 (0.002)	3.796 (0.240)	NA
ε-hBOA-Base	**7.574 (1.393)**	**0.066 (0.014)**	3.713 (0.372)	NA
ε-hBOA-Archive	7.707 (1.185)	0.061 (0.015)	**3.666 (0.305)**	NA
ε-hBOA-RTR	12.959 (1.703)	0.018 (0.006)	3.783 (0.202)	NA
ε-hBOA-Static	11.945 (0.778)	0.005 (0.001)	3.908 (0.258)	NA
ε-hBOA-Hybrid	14.952 (1.532)	0.007 (0.004)	3.787 (0.272)	NA
NFE=2,000,000	Conv.$\times 10^3$	ε-Perf.	ε-Ind.	Hyper.$\times 10^{-6}$
ε-NSGAII	8.029 (1.001)	0.086 (0.014)	2.903 (0.170)	2.014 (0.296)
ε-hBOA-Base	4.897 (1.075)	0.243 (0.040)	3.252 (0.415)	2.096 (0.323)
ε-hBOA-Archive	5.688 (1.015)	0.208 (0.045)	3.136 (0.288)	2.139 (0.265)
ε-hBOA-RTR	10.413 (1.964)	0.074 (0.021)	3.365 (0.222)	2.492 (0.326)
ε-hBOA-Static	**3.187 (0.388)**	**0.414 (0.023)**	2.927 (0.223)	**1.830 (0.388)**
ε-hBOA-Hybrid	6.375 (1.478)	0.195 (0.049)	2.908 (0.192)	1.992 (0.324)

last place to third place, and ε-hBOA-Hybrid moves from 5th place to 4th place. It should be noted that the ε-NSGAII and ε-hBOA-Hybrid versions are equivalent in Figure 5A since ε-NSGAII is performing the search early in the ε-hBOA-Hybrid configuration's run. In Figure 5C, a pronounced shift in success rate is exhibited where the ε-hBOA-Static configuration moves to first place and all other configurations fail to find the reference set to an accuracy of 70-percent in 200,000 evaluations.

For the 58 point test case, the success rate plots shown in Figures 5D through 5F indicate that the ε-hBOA-Static configuration is again the fastest and most reliable for all ε-performance thresholds of 0.1, 0.2, and 0.3. In addition, as the success rate threshold becomes more rigorous, other configurations fail to quantify the reference set at increased accuracy. For example, the ε-NSGAII and ε-hBOA-RTR configurations completely fail to find 20-percent of the reference set, while all but the ε-hBOA-Static configurations fail to find anything above 30-percent of the reference set in 2-million design evaluations.

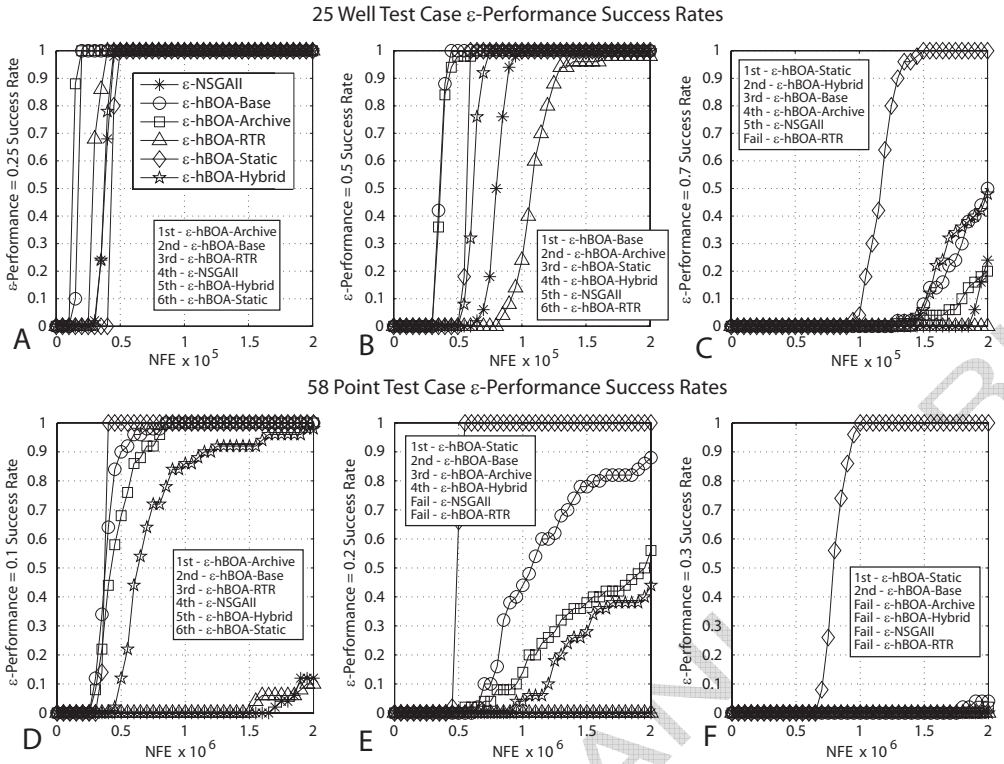

Fig. 5. Epsilon-performance success rates plotted versus function evaluations (NFE) at specified thresholds for both the 25 well and 58 point test cases.

5.3 Balancing Efficiency and Reliability

Figure 6 shows the dynamic performances of three configurations: ε-NSGAII, ε-hBOA-Base, and ε-hBOA-Static. The choice of these configurations was based in part on their overall performance and because they represent three fundamentally different algorithms. Figure 6 displays runtime dynamics for the convergence and ε-performance metrics on both the 25 well and 58 point test cases. Dynamic performance is captured by plotting the metric attained by each seed as a function of the number of designs evaluated for all 50 random seed trials of the three algorithms. For the 25 well test case (Figures 6A and 6B), it is clear that the final effectiveness of all configurations is somewhat comparable, which is expected given that the problem's size is modest. Kollat and Reed [9] have already demonstrated that the ε-NSGAII can successfully solve this test case. This is an interesting lower bound problem complexity where the ε-NSGAII has the maximal chance of exceeding the ε-hBOA configurations' performances. The ε-NSGAII's search proceeds very quickly initially (i.e., NFE $= 10,000$), but is rapidly overtaken by the ε-hBOA-Base configuration. The ε-hBOA-Base exhibits the fastest initial performance of the three algorithms compared. The ε-hBOA-Static configuration is slow to start, but ultimately achieves the best performance of the three configurations after an initial lag period for evolutionary search.

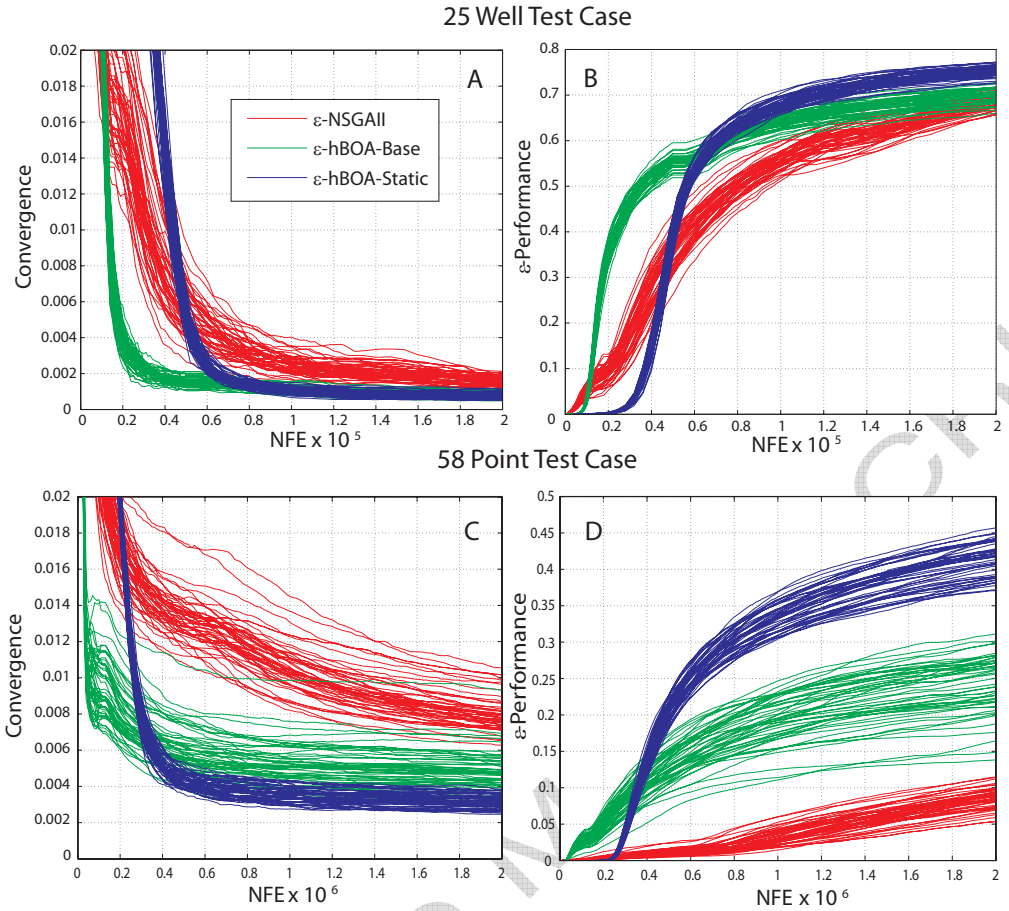

Fig. 6. Convergence and ε-performance dynamics for the ε-NSGAII, ε-hBOA-Base, and ε-hBOA-Static configurations. Plots show each of the 50 random seed trials for each configuration plotted as metric value versus function evaluations (NFE).

For the 58 point test case (Figures 6C and 6D), the difference in final performance of each configuration is more pronounced. The ε-NSGAII does not scale well for the 58 point test case. While the ε-hBOA-Base configuration provides a very rapid initial solution approximation, the ε-hBOA-Static configuration quickly overtakes the performance of the ε-hBOA-Base configuration. It should be noted that the ε-hBOA-Base and the ε-hBOA-Static configurations represent a tradeoff between efficiency and the ultimate search effectiveness. The ε-hBOA-Base has the strong advantage of eliminating the need to specify any search parameters while the ε-hBOA-Static provides highly reliable search. In general for users with severe computational constraints, the ε-hBOA-Base would be the superior choice for attaining rapid approximations while not requiring any search parameters. If users can use high performance computing or other means for reducing their computational constraints, use of the ε-hBOA-Static configuration would be beneficial. As was done in this study, the two configurations could be used in tandem by using the ε-hBOA-Base's archive size from a preliminary run to determine an effective population size for the ε-hBOA-Static configuration.

5.4 Understanding Convergence Versus Diversity

In order to further assess performance differences between the ε-NSGAII, ε-hBOA-Base, and ε-hBOA-Static configurations, the best random seed trial from each configuration (based on ε-performance end-of-run metric values) was used to generate a Pareto approximation for the 58 point test case containing 15,355 solutions. In total, 8-percent of this set was contributed by all three configurations, and 32.4-percent of the set was contributed by both configurations of the ε-hBOA. However, the ε-hBOA-Static configuration contributed 37.4-percent uniquely to the set (meaning that no other configuration found these particular solutions). This represents nearly four times the unique contribution of the ε-hBOA-Base configuration (10.5-percent), and eight times the unique contribution of the ε-NSGAII (4.2-percent). In all, the ε-hBOA-Static configuration would have found 82-percent of this set.

Figure 7 shows the approximate Pareto set generated from the best trials of the ε-NSGAII, ε-hBOA-Base, and ε-hBOA-Static configurations for the 58 point test case. COST, ERROR, and UNCERT are plotted on the X, Y, and Z-axes. The plot shown in Figure 7 differs from the plotting technique of Figure 3 in that MASS objective is portrayed by the orientation of the solution cones (pointing up means high MASS and pointing down means low MASS). This is because color is reserved in this plot to reflect which algorithm generated the solution (blue represents ε-hBOA-Static, green - ε-hBOA-Base, and red - ε-NSGAII). Subplots B, C, and D in Figure 7 show detailed views of the regions highlighted in subplot A. These detailed views show the predominant contribution of the ε-hBOA-Static configuration indicated by the blue points. In general, structural differences between the ε-hBOA-Base, and ε-hBOA-Static solution contributions seem to be limited to the shear quantity of solutions discovered. In other words, the ε-hBOA-Static configuration finds many more solutions, ultimately filling out portions of the objective space more densely. An interesting feature of these detailed views is that they show how the ε-NSGAII contributes uniquely to the set. Figure 7B shows that the ε-NSGAII finds several solutions at the farthest extent of the set corresponding to very high COST where most of the potential monitoring points are sampled. In addition, there are regions of unique contribution of the ε-NSGAII visible in Figures 7C and 7D. The unique contributions of the ε-NSGAII are all characterized by their sparsity. This indicates that the ε-NSGAII is better at finding solutions in regions of the set with low solution density. This explains the highly ranked performance of the ε-NSGAII in terms of the ε-indicator and hypervolume metrics in Tables 1 and 2, because these metrics are sensitive to the geometric distribution of solutions across the objective space.

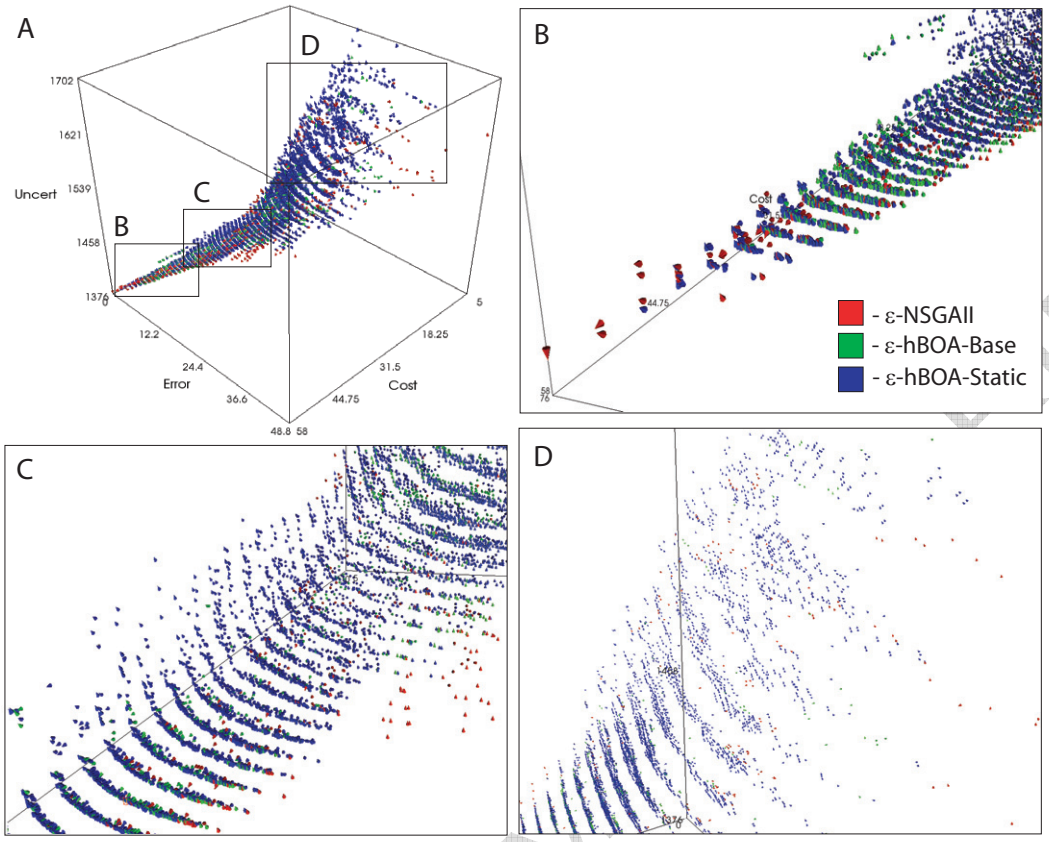

Fig. 7. Pareto approximation set generated by combining the best trial runs of each the ε-NSGAII, ε-hBOA-Base, and ε-hBOA-Static configurations. COST, ERROR, and UNCERT are plotted on the X, Y, and Z-axes while the MASS objective is portrayed by the orientation of the solution cone (up = high MASS, down = low MASS). Red solutions were contribution by the ε-NSGAII, green solutions by the ε-hBOA-Base, and blue solutions by the ε-hBOA-Static configuration.

6 Discussion

The results of this study indicate that the ε-hBOA is more effective than the ε-NSGAII at optimizing both the 25 well and 58 point LTM test cases. Performance differences between the ε-NSGAII and ε-hBOA are notably more pronounced on the larger 58 point LTM test case. Relative performance differences between the various configurations of the ε-hBOA are informative with regard to the factors that affect the quality of Bayesian network models. The relatively poor performance of the ε-hBOA-RTR configuration can be attributed to a lack of selection pressure, ultimately slowing the progress of the algorithm. Restricted tournament recombination randomly selects solutions from the parent and child populations and places them in competition within binary tournaments to determine which solution proceeds to the next generation. Alternatively, crowded binary tournament selection utilized by the ε-NSGAII first ranks both parents and children according to their relative

position with respect to non-domination. The subsequent population is then filled by allowing the highest ranked solutions to proceed to the next generation. This recombination method places more emphasis on highly fit solutions and hence increases the selection pressure of the algorithm.

The changes in the relative performances of each configuration early versus late in the runs (refer to Tables 1 and 2 as well as Figures 5 and 6) can be explained in terms of population size effects. The ε-hBOA-Base configuration uses an arbitrarily small initial population size, which allows it to rapidly approximate the Pareto set. However, the initial small population provides a reduced statistical sample for optimal model building [29] and as a result, the long-term performance of this configuration suffers. Alternatively, the ε-hBOA-Static configuration uses a static population size that is initially very large, which initially slows the algorithm's evolutionary exploration. In the long term however, the ε-hBOA-Static configuration yields more accurate models for decision interdependencies and enhanced search.

Performance differences between the ε-NSGAII and the ε-hBOA configurations can largely be attributed to how solution density impacts ε-NSGAII's SBX crossover operator versus ε-hBOA's Bayesian network model building. Figure 7 showed that the ε-NSGAII is better at finding solutions in sparsely populated regions of the objective space. The SBX crossover operator utilized by the ε-NSGAII is a parent-centric mating approach whereby child solutions have a higher probability of being generated close to their parents [73]. When the ε-NSGAII finds a solution in these sparsely populated regions, the SBX crossover operator emphasizes a relatively local search (i.e., small changes in the decisions), ultimately allowing it to search these regions of the space more thoroughly. However, Figure 7 also showed that the ε-hBOA was more effective at generating solutions in densely populated regions. There are clearly more ways to choose 30 of the 58 sampling points than 56 of the 58 points. These combinatorial differences affect solution density throughout the search space. This ultimately diminishes the ability of the ε-hBOA to model sparse areas of the search space. Since it is more difficult for the ε-hBOA to learn the structure of these sparse regions, the net result is that it is less likely that Pareto-optimal solutions will be simulated in these regions.

It was shown that the ε-hBOA-Base and ε-hBOA-Static versions were the top-performing search schemes. In choosing between these two configurations, the computational resources and expertise of the user must be considered. Overall, the ε-hBOA-Static configuration had the best long-term performance, but required careful specification of the population size and significant run-time to achieve high quality model building. In contrast, the ε-hBOA-Base configuration is superior for rapid solution set approximations and it eliminates the need to specify the population size parameter. If appro-

priate computational resources are available, these two configurations could be used in tandem. If no prior knowledge is available regarding the Pareto set size (which is more typical), the ε-hBOA-Base configuration could be run to determine the optimal population size for the ε-hBOA-Static configuration. Once an approximation of the Pareto optimal solution set size is determined, the population size of the ε-hBOA-Static configuration can be specified based on a one-to-one ratio relative to the Pareto set size. In addition, since the reliability of the ε-hBOA-Static configuration is high, random seed trial analysis can be reduced or eliminated, dramatically reducing the computational burden of using the ε-hBOA-Static configuration.

The success of the ε-hBOA relative to the ε-NSGAII implies that there are strongly inter-related decisions within the LTM design problem and that the ε-hBOA can exploit these dependencies through Bayesian network model building. The physical contaminant plume structure implicit to the LTM design problem likely does present some degree of hierarchical difficulty, also made apparent by the success of the ε-hBOA relative to the ε-NSGAII. The 25 well and 58 point test cases provide a means for testing if hierarchical problem difficulty is impacting the performance of the search algorithms. The ε-NSGAII has been well documented [26] in solving the smaller 25 well test case efficiently and reliably. If the sampling decisions were statistically independent as assumed by the ε-NSGAII's mating and mutation operators, then the ε-NSGAII should have been able to attain superior performance for the smaller test case. Figure 6 demonstrates that this is not the case and that even for the small 25 well test case, the ε-hBOA has superior performance. Moreover, the large test case success rate plots and runtime dynamics for the ε-NSGAII indicate that the algorithm is not reliable and that the best case performance for the algorithm poorly represents its expected performance.

The region where the ε-NSGAII has the most distinct performance advantage is very near the solution where all points are sampled. This region is actually the least challenging to predict by the decision maker and likely of limited interest. The compromise regions near the center of the reference set shown in Figure 7 were best quantified by the ε-hBOA and are generally of most interest for exploring the impacts of cost-savings on the remaining objectives. The lower bound cost portion of the tradeoffs (i.e., regions of high errors in Figure 7) is also an interesting region in terms of search difficulty. This portion of the tradeoff surface is at the feasibility boundary for the constrained formulation of this problem. Somewhat surprisingly, this region's large objective ranges and sparseness do not impair the ε-hBOA as illustrated in Figure 7. The algorithm is able to find a superior representation of this portion of the objective space relative to the ε-NSGAII. This result implies that the overall success or failure of the ε-hBOA is more heavily influenced by statistical sampling than by the problem's constraint. The feasibility boundary still has a large number of potential solution combinations that can be sampled and

modeled (i.e., designs consisting of five sampling points drawn from the full set of 58 potential locations).

7 Conclusions

This study contributes a new MOEA termed the Epsilon-dominance Hierarchical Bayesian Optimization Algorithm (ε-hBOA), for solving large, multiobjective groundwater monitoring design problems. The new ε-hBOA was tested on a 25 well LTM design test case (a relatively small test case with around 33-million possible designs) and a larger 58 point LTM design test case containing over 2.88×10^{17} possible designs. Both test cases were optimized for four design objectives - COST, ERROR, UNCERT, and MASS. The Pareto-optimal set for the 25 well test case contained 2,472 designs and the Pareto-approximation for the 58 point test case contained 22,333 designs. The ε-NSGAII , a top performing MOEA was tested relative to five configurations of the ε-hBOA. The ε-hBOA-Base configuration utilized dynamic population sizing based on a 25-percent ε-nondominance archive injection rate similar to the ε-NSGAII. The ε-hBOA-Archive configuration included the current ε-nondominance archive in the model building following each generation. The ε-hBOA-RTR configuration utilized restricted tournament recombination rather than the crowded binary tournament selection utilized by the ε-NSGAII. The ε-hBOA-Static configuration replaced dynamic population sizing with a static population size that was specified *a priori*. The ε-hBOA-Hybrid configuration utilized the low cost search and dynamic population sizing of ε-NSGAII initially, and subsequently switched to a static population size based on this initial search. A comprehensive, metrics based analysis was performed to determine each the effectiveness, efficiency, and reliability of each ε-hBOA configuration relative to the ε-NSGAII.

The results of the study demonstrate the scaling limitations of the ε-NSGAII relative to the ε-hBOA. The ε-hBOA uses Bayesian network model building to learn the complex interdependencies which exist within the LTM design problem. In addition, because of ε-hBOA's success relative to the ε-NSGAII, the LTM problem is believed to exhibit some degree of hierarchical difficulty due to the physical structure of the contaminant plume. It was shown that the ε-hBOA-Base and ε-hBOA-Static versions were the top-performing configurations of the ε-hBOA. Overall, the ε-hBOA-Static configuration performed best, but required careful specification of the population size, and significant run-time to achieve high quality solutions. In contrast, the ε-hBOA-Base configuration was superior for rapid approximations and it eliminated the need to specify an optimal population size *a priori* since it automatically adapts its size based on search progress. If computational resources are limited, the ε-hBOA-Base configuration would be an appro-

priate choice to obtain rapid approximation to the Pareto set. However, the ε-hBOA-Base configuration could also be used to determine the optimal population size needed by the ε-hBOA-Static configuration and a subsequent run can be conducted using the ε-hBOA-Static configuration to achieve the highest quality solution possible. Since the reliability of the ε-hBOA-Static configuration is high, random seed trial analysis becomes less important, ultimately reducing the computational burden of this approach.

While this study demonstrated that the ε-hBOA is an effective improvement over current MOEAs at solving LTM design applications, the monitoring design test cases examined represent a lower bound problem complexity because they represent only a snapshot in time. The size and complexity of the search space rapidly increases when spatiotemporal monitoring decisions are considered. In addition, monitoring complexities (e.g., the existence of multiple contaminants) also adds to the difficulty of the monitoring design problem. The authors of this study are working to develop a highly reliable, parallel extension of the ε-hBOA capable of running on massively parallel computing resources to further address the computational scaling issues associated with groundwater monitoring design.

Acknowledgements

The authors would like to thank Martin Pelikan, the original developer of the Hierarchical Bayesian Optimization Algorithm, for providing the source code required to make this study possible. The authors of this work were partially supported by the United States National Science Foundation under CAREER grant CBET-0640443. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the United States National Science Foundation. The first author has also been funded in part by a Science to Achieve Results (STAR) Graduate Research Fellowship (agreement No. FP-916820) awarded by the U.S. Environmental Protection Agency (EPA). This study has not been formally reviewed by the EPA. The views expressed in this document are solely those of the authors and the EPA does not endorse any products or commercial services mentioned in this publication.

References

[1] Task Committee on Long-Term Groundwater Monitoring Design, Long-Term Groundwater Monitoring: The State of the Art, American Society of Civil Engineers, Reston, VA, 2003.

[2] D. P. Lettenmaier, Dimensionality problems in water quality network design, Water Resources Research 13 (6) (1979) 1692–1700.

[3] B. J. Wagner, Sampling design methods for groundwater modeling under uncertainty, Water Resources Research 31 (10) (1995) 2581–2591.

[4] P. Reed, B. S. Minsker, A. J. Valocchi, Cost effective long-term groundwater monitoring design using a genetic algorithm and global mass interpolation, Water Resources Research 36 (12) (2000) 3731–3741.

[5] S. Yan, B. Minsker, Optimal groundwater remediation design using an adaptive neural network genetic algorithm, Water Resources Research 42 (2006) doi:10.1029/2005WR004303.

[6] J. Wu, C. Zheng, C. C. Chien, Cost-effective sampling network design for contaminant plume monitoring under general hydrogeological conditions, Journal of Contaminant Hydrology 77 (2005) 41–65.

[7] P. Reed, B. S. Minsker, Striking the balance: Long-term groundwater monitoring design for conflicting objectives, Journal of Water Resources Planning and Management 130 (2) (2004) 140–149.

[8] P. Reed, J. B. Kollat, V. Devireddy, Using interactive archives in evolutionary multiobjective optimization: Case studies for long-term groundwater monitoring design, Environmental Modelling & Software 22 (2007) 683–692.

[9] J. B. Kollat, P. M. Reed, Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design, Advances in Water Resources 29 (6) (2006) 792–807.

[10] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons LTD, New York, NY, 2001.

[11] T. Back, D. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation, Bristol, UK, 2000.

[12] R. Salomon, Evolutionary algorithms and gradient search: Similarities and differences, IEEE Trans. Evol. Computation 2 (2) (1998) 45–55.

[13] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company, Reading, MA, 1989.

[14] V. Pareto, Cours D'Economie Politique, Vol. Volumes 1 and 2, Rouge, Lausanne, 1896.

[15] D. S. Knopman, C. I. Voss, Multiobjective sampling design for parameter estimation and model discrimination in groundwater solute transport, Water Resources Research 25 (10) (1989) 2245–2258.

[16] I. Bogardi, A. Bardossy, Multicriterion network design using geostatistics, Water Resources Research 21 (2) (1985) 199–208.

[17] D. S. Knopman, C. I. Voss, S. P. Garabedian, Sampling design for groundwater solute transport: tests methods and analysis of cape cod tracer test data, Water Resources Research 27 (5) (1991) 925–949.

[18] P. D. Meyer, A. J. Valocchi, J. W. Eheart, Monitoring network design to provide initial detection of groundwater contamination, Water Resources Research 30 (9) (1994) 2647–2659.

[19] B. R. James, S. M. Gorelick, When enough is enough: The worth of monitoring data in aquifer remediation design., Water Resources Research 30 (12) (1994) 3499–3513.

[20] N.-Z. Sun, Inverse Problems in Groundwater Modeling, Vol. 6 of Theory and Applications of Transport in Porous Media, Kluwer Academic Publishers, New York, NY, 1994.

[21] S. E. Cieniawski, J. W. Eheart, S. R. Ranjithan, Using genetic algorithms to solve a multiobjective groundwater monitoring problem, Water Resources Research 31 (2) (1995) 399–409.

[22] P. Storck, J. W. Eheart, A. J. Valocchi, A method for the optimal location of monitoring wells for detection of groundwater contamination in three-dimensional aquifers, Water Resources Research 33 (9) (1997) 2081–2088.

[23] L. M. Nunes, M. C. Cunha, L. Ribeiro, Groundwater monitoring network optimization with redundancy reduction, Journal of Water Resources Planning and Management 130 (1) (2003) 33–43.

[24] L. M. Nunes, M. C. Cunha, L. Ribeiro, Optimal space-time coverage and exploration costs in groundwater monitoring networks, Environmental Monitoring and Assessment 93 (2004) 103–124.

[25] P. Reed, B. S. Minsker, D. E. Goldberg, A multiobjective approach to cost effective long-term groundwater monitoring using an elitist nondominated sorted genetic algorithm with historical data, Journal of Hydroinformatics 3 (2) (2001) 71–90.

[26] J. B. Kollat, P. M. Reed, A framework for visually interactive decision-making and design using evolutionary multiobjective optimization (VIDEO), Environmental Modelling and Software.

[27] S. W. Mahfoud, Population sizing for sharing methods, Tech. Rep. IlliGAL Report No. 94005, University of Illinois at Urbana-Champaigne, Urbana, IL (1994).

[28] S. W. Mahfoud, Niching methods for genetic algorithms, Tech. Rep. IlliGAL Report No. 95001, University of Illinois at Urbana-Champaigne, Urbana, IL (1995).

[29] M. Pelikan, Bayesian optimization algorithm: From single level to hierarchy, Ph.D. thesis, University of Illinois (2002).

[30] J. B. Kollat, P. M. Reed, A computational scaling analysis of multiobjective evolutionary algorithms in long-term groundwater monitoring applications, Advances in Water Resources 30 (2007) 408–419.

[31] R. M. Maxwell, S. F. Carle, A. F. B. Tompson, Contamination, risk, and heterogeneity: on the effectiveness of aquifer remediation, Environmental Geology (2007) 10.1007/s00254–007–0955–8.

[32] G. E. Fogg, C. D. Noyes, F. S. Carle, Geologically based model of heterogeneous hydraulic conductivity in an alluvial setting, Hydrogeology Journal 6 (1998) 131–143.

[33] G. E. Fogg, F. S. Carle, C. Green, Connected-network paradigm for the alluvial aquifer system, in theory, modeling, and field investigation in hydrogeology, Geological Society of America 348 (2000) 25–42.

[34] W. Graham, D. McLaughlin, Stochastic analysis of nonstationary subsurface solute transport 1: Unconditional moments, Water Resources Research 25 (2) (1989) 215–232.

[35] W. Graham, D. McLaughlin, Stochastic analysis of nonstationary subsurface solute transport 2: Conditional moments, Water Resources Research 25 (11) (1989) 2331–2355.

[36] C. V. Deutsch, A. G. Journel, GSLIB: Geostatistical Software Library and User's Guide, Oxford University Press, New York, NY, 1998.

[37] P. Goovaerts, Geostatistics for Natural Resources Evaluation, Oxford University Press, New York, NY, 1997.

[38] P. Reed, T. Ellsworth, B. S. Minsker, Spatial interpolation methods for nonstationary plume data, Ground Water 42 (2) (2004) 190–202.

[39] A. Journel, C. Deutsch, Rank order geostatistics: A proposal for a unique coding and common processing of diverse data, in: E. Baafi, N. Schofield (Eds.), Proceedings of the 5th International Geostatistics Congress, Kluwer Academic Publishers, Wollongton, Australia, 1997.

[40] C. C. Coello, D. A. Van Veldhuizen, G. B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, New York, NY, 2002.

[41] J. B. Kollat, P. M. Reed, The value of online adaptive search: A performance comparison of NSGA-II, ϵ-NSGAII, and ϵMOEA, in: C. C. Coello, A. H. Aguirre, E. Zitzler (Eds.), The Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005), Lecture Notes in Computer Science 3410, Springer Verlag, Guanajuato, Mexico, 2005, pp. 386–398.

[42] Y. Tang, P. Reed, J. B. Kollat, Parallelization strategies for rapid and robust evolutionary multiobjective optimization in water resources applications, Advances in Water Resources 30 (2007) 335–353.

[43] P. Reed, M. Ferringer, T. Thompson, J. Kollat, Parallel evolutionary multi-objective optimization on large, heterogeneous clusters: An applications perspective, IEEE Transactions on Evolutionary Computing. (In Review).

[44] Y. Tang, P. Reed, T. Wagener, How efficient and effective are evolutionary multiobjective algorithms at hydrologic model calibration?, Hydrology and Earth Systems Science 10 (2006) 289–307.

[45] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Computation 6 (2) (2002) 182–197.

[46] K. Deb, R. B. Agrawal, Simulated binary crossover for continuous search space, Tech. Rep. IITK/ME/SMD-94027, Indian Institute of Technology, Kanpur (1994).

[47] G. R. Harik, F. G. Lobo, A parameter-less genetic algorithm, Tech. Rep. IlliGAL 99009, University of Illinois at Urbana-Champaign (1999).

[48] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining convergence and diversity in evolutionary multiobjective optimization, Evolutionary Computation 10 (3) (2002) 263–282.

[49] K. Deb, M. Mohan, S. Mishra, A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions, Tech. Rep. KanGAL 2003002, Indian Institute of Technology Kanpur (2003).

[50] D. E. Goldberg, The Design of Innovation: Lessons from and for Competent Genetic Algorithms, Kluwer Academic Publishers, Norwell, MA, 2002.

[51] M. Pelikan, Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms (Studies in Fuzziness and Soft Computing), Springer-Verlag, 2005.

[52] D. Heckerman, A tutorial on learning with bayesian networks, Tech. Rep. MSR-TR-95-06, Microsoft Research Advanced Technology Division (1996).

[53] B. McKay, F. Oggier, G. Royle, N. Sloane, I. Wanless, H. Wilf, Acyclic digraphs and eigenvalues of (0,1)-matrices, Journal of Integer Sequences 7.

[54] D. Heckerman, D. Geiger, M. Chickering, Learning bayesian networks: the combination of knowledge and statistical data, Tech. Rep. MSR-TR-94-09, Microsoft Research (1994).

[55] M. Pelikan, H. Mühlenbein, The bivariate marginal distribution algorithm, Springer-Verlag, 1999, pp. 521–535.

[56] G. Schwarz, Estimating the dimension of a model, The Annals of Statistics 6 (1978) 461–464.

[57] H. A. Simon, The Sciences of the Artificial, MIT Press, 1968.

[58] M. Pelikan, K. Sastry, D. E. Goldberg, Multiobjective hboa, clustering, and scalability, in: Genetic and Evolutionary Computation - GECCO 2005, Proceedings of the 2005 conference on Genetic and evolutionary computation, ACM Press, 2005, pp. 663–670.

[59] M. Pelikan, K. Sastry, D. E. Goldberg, Sporadic model building for efficiency enhancement of hierarchical boa, in: Genetic and Evolutionary Computation - GECCO 2006, Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM Press, 2006, pp. 405–412.

[60] M. Pelikan, K. Sastry, M. V. Butz, D. E. Goldberg, Performance of evolutionary algorithms on random decomposable problems, in: Parallel Problem Solving from Nature - PPSN IX, Lecture Notes in Computer Science 4193, Springer Verlag, 2006, pp. 788–797.

[61] M. Pelikan, D. E. Goldberg, Hierarchical boa solves ising spin glasses and MAXSAT, in: Genetic and Evolutionary Computation - GECCO 2003, PT II, Lecture Notes in Computer Science 2724, Springer Verlag, 2003, pp. 1271–1282.

[62] M. Pelikan, A. K. Hartmann, K. Sastry, Hierarchical boa, cluster exact approximation, and ising spin glasses, in: Parallel Problem Solving from Nature - PPSN IX, Lecture Notes in Computer Science 4193, Springer Verlag, 2006, pp. 122–131.

[63] G. R. Harik, E. Cuantu-Paz, D. E. Goldberg, B. L. Miller, The gambler's ruin problem, genetic algorhtms, and the sizing of populations, in: Proceedings of the 1997 IEEE Conference on Evolutionary Computation, IEEE Press, Piscataway, NJ, 1997, pp. 7–12.

[64] K. Deb, S. Jain, Running performance metrics for evolutionary multi-objective optimization, Tech. Rep. KanGAL 2002004, Indian Institute of Technology Kanpur (2002).

[65] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132.

[66] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms - a comparative case study, in: A. Eiben, T. Back, M. Schoenauer, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature (PPSN V), Lecture Notes in Computer Science, Springer-Verlag, Berlin, Amsterdam, The Netherlands, 1998, pp. 292–301.

[67] E. Zitzler, D. Brockhoff, L. Thiele, The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration, in: S. O. et al. (Ed.), The Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007), Lecture Notes in Computer Science 4403, Springer Verlag, Japan, 2007, pp. 862–876.

[68] J. Knowles, L. Thiele, E. Zitzler, A tutorial on the performance assessment of stochastic multiobjective optimizers, 214, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich (2005).

[69] D. Thierens, D. E. Goldberg, A. G. Pereira, Domino convergence, drift, and the temporal-salience structure of problems, in: The 1998 IEEE International Conference on Evolutionary Computation, IEEE Press, 1998, pp. 535–540.

[70] P. Reed, B. S. Minsker, D. E. Goldberg, Designing a competent simple genetic algorithm for search and optimization, Water Resources Research 36 (12) (2000) 3757–3761.

[71] N. Khan, D. E. Goldberg, M. Pelikan, Multi-objective bayesian optimization algorithm, Tech. rep., University of Illinois at Urbana-Champaign (March 2002 2002).

[72] W. Conover, Practical nonparametric statistics, 3rd Edition, Wiley series in probability and statistics. Applied probability and statistics section., Wiley, New York, 1999.

[73] K. Deb, A. Anand, D. Joshi, A computationally efficient evolutionary algorithm for real-parameter optimization, Evolutionary Computation 10 (4) (2002) 371–395.