

This article was originally published in a journal published by Elsevier, and the attached copy is provided by Elsevier for the author's benefit and for the benefit of the author's institution, for non-commercial research and educational use including without limitation use in instruction at your institution, sending it to specific colleagues that you know, and providing a copy to your institution's administrator.

All other uses, reproduction and distribution, including without limitation commercial reprints, selling or licensing copies or access, or posting on open internet sites, your personal or institution's website or repository, are prohibited. For exceptions, permission may be sought for such use through Elsevier's permissions site at:

<http://www.elsevier.com/locate/permissionusematerial>

# A computational scaling analysis of multiobjective evolutionary algorithms in long-term groundwater monitoring applications

J.B. Kollat, P.M. Reed \*

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

Received 25 January 2006; received in revised form 9 May 2006; accepted 24 May 2006  
Available online 14 July 2006

## Abstract

This study contributes a detailed assessment of how increasing problem sizes (measured in terms of the number of decision variables being considered) impacts the computational complexity of using multiple objective evolutionary algorithms (MOEAs) to solve long-term groundwater monitoring (LTM) applications. The epsilon-dominance non-dominated sorted genetic algorithm II ( $\epsilon$ -NSGAI), which has been shown to be an efficient and reliable MOEA, was chosen for the computational scaling study. Four design objectives were chosen for the analysis: (i) sampling cost, (ii) contaminant concentration estimation error, (iii) local uncertainty, and (iv) contaminant mass estimation error. The true Pareto-optimal solution set was generated for 18–25 well LTM test cases in order to provide for rigorous algorithm performance assessment for problems of increasing size. Results of the study indicate that the  $\epsilon$ -NSGAI exhibits quadratic computational scaling with increasing LTM problem size. However, if the user is willing to accept an approximation to the Pareto-optimal solution set,  $\epsilon$ -dominance can be used to reduce the computational scaling of MOEAs to be linear with increasing problem sizes. This study provides a basis for advancing the size and scope of water resources problems that can be effectively solved using MOEAs.

© 2006 Published by Elsevier Ltd.

*Keywords:* Long-term groundwater monitoring; Evolutionary algorithms; Multi-objective optimization; Computational scaling

## 1. Introduction

This study contributes a detailed assessment of how increasing problem sizes impacts the computational complexity of using multiple objective evolutionary algorithms (MOEAs) in long-term groundwater monitoring (LTM) applications. Problem size in this study is measured in terms of the number of design decision variables being considered. Computational complexity (or scaling) can be defined as a measure of how problem sizes impact the growth rate of the average number of design evaluations required by an MOEA to approximate a solution to an application. Building on a recent comparative analyses of

MOEA effectiveness [1–3], this study characterizes the computational complexity of the epsilon-dominance non-dominated sorted genetic algorithm II ( $\epsilon$ -NSGAI) [1,2] developed by the authors. This algorithm has been proven to be more efficient and reliable relative to other state-of-the-art MOEAs [1,2] in the LTM application area. This study's computational scaling analysis is based on a suite of LTM test cases formulated to test a range of problem sizes. Formally, LTM design can be defined as the assessment of groundwater quality over long time-scales to provide “sufficient and appropriate information” to assess and possibly modify mitigation or contaminant control measures to ensure that they are adequately protective of human and ecological health [4].

In general, groundwater monitoring design has been shown to be a challenging optimization problem with multiple conflicting objectives and very large discrete decision

\* Corresponding author. Tel.: +1 814 8632940; fax: +1 814 8637304.  
E-mail addresses: [juk124@psu.edu](mailto:juk124@psu.edu) (J.B. Kollat), [pmr11@psu.edu](mailto:pmr11@psu.edu), [preed@engr.psu.edu](mailto:preed@engr.psu.edu) (P.M. Reed).

spaces [4–13]. Lettenmaier [14] referred to the scaling challenges posed by the LTM network design problem as being a “curse of dimensionality”. Knopman and Voss [5,15] recognized that the groundwater quality network design problem has many mathematical similarities to the classical combinatorial knapsack problem (i.e., discrete decision spaces that grow exponentially with increasing problem size). Reed and Minsker [13] used the LTM problem to illustrate that MOEAs are capable of solving a new problem class [16,17] that they termed high-order Pareto optimization (i.e., problems with three or more design objectives). In general, 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 [18]. The Pareto-optimal front is obtained by plotting these solutions according to their objective values yielding an  $M - 1$  dimensional surface where  $M$  is the total number of design objectives. MOEAs’ can solve highly nonlinear, discrete, and non-convex problems without differentiation [19–21] and their population-based search enables them to evolve entire tradeoff (or Pareto) surfaces within a single optimization run for problems with huge decision spaces.

There has been a modern confluence of systems analysis research towards approaches that emphasize multiple objectives (see reviews [22–25]). This trend is clearly evident in the water resources literature over the past decade [3,9,12,13,26–30]. Recent MOEA applications demonstrate that a growing body of researchers in both the water resources and broader systems analysis communities are seeking to use MOEAs in high-order Pareto optimization [13,16,17,25,31–33]. Moreover, many recent multiobjective optimization applications within the water resources literature are seeking to solve applications with large numbers of continuous, integer, and binary decisions [34–38]. For example, recent MOEA applications in hydrologic model calibration [39], non-point source pollution management [34], groundwater management [36], and distribution systems [35] consider complex integer, continuous, or mixed decisions. It should be noted that a key strength of MOEAs is their ability to rapidly approximate the true Pareto surface even if it is not exactly quantified, which can be sufficient given computational constraints.

The LTM problem provides an excellent means of assessing MOEA computational scaling due to the problem’s large array of potential design objectives and the discrete 0/1 decision variable formulation. The 0/1 decision variable formulation used in this study represents yes/no decisions on whether to sample from a predetermined monitoring well location. The 0/1 programming formulation allows for enumerative analyses of modestly sized LTM applications. In this study, enumerations were developed for various problem sizes with an upper bound representative of what could be practically enumerated given computational constraints. In the broader context of multiple

objective water resources applications, the 0/1 decision variable formulations considered in this study provide a lower bound estimate of the computational complexities for using MOEAs in water resources applications with more complex decisions (e.g., mixed integer formulations). The purpose of this study is to provide guidance on the current computational complexity of MOEAs to clarify future research paths that will allow them to solve larger water resources applications efficiently and reliably.

The MOEA computational scaling analysis presented in this paper proceeds as follows. Section 2 presents the LTM test cases and design objectives. The  $\epsilon$ -NSGAII is then introduced in Section 3. The methodology used to develop the various sized LTM test cases used to demonstrate the computational scaling characteristics of the  $\epsilon$ -NSGAII is presented in Section 4. Section 5 provides a detailed description of the computational experiment as well as the parametrization of the  $\epsilon$ -NSGAII. Section 6 provides detailed illustrations of the scaling characteristics of the LTM problem and the ability of the  $\epsilon$ -NSGAII to approximate the Pareto-optimal solution set. The  $\epsilon$ -dominance concept is also demonstrated in this section as a means of approximating the Pareto-optimal solution set, ultimately reducing computational requirements. Section 7 provides a discussion regarding the algorithm’s computational scaling and the implications of this study for future water resources research. Conclusions of the study are presented in Section 8.

## 2. Methodology

### 2.1. Test case development

The LTM test case used in this study is based on a 50-million node flow and transport simulation originally developed by Maxwell et al. [40]. This test case represents the migration of a hypothetical perchloroethylene (PCE) plume originating from an underground storage tank. The hydrogeology of the site has been extensively characterized and is based on a highly heterogeneous alluvial aquifer located at the Lawrence Livermore National Laboratory in Livermore, California. Concentration data are provided at 58 hypothetical sampling locations in a 29 well monitoring network for a snapshot in time 8 years following the initial release of contaminant. Each well has one to three predetermined sampling locations available along its vertical axis and the sampling domain extends 650 m in the  $x$ -direction, 168 m in the  $y$ -direction, and 38.4 m in the  $z$ -direction with a minimum horizontal spacing of 10 m between wells. Additional details on this test case can be found in Reed et al. [41].

### 2.2. Objective formulation

Four design objectives were chosen for this study, each of which were minimized: (i) sampling cost, (ii) relative error of local contaminant concentration estimates, (iii)

local contaminant concentration estimation uncertainty, and (iv) contaminant mass estimation error. Objectives (ii)–(iv) were obtained using the Quantile Kriging method described in Section 2.3. Eq. (1) represents the objective formulation where  $\mathbf{F}(\mathbf{x}_\kappa)$  is a vector valued performance function in which the four objectives: cost ( $f_{\text{cost}}$ ), concentration estimation error ( $f_{\text{conc}}$ ), local uncertainty ( $f_{\text{uncert}}$ ), and mass estimation error ( $f_{\text{mass}}$ ) are minimized.

$$\text{Minimize } \mathbf{F}(\mathbf{x}_\kappa) = (f_{\text{cost}}(\mathbf{x}_\kappa), f_{\text{conc}}(\mathbf{x}_\kappa), f_{\text{uncert}}(\mathbf{x}_\kappa), f_{\text{mass}}(\mathbf{x}_\kappa)), \quad \forall \kappa \in \Omega \quad (1)$$

$$\text{Subject to } U(\mathbf{x}_\kappa) = 0 \quad (2)$$

The objectives are all a function of the vector  $\mathbf{x}_\kappa$  representing the  $\kappa$ th sampling plan in the decision space  $\Omega$ . Each component  $i$  of a sampling plan  $\kappa$  is determined from Eq. (3) resulting in a string of binary digits indicating whether or not a well is sampled. Eq. (2) subjects  $\mathbf{F}(\mathbf{x}_\kappa)$  to the constraint that no points in the interpolation domain remain unestimated. Quantile Kriging (described in Section 2.3) was used to interpolate contamination estimates at unsampled locations throughout the plume. Well sampling schemes that contain too few wells or wells that are poorly distributed in space may not have a sufficient number of data points in the Kriging neighborhoods to perform interpolation and hence result in a number of unestimated points,  $U(\mathbf{x}_\kappa)$ , in the interpolation domain (violating the constraint described by Eq. (2)).

$$x_{\kappa,i} = \begin{cases} 1, & \text{if the } i\text{th well is sampled} \\ 0, & \text{otherwise} \end{cases} \quad \forall \kappa, i \quad (3)$$

The sampling cost objective quantifies the monitoring cost of a particular sampling scheme using Eq. (4). The coefficient,  $C_S$  defines the cost per sample (normalized to one in this study). Additionally, if a well is sampled, it is assumed that all locations along its vertical axis are sampled resulting in a cost coefficient ranging from 1 to 3. The cost objective is ultimately quantified by summing the cost coefficients of each of the wells sampled in a particular scheme.

$$f_{\text{cost}}(\mathbf{x}_\kappa) = \sum_{i=1}^{n_{\text{well}}} C_S(i)x_{\kappa,i} \quad (4)$$

The relative error of local contaminant concentration estimates objective measures how the Kriged estimate of the plume using the  $\kappa$ th sampling plan differs from that obtained by sampling from all well locations. Eq. (5) quantifies the concentration error objective by summing the squared differences between the concentration estimate at a grid location  $\mathbf{u}_j$  using all wells,  $c_{\text{all}}(\mathbf{u}_j)$ , and the concentration estimate at the same grid location using the  $\kappa$ th sampling plan,  $c_\kappa(\mathbf{u}_j)$ .

$$f_{\text{conc}}(\mathbf{x}_\kappa) = \sum_{j=1}^{n_{\text{est}}} (c_{\text{all}}(\mathbf{u}_j) - c_\kappa(\mathbf{u}_j))^2 \quad (5)$$

Local contaminant concentration estimation uncertainty is quantified by summing the estimation standard deviations

obtained from Kriging at each grid location  $\mathbf{u}_j$  using Eq. (6). The standard error weight coefficient,  $A_j$ , can be used to assign importance to uncertainty estimates at different locations in the interpolation domain. For this study,  $A_j$  was assumed constant across the interpolation domain and was assigned a value of  $2\sqrt{3}$  based on the standard deviation of a uniform distribution.

$$f_{\text{uncert}}(\mathbf{x}_\kappa) = \sum_{j=1}^{n_{\text{est}}} A_j \sigma(\mathbf{u}_j) \quad (6)$$

The contaminant mass estimation error objective quantifies the relative error between the total mass of dissolved contaminant estimated using all well locations,  $\text{Mass}_{\text{all}}$ , and the contaminant mass estimated from the  $\kappa$ th sampling plan,  $\text{Mass}_\kappa$ . Eq. (7) expresses the relative mass estimation error in terms of a percentage.

$$f_{\text{mass}}(\mathbf{x}_\kappa) = \left| \frac{\text{Mass}_{\text{all}} - \text{Mass}_\kappa}{\text{Mass}_{\text{all}}} \right| \cdot 100\% \quad (7)$$

If a well sampling scheme results in unestimated points in the interpolation domain (violating the constraint described by Eq. (2)), the objectives are penalized to ensure that infeasible sampling schemes are eliminated from consideration. Eq. (8) is applied to each objective function if a feasibility violation occurs, resulting in solutions with lower fitness (i.e., higher objective values in a minimization problem).

$$\mathbf{F}_{\text{penalty}}(\mathbf{x}_\kappa) = \begin{cases} f_{\text{cost}}^{\text{penalty}} = f_{\text{cost}} + f_{\text{cost}}^{\text{max}} \\ f_{\text{conc}}^{\text{penalty}} = f_{\text{conc}} + n_{\text{est}} + U(\mathbf{x}_\kappa) + f_{\text{cost}}^{\text{max}} \\ f_{\text{uncert}}^{\text{penalty}} = f_{\text{uncert}} + n_{\text{est}} + U(\mathbf{x}_\kappa) + f_{\text{cost}}^{\text{max}} \\ f_{\text{mass}}^{\text{penalty}} = f_{\text{mass}} + n_{\text{est}} + U(\mathbf{x}_\kappa) + f_{\text{cost}}^{\text{max}} \end{cases} \quad (8)$$

Eq. (8) penalizes the objective functions based on the maximum cost of a sampling scheme,  $f_{\text{cost}}^{\text{max}}$  (which is dependent on the test case size), the total number of estimation points in the grid,  $n_{\text{est}}$  (in this case 1666 – chosen based on computational feasibility), and the total number of unestimated points,  $U(\mathbf{x}_\kappa)$ , in the infeasible sampling plan. For example, if a particular sampling plan for the 25 well test case resulted in 10 unestimated points in the interpolation grid, the fitness penalty added to the design's cost objective would be 47, and the fitness penalty added to the values for the concentration error, uncertainty, and mass error objectives would be 1723. Since the maximum cost of the system was known based on the test case data, Eq. (8) is defined so that all infeasible solutions will have costs that exceed the maximum feasible cost (i.e., for this test case, 47). The exact ranges of the other objectives were not known *a priori*, so 1723 is a conservative penalty for the uncertainty, mass error, and concentration error objectives that ensures that when penalized, their fitness values will exceed their maximum feasible values. Penalizing solutions rather than eliminating them ensures that sampling schemes which are “almost” feasible are given the opportunity to be further evolved by the MOEA into feasible

designs (for more details on this problem formulation, see Reed and Minsker [13]).

### 2.3. Spatial interpolation

Spatial interpolation of the contamination plume was conducted using Quantile Kriging (QK) based on the recommendations of Reed et al. [41]. Kriging provides a minimum error variance estimate value at an unsampled location provided the data at the sampled locations [42]. QK extends Ordinary Kriging (OK) by transforming the sample values to quantile space according to their rank. The quantile values represent the empirical cumulative distribution function (CDF) of the sample values, resulting in normalized data. Samples are Kriged in quantile space and then transformed back to concentration space using the generated CDF [43,44]. Since OK assumes stationarity of the concentration mean, moving local neighborhoods are used to estimate the expected value at each location [42]. Reed et al. [41] found that QK showed the least bias with respect to variability of PCE concentrations and preferential sampling, and was most robust in representing the plume when compared to five other interpolation methods.

For this study, the contamination plume was interpolated using a C translation of KT3D, a 3-dimensional Kriging library written in Fortran as part of the GSLIB software package [42]. For the 29 well test case, a geostatistical analysis revealed a spherical variogram structure with 0.005 nugget and 18 m range. The interpolation grid was defined by 34 blocks in the  $x$ , 7 blocks in the  $y$ , and 7 blocks in the  $z$  directions, resulting in 1666 regularly spaced estimation points. The search neighborhood size was based on an ellipsoid structure with axes lengths equal to half of each the  $x$ ,  $y$ , and  $z$  extents of the study region. The search neighborhood was divided into octants, and a maximum of one data point from each octant was used in the estimation, ensuring that clustered data points did not bias interpolation estimates. Smaller test cases which were generated to demonstrate the computational scaling characteristics of the LTM problem utilized the same geostatistical parametrization of the 29 well test case because of its higher information content. Reed et al. [41] provides a more detailed description of Quantile Kriging for interested readers.

### 3. Optimization algorithm

The  $\epsilon$ -NSGAII was chosen for this study based on its superior performance relative to the original non-dominated sorted genetic algorithm II (NSGAII) and the epsilon-dominance multi-objective evolutionary algorithm ( $\epsilon$ MOEA), and its competitive to superior performance relative to the strength Pareto evolutionary algorithm 2 (SPEA2) on the LTM problem formulated in Section 2 [1]. All of these MOEAs use real parameter simulated binary cross-over (SBX) [45], polynomial mutation [32], and elitism [32]. Since all of the algorithms use the same primary search operators, it is expected that their performance will scale similarly to the  $\epsilon$ -NSGAII on this application.

The  $\epsilon$ -NSGAII builds on its parent algorithm, the NSGAII [46], by adding  $\epsilon$ -dominance archiving [47,31], adaptive population sizing [48], and automatic termination to minimize the need for extensive parameter calibration as demonstrated by Reed et al. [12]. The concept of  $\epsilon$ -dominance allows the user to specify the precision with which they want to quantify each objective in a multi-objective problem. Fig. 1 demonstrates the concept of  $\epsilon$ -dominance using a three step approach for a two-objective minimization problem. First, a user specified  $\epsilon$  grid is applied to the search space of the problem. Larger  $\epsilon$  values result in a coarser grid (and ultimately fewer solutions) while smaller  $\epsilon$  values produce a finer grid. Grid blocks containing multiple solutions are then examined and only the solution closest to the bottom left corner of the block is kept (assuming minimization of all objectives). In the second step, non-domination sorting based on the grid blocks is then conducted resulting in a “thinning” of solutions (step 3) and promoting a more even search of the objective space. Epsilon-dominance allows the user to define objective precision requirements that make sense for their particular application. The interested reader can refer to prior work by Laumanns et al. [47] and Deb et al. [31] for a more detailed description of  $\epsilon$ -dominance.

The  $\epsilon$ -NSGAII uses a series of “connected runs” where small populations are initially exploited to pre-condition search and automatically adapt population size commensurate with problem difficulty. As the search progresses, the population size is automatically adapted based on the

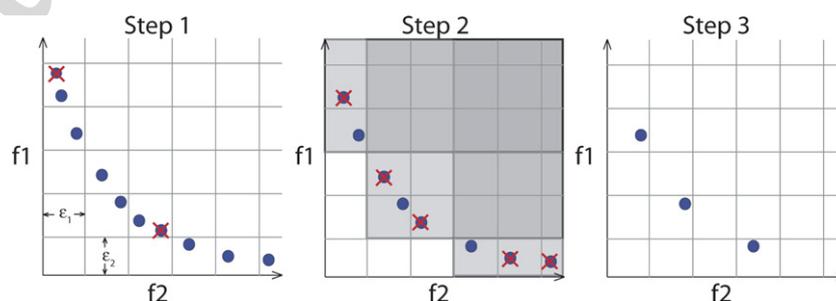


Fig. 1. Illustration of the  $\epsilon$ -dominance concept.

number of  $\epsilon$ -non-dominated solutions that the algorithm has found. Epsilon-non-dominated solutions found after each generation are stored in an archive and subsequently used to direct the search. Theoretically, this approach allows the MOEA's population size to increase or decrease, and in the limit when the  $\epsilon$ -dominance archive size stabilizes, the  $\epsilon$ -NSGAI's "connected runs" are equivalent to a diversity-based EA search enhancement recommended by Goldberg [49] termed "time continuation". The search is terminated across all runs (i.e., across all populations used) if the number and quality of solutions has not increased above a user specified  $\Delta$ -percent across two successive runs.

The primary goal in the development of the  $\epsilon$ -NSGAI was to provide a highly reliable and efficient MOEA which minimizes the need for traditional EA parametrization and allows the user to focus on problem specific search quality goals. Computational savings can be viewed in two contexts: (i) the use of minimal population sizes and (ii) the elimination of trial-and-error application runs to determine search parameters. Although the adaptation of population size will differ depending on the random seed chosen, exploiting small populations to precondition search will on average greatly reduce computation times. Readers interested in a more detailed description of the  $\epsilon$ -NSGAI can refer to the study by Kollat and Reed [1].

#### 4. Test cases used in scaling analysis

For the computational scaling analysis, test cases of varying size were derived from the full 29 well test case (described in Section 2.1) to demonstrate the effects of increasing problem sizes (defined as the number of sampling wells) on the computational demands of using the  $\epsilon$ -NSGAI to approximate the four-objective Pareto surface. To accomplish this, the least important wells were eliminated from the full 29 well test case based on previous results attained by Reed and Minsker [13] in order to develop a set of smaller test cases which could be enumerated within a reasonable time frame. The relative impor-

tance of wells was defined in this case by the well sampling frequency distribution associated with Reed and Minsker's best approximation to the true Pareto-front obtained using the original NSGAI. Based on preliminary enumeration analyses, it was then determined that a 25 well test case represented the upper bound problem size which could be enumerated in a reasonable time frame (approximately 6 days of continuous computing on a Pentium IV 3.0 GHz processor). The true four-objective Pareto-optimal solution set was then generated for the 25 well test case by evaluating all  $2^{25}$  (over 33.5-million) possible well sampling schemes in terms of the four design objectives defined previously in Section 2.2. For the enumeration, infeasible solutions (violating the constraint described by Eq. (2)) and inferior solutions (i.e., solutions dominated in terms of at least one design objective) were eliminated from consideration, resulting in the true Pareto-optimal solution set. Smaller test cases were then generated by eliminating individual wells from the 25 well test case based on the well sampling frequency distribution obtained from the 25 well enumeration. Test cases ranging from 18 to 25 wells were then developed based on this methodology and each test case was subsequently enumerated to obtain the true four-objective Pareto-optimal solution set for each test case. Knowing the Pareto-optimal set for each test case allows for rigorous assessment of the computational scaling of the  $\epsilon$ -NSGAI algorithm. A cross-sectional slice of the simulated PCE contamination plume is shown in Fig. 2 along with the 29 well sampling locations. The table associated with the figure indicates which wells were eliminated in order to create each of the 18–25 well test cases explored in this study.

#### 5. Computational experiment

The computational scaling characteristics of the  $\epsilon$ -NSGAI are tested in this study using the true Pareto-optimal solution sets for each of the 18–25 well test cases. Following enumeration, the  $\epsilon$ -NSGAI was used to

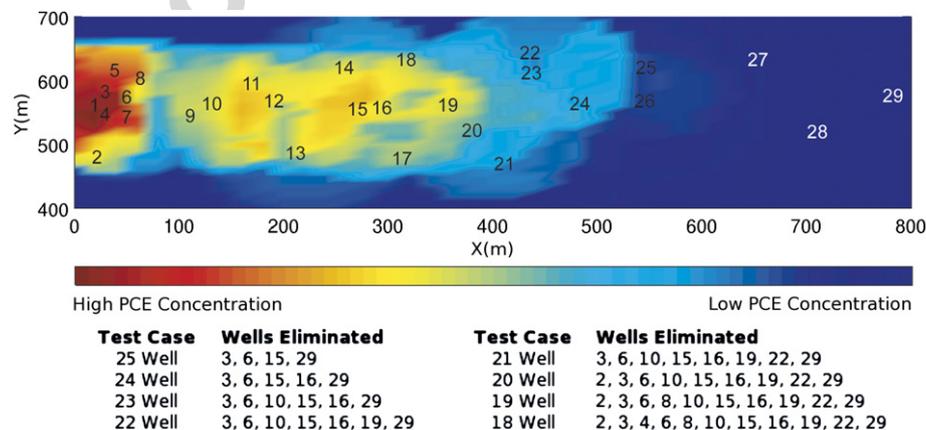


Fig. 2. Cross-sectional slice of the PCE contamination plume representing the LTM test case used in this study. There are 29 well locations available for sampling with one to three sampling locations available along each well's vertical axes. Wells eliminated to create each of the 18–25 well test cases are shown in the figure.

approximate these tradeoffs and its evolutionary operators were parameterized as follows: probability of cross-over –  $p_c = 1.0$ , probability of mutation –  $p_m = 1/N$  where  $N$  is the population size, cross-over distribution index –  $\eta_c = 15$ , and the mutation distribution index –  $\eta_m = 20$ . The  $\epsilon$ -NSGAI's adaptive population sizing was initialized using 10 individuals. Epsilon resolution settings for the four design objectives:  $\epsilon_{\text{cost}}$ ,  $\epsilon_{\text{conc}}$ ,  $\epsilon_{\text{uncert}}$ , and  $\epsilon_{\text{mass}}$  were set to 1.0, 0.0001, 0.0001, and 0.000001, respectively. These values represent the precision with which each objective is quantified and were chosen in this case to represent the full precision Pareto-optimal set. Since MOEA search is initialized with randomly generated populations and since evolutionary operators are probabilistic, the process can yield high variability in search efficiency and reliability. It is standard practice to overcome this variability by running EMO algorithms for a distribution of “seeds” for the random number generator which is used to initialize and guide their probabilistic search. In this study, our scaling analysis across the eight different LTM test cases was characterized using 50 random seed trial runs for each problem size (i.e., a total of  $8 \times 50 = 400$  trial runs).

Since MOEAs are stochastic search methods, the approach to the true Pareto-front of difficult problems usually occurs asymptotically in terms of the number of design evaluations needed to perfectly capture the optimal set of solutions. To ensure the computational tractability of this study, 80% of the true Pareto-optimal set was approximated for each of the problem sizes. In order to accurately quantify the percentage of Pareto-optimal solutions found, the  $\epsilon$ -performance metric [2] was used to determine the percentage of algorithm solutions found within a user specified  $\epsilon$  distance of the reference (i.e., Pareto-optimal) set, which was generated by enumerating all possible well sampling schemes. The first step in the calculation of the metric is to apply the  $\epsilon$ -dominance concept to the reference set according to user specified precision values (see Fig. 1). The proportion of solutions found within  $\epsilon$  hypercubes of the  $\epsilon$ -dominated reference set is then measured by matching solutions from the algorithm set to the reference set. Reference set solutions with a matching algorithm solution receive an indicator score while those with no matching solution receive no score (see Fig. 3 for an example calculation of the metric). Reference solutions with multiple matching approximation solutions use the solution which is closest in terms of Euclidean distance, allowing the additional solutions to be matched with other reference solutions which may have overlapping  $\epsilon$  hypercubes. The values of this metric range from zero to one, where a metric value of one indicates 100% convergence to within  $\epsilon$  of the reference set.

For this study, the  $\epsilon$ -performance metric was used to rigorously evaluate the algorithm's progress towards the true Pareto-front by greatly restricting the  $\epsilon$ -tolerance around the reference set solutions. Epsilon-tolerance values used by the  $\epsilon$ -performance metric for each of the four design objectives ( $\epsilon_{\text{cost}}$ ,  $\epsilon_{\text{conc}}$ ,  $\epsilon_{\text{uncert}}$ , and  $\epsilon_{\text{mass}}$ ) were set similarly

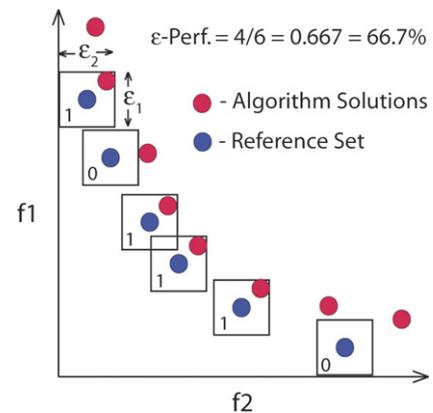


Fig. 3. Example calculation of the  $\epsilon$ -performance metric.

to the  $\epsilon$ -dominance settings (1.0, 0.0001, 0.0001, and 0.000001, respectively) as these values were chosen for a high level of objective precision. This means that if a solution was not found within these tolerance limits for each reference set solution, then it was not considered a true solution to the problem and was not included in the calculation of the metric. A target  $\epsilon$ -performance level of 80% convergence to within the specified  $\epsilon$  tolerance of the reference set (i.e., 80% of the reference set has been quantified to a very close precision) was then used as the sole basis for the runtime of the algorithm.

Although the LTM problem described is inherently suited to a binary 0/1 representation within the MOEA framework (i.e., individuals are represented as strings of ones and zeros to indicate whether or not a well is sampled), a real-coded representation was used in this study. This choice was based on previous study results which indicated superior performance of the algorithm when using a real-coded representation for this particular problem (the improved performance was attributed to the utilization of real-coded evolutionary operators). Therefore, the binary representation of the well sampling schemes was converted to a real-coded representation using variables ranging from 0.0 to 1.0. If the algorithm generated a variable less than 0.5, it was changed to 0.0 and variables greater than or equal to 0.5 were changed to 1.0.

## 6. Results

Enumeration of the 18–25 well test cases for the four design objectives revealed a linear relationship between problem size and the number of Pareto-optimal solutions ranging from 525 solutions for the 18 well test case, to 2439 solutions for the 25 well test case. Fig. 4 shows the number of Pareto-optimal solutions plotted versus problem size (in terms of the number of sampling wells). Annotations shown in the figure provide the exact number of Pareto-optimal solutions for each test case. In addition, the 0.98  $R$ -squared value confirms the linear trend.

Example visualizations of the four-objective Pareto fronts for both the 18 and the 25 well enumerated test cases

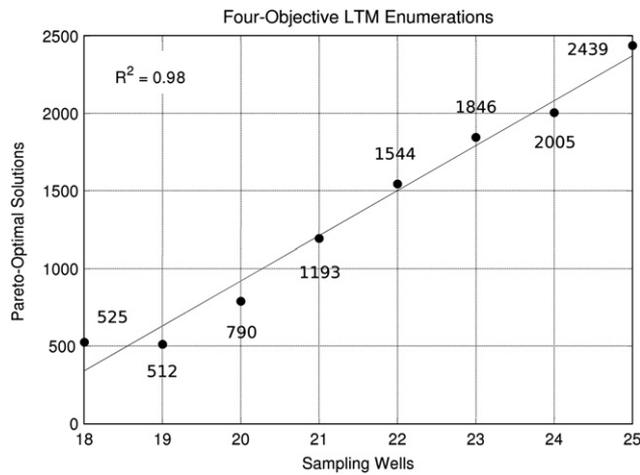


Fig. 4. Number of Pareto-optimal solutions versus problem size for each of the 18–25 well test cases.

are presented in Fig. 5. The sampling cost, concentration estimation error, and local uncertainty objectives are plotted on the  $x$ ,  $y$ , and  $z$  coordinate axes and the fourth objective, mass estimation error, is represented by the marker color. Fig. 5a and c show the actual Pareto-optimal surfaces (in this case a volume of points because there are four design objectives) for the 18 and 25 well test cases, respectively. Fig. 5b and d show planar projections of the Pareto-

optimal solution sets onto the planes formed by the  $x$ ,  $y$ , and  $z$  coordinate axes of the 18 and 25 well test cases respectively. Viewing the results using the projections shown in Fig. 5b and d provides a better representation of the tradeoffs between the design objectives. The interested reader is invited to explore the electronic version of this article which provides a full color illustration of Fig. 5. The enumeration of the 18–25 well test cases reveals a great deal regarding the scaling properties of the LTM problem. The main observation to be drawn from Fig. 5 is the strong geometric similarity between all of the Pareto-optimal solution sets (Fig. 5) illustrates this for both the 18 and 25 well test cases). This indicates that for each of the problem sizes used on our scaling analysis, we are solving problems with very similar structures.

Table 1 provides an overview of the total number of solutions in each of the problems' decision spaces, the total number of Pareto-optimal solutions, and the percentage of their decision spaces that were infeasible. The percentage of search space which is infeasible according to the objective formulation presented in Section 2.2 decreases (from 65.1% to 45.6%) as the number of wells is increased. For the 18 well test case, there are a total of 525 Pareto-optimal designs out of the 262,144 potential sampling schemes and for the 25 well test case, there are a total of 2439 Pareto-optimal designs out of the 33,554,432 potential sampling schemes.

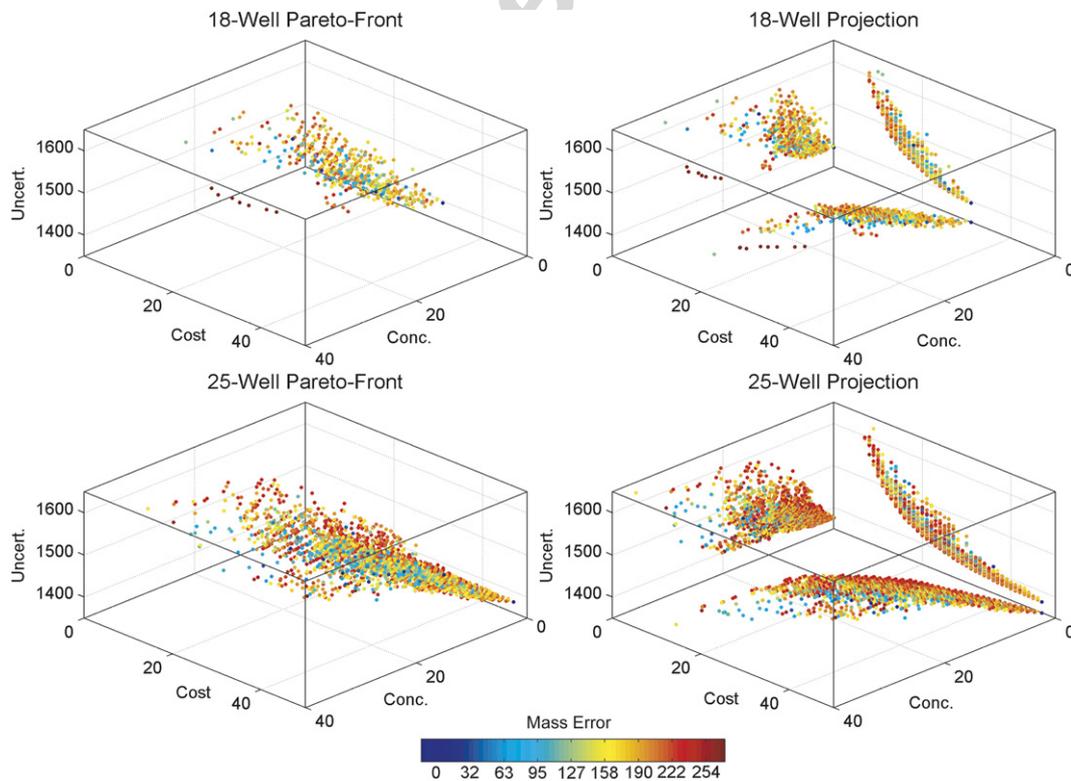


Fig. 5. Four objective Pareto fronts for the 18 and 25 well test cases. Plots A and C show the actual Pareto surface and plots B and D show the planar projections of the cost, concentration error, and uncertainty objectives onto the planes formed by the  $x$ ,  $y$ , and  $z$  coordinate axes. Mass error is represented by the color of the markers.

Table 1  
Enumeration data for the 18–25 well test cases

Wells	Total possible solutions	Pareto-optimal solutions	% Infeasible
18	262,144	525	65.1
19	524,288	512	64.0
20	1,048,576	790	55.8
21	2,097,152	1193	54.1
22	4,194,304	1544	51.4
23	8,388,608	1846	48.2
24	16,777,216	2005	47.8
25	33,554,432	2439	45.6

Data shown includes the number of possible solutions, the number of Pareto-optimal solutions, and the percentage of the search space which was infeasible.

Fig. 6 shows the average number of function evaluations required by the  $\epsilon$ -NSGAI to attain an  $\epsilon$ -performance measure of 80% versus problem size. The upper curve (composed of circular markers) indicates a quadratic growth rate between the  $\epsilon$ -NSGAI's computational requirements and the LTM problem size. The points on the curve represent an average of the total function evaluations required for 50 random seed trials and the error bars represent the 90th percentile range of random seed performance. In this plot, we can see that the computational cost of solving the LTM problem grows quadratically from approximately 34,000 function evaluations for the 18 well test case to over 590,000 function evaluations for the 25 well test case (on average). In addition, the variability in the computational demands posed by achieving the target performance level increases substantially with increasing problem size. The increasing variability in computational demands with larger problem sizes shows that random seed effects will have

a more severe impact on the reliability of MOEAs with increasing problem sizes. For instance, the range of performance of the 18 well case is approximately 16,900 function evaluations while the range of performance of the 25 well case is approximately 292,000 function evaluations. Both ranges represent approximately 50% of the mean, but for the 25 well test case, this level of reliability in terms of random seed performance has huge implications with regard to the computational requirements of the LTM problem (especially when these runs required approximately 4 h per random seed running on a Dell Pentium IV 3.0 GHz processor running Microsoft® Windows XP).

Epsilon-dominance archiving can be used to limit the quadratic growth of the LTM problem if users are willing to accept an increasingly coarser approximation to the Pareto front with increasing problem size. Since both the 18 and 19 well test cases contained approximately 500 Pareto-optimal solutions,  $\epsilon$ -dominance settings were manipulated for the 20–25 well test cases to result in Pareto-optimal sets containing approximately 500 solutions each. This results in a reduction of the Pareto-optimal set to a size similar to the 18 and 19 well test cases while at the same time, incorporating additional sampling wells. Epsilon-dominance settings used to scale the 20–25 well test cases are shown in Table 2. Readers should note that this analysis clearly shows that the computational demands of MOEAs are closely linked to the size of the Pareto-optimal set being approximated. Since the total number of evaluations used by an MOEA is a function of its population size and run duration, these results also enforce the importance of the adaptive population sizing used by the  $\epsilon$ -NSGAI.

Results achieved by the  $\epsilon$ -NSGAI through the  $\epsilon$ -dominance approximation of the 20–25 well test cases are shown in Fig. 6 (star markers). This figure demonstrates that the  $\epsilon$ -NSGAI's  $\epsilon$ -dominance archiving provides a mechanism for approximating the Pareto-optimal set, limiting its size, and potentially attaining a linear scaling of the computational cost with respect to problem size. Similarly to the quadratic scaling results, the linearly scaled approximation results are shown with the 90th percentile range of random seed performance indicated by error bars.

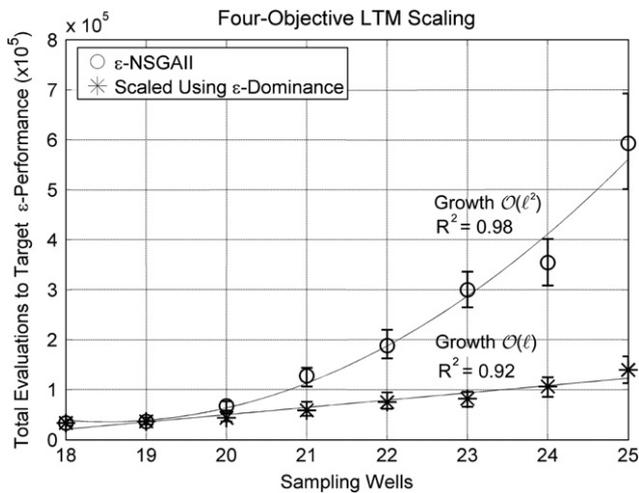


Fig. 6. Computational scaling results for the  $\epsilon$ -NSGAI applied to the 18–25 well LTM test cases. Also shown are the results attained using the  $\epsilon$ -dominance concept to approximate the Pareto set. The  $\epsilon$ -NSGAI's computational demands increase quadratically [ $O(l^2)$  where  $l$  is problem size] when solving successively larger LTM problems. However, using  $\epsilon$ -dominance to constrain the Pareto-set size to approximately 500 solutions results in a linear scaling [ $O(l)$ ] of computational demands with increasing problem size.

Table 2  
Epsilon settings used to approximate the Pareto-optimal sets of the 20–25 well test cases

Wells	$\epsilon_{cost}$	$\epsilon_{conc}$	$\epsilon_{uncert}$	$\epsilon_{mass}$	Sols.
18	1.0	0.0001	0.0001	0.000001	516
19	1.0	0.0001	0.0001	0.000001	498
20	1.0	0.4	0.7	0.1	503
21	1.0	0.9	1.4	0.5	503
22	1.0	1.0	1.5	0.5	499
23	1.0	1.1	1.7	0.8	505
24	1.0	1.0	1.5	0.8	500
25	1.0	1.1	1.7	0.8	499

The original  $\epsilon$  settings used in the first portion of the study for the 18 and 19 well test cases are shown as well.

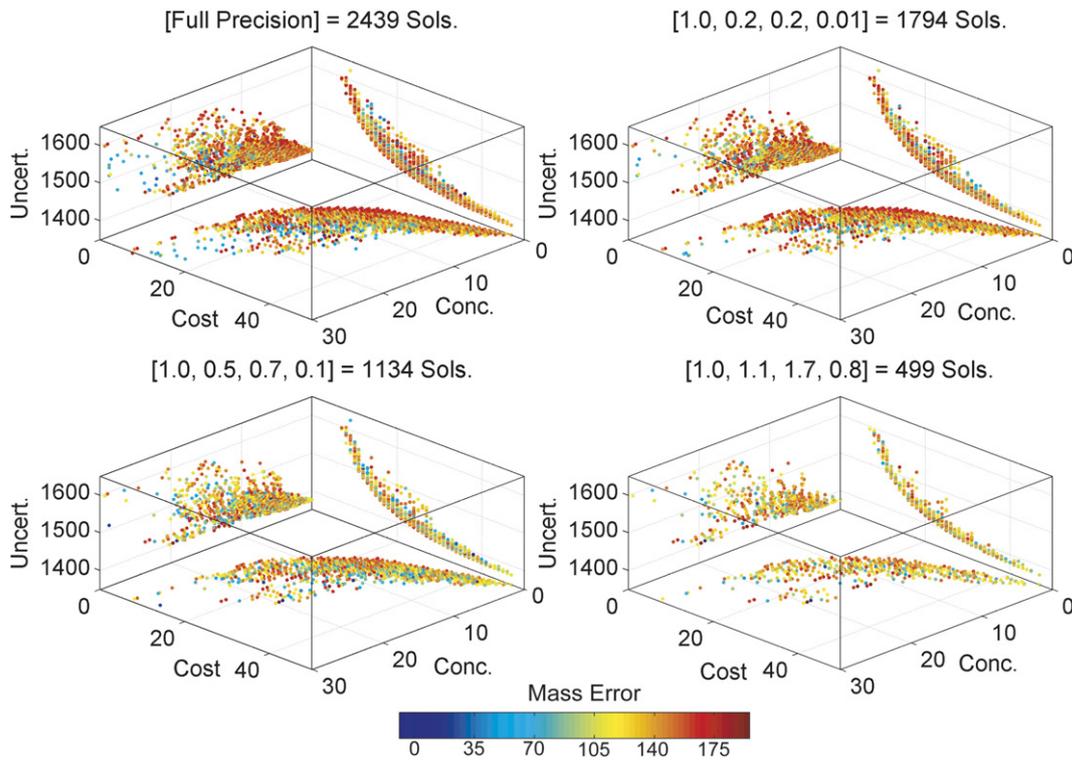


Fig. 7. Using  $\epsilon$ -dominance to control Pareto set size of the 25 well test case. Starting at the full Pareto-optimal set shown in plot A, we can see that by successively relaxing the precision requirements of the problem through the manipulation of the  $\epsilon$ -dominance parameters for each objective (shown in square brackets for plots B–D), the set size is reduced while maintaining geometric similarity to the full Pareto set.

Fig. 7 illustrates the application of  $\epsilon$ -dominance to the full Pareto-optimal set of the 25 well test case to yield successive reductions in the set size. Fig. 7a shows the planar projections of the full Pareto-optimal set for the 25 well test case shown in Fig. 5d. The cost, concentration error, and uncertainty objectives are represented by  $x$ ,  $y$ , and  $z$  coordinates and the mass error objective is represented by the color of the markers. Proceeding from Fig. 7a to b, we can see that applying  $\epsilon$  values of 1.0, 0.2, 0.2, and 0.01 to the cost, concentration error, uncertainty, and mass error objectives respectively reduces the Pareto-optimal set (through approximation) to 1794 solutions. Proceeding to Fig. 7c and d, we apply even more relaxed  $\epsilon$  settings to the objectives until finally the set is reduced to 499 solutions (as was used for the scaling analysis). Fig. 7 clearly shows that although the Pareto-optimal set size is reduced, the geometric properties of the set (including the extents of the objectives) are well preserved. This also highlights that using  $\epsilon$ -dominance archiving to approximate the Pareto-optimal solution set will still yield an excellent representation of the design tradeoff geometries to support decision making.

Fig. 8 illustrates the computational savings achieved by using  $\epsilon$ -dominance to control the size of the Pareto-optimal sets. Computational savings are expressed in terms of a percentage reduction in the average function evaluations required by the  $\epsilon$ -NSGAI (circle markers) and percentage reduction in random trial variability (star markers).

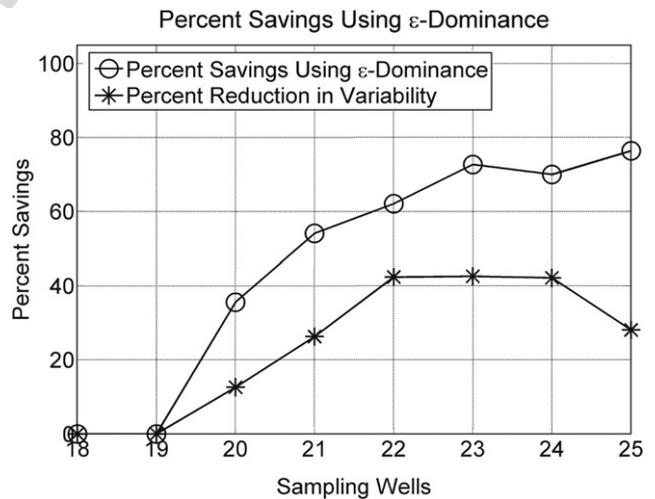


Fig. 8. Percent computational savings (circle markers) and percent reduction in variability (star markers) resulting from the use of  $\epsilon$ -dominance to control Pareto set size.

Random trial variability was measured in this study using the 90th percentile interquartile range of random trial performance. In other words, the best 5% and worst 5% of the random trials were eliminated from consideration when computing the range (to eliminate outliers). For the 25 well test case, the average computational cost decreases from over 593,000 function evaluations to generate the full Pareto-optimal set, to approximately 140,000 function

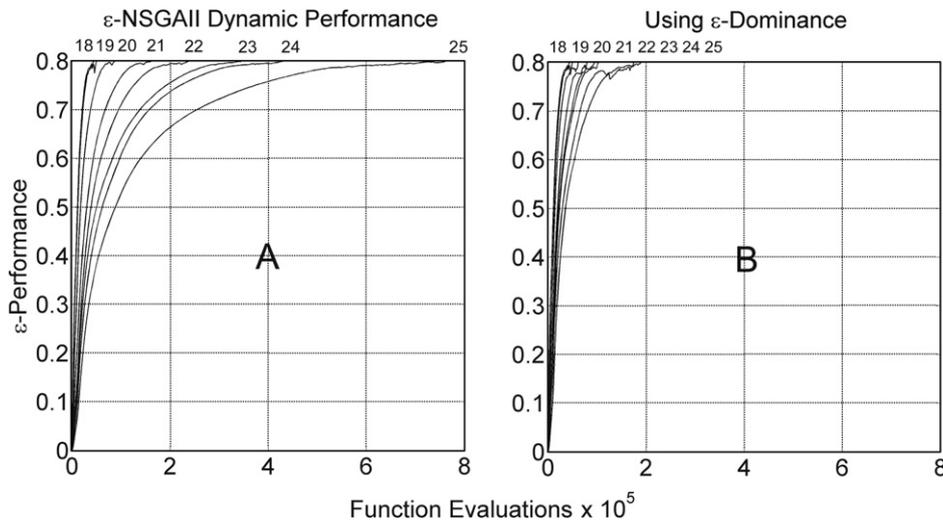


Fig. 9. Dynamic algorithm performance of the 18–25 well test cases. These plots show the  $\epsilon$ -performance of the 50 random seed trial runs for each test case plotted versus function evaluations. The mean performance of each test case is indicated by a solid line and test case correspondence is labeled above each plot. Plot A shows the dynamic performance of the  $\epsilon$ -NSGAI when searching for the full Pareto-optimal set and plot B shows the dynamic performance of the algorithm when  $\epsilon$ -dominance is used to approximate the set.

evaluations required to generate a 500 solution approximation to the full set. This represents a 76% decrease in the computational requirements of the 25 well test case when an approximation is accepted. Both Figs. 6 and 8 show that the reliability of the algorithm is greatly improved for increasing problem sizes (error bars in Fig. 6 and star markers in Fig. 8) when  $\epsilon$ -dominance is used for approximation. For example, the random seed trial variability (90th percentile) for the 25 well test case is reduced from over 191,000 function evaluations when generating the full Pareto-optimal set, to approximately 54,000 function evaluations when generating an approximation containing 500 solutions. This represents a 72% reduction (or improvement) in the range of random seed performance. In general, Fig. 8 shows that  $\epsilon$ -dominance approximation can provide increasing computational savings and improved reliability for increasing problem sizes.

Fig. 9 presents the  $\epsilon$ -NSGAI's runtime results for the  $\epsilon$ -performance metric versus total design evaluations for each of the 18–25 well test cases. The dynamic results shown in the figure represent the mean performance across the 50 random trials used to solve each test case. The test cases associated with each performance line are annotated above the plot. Fig. 9a shows the scaling dynamics which the  $\epsilon$ -NSGAI achieves when seeking the full Pareto-optimal set for each test case. This plot clearly shows the influence of problem size on search efficiency as the dynamics of each test case clearly differ from one another. Smaller test cases (namely the 18–21 well cases) approach the target level of performance ( $\epsilon$ -performance = 80%) very quickly. At approximately the 21 well test case and beyond, the algorithm's progress toward the target level of performance becomes increasingly more computationally demanding and reflects the quadratic growth in computational cost associated with increasing problem size as observed previ-

ously in Fig. 6. Fig. 9b presents the runtime dynamics of the  $\epsilon$ -NSGAI when  $\epsilon$ -dominance archiving is used to approximate the Pareto-optimal solution set. This plot presents the dynamics of the LTM test cases which were scaled according to the  $\epsilon$  settings presented in Table 2. In this figure, we can see that the search progress towards the target  $\epsilon$ -performance level of 80% is dramatically faster than when seeking the full Pareto-optimal set. In fact, if the dynamics achieved when approximating the Pareto set for all test cases in Fig. 9b were projected onto the dynamics achieved when searching for the full Pareto set shown in Fig. 9a, they would easily fit within the dynamics of the 18–22 well test cases.

## 7. Discussion

This study demonstrated the effects of increasing LTM problem size on the computational requirements of the  $\epsilon$ -NSGAI. The decision space of the LTM problem increases according to  $2^l$  where  $l$  represents the number of binary decisions (in this case, a yes/no decision of whether or not to sample from a predetermined well location). However, enumeration of the 18–25 well test cases revealed a linear relationship between problem size and the number of Pareto-optimal solutions. Test cases below 18 wells are easily tractable given current computational ability but enumeration of the 25 well test cases required 6 days of continuous computing on a Pentium IV, 3 GHz processor. Each time a well is added, the computational requirements of complete enumeration are doubled, ultimately motivating the need for efficient algorithms which are capable of generating the Pareto-optimal solution set (or an approximation thereof) without evaluating every possible solution. This study showed that the  $\epsilon$ -NSGAI was capable of generating a close approximation (to within

80% in terms of the  $\epsilon$ -performance metric) of the Pareto-optimal solution set for the 25 well test case in approximately 4.3 h (on average) as opposed to the 6-days required for enumeration. This represents a 97% reduction in computational requirements. Computational requirements were further reduced (76%) by accepting a 499 solution approximation to the true 2439 solution Pareto-optimal front through the utilization of  $\epsilon$ -dominance. The use of  $\epsilon$ -dominance revealed that through approximation, the computational complexity of using MOEAs to solve LTM problems could be reduced from quadratic to approximately linear scaling within the range of test cases examined. This has significant implications for the future of LTM design. For example, the largest test case examined in this study consisted of 25 decision variables occurring at a single point in time. If however, the site required quarterly sampling, the addition of a temporal component quickly increases the decision space to 100 variables or  $2^{100}$  potential sampling schemes. As it currently requires approximately 500,000 function evaluations to solve the 25 well test case, the introduction of quarterly sampling represents a 16-fold increase in the number of function evaluations (i.e., 8-million) because of the  $\epsilon$ -NSGAI's quadratic scaling. In addition, if a space-time evaluation scheme is used, the computational requirements of each function evaluation will be greatly increased from that required by Quantile Kriging as was used in this study.

The LTM test cases analyzed in this study represent a lower bound in terms of the computational complexity of using MOEAs in water resources applications. More complex water resources applications with tens or hundreds of integer or continuous decision variables may have a more severe growth rate for their Pareto-optimal set sizes and computational demands. In this study,  $\epsilon$ -dominance was explored as a method of controlling the computational scaling of the LTM problem as the number of monitoring wells was increased. This method provides a means of approximating the Pareto-optimal set based on user defined precision goals and a willingness to accept the approximation. However, as problem size increases, the severity of the approximation increases as well. Although optimization algorithms which are capable of attaining the true Pareto-front are the ideal goal, this study demonstrates that in reality, water resources scientists and engineers will have to accept approximations to their applications' Pareto-sets and advance the field by investigating innovative new algorithms that scale subquadratically.

## 8. Conclusions

This study contributed a detailed assessment of how increasing problem sizes (measured in terms of the number of decision variables being considered) impacted the computational complexity of using the  $\epsilon$ -NSGAI to solve an LTM application. LTM test cases composed of 18–25 sampling wells were first enumerated in terms of four design objectives. The  $\epsilon$ -NSGAI was then used to approximate

the Pareto-optimal solution set of each test case to an  $\epsilon$ -performance level of 80%. Results of the study indicated linear scaling of Pareto set size versus problem size. The  $\epsilon$ -NSGAI required quadratic scaling but the incorporation of  $\epsilon$ -dominance to approximate the Pareto-optimal set resulted in approximately linear computational scaling. Although MOEAs are capable of solving challenging water resources applications, the consideration of larger problems will require users to accept approximations to their Pareto-optimal sets as well as research that will develop improved algorithms that are capable of scaling subquadratically.

## References

- [1] Kollat JB, Reed PM. Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design. *Adv Water Resour* 2006;29(6):792–807.
- [2] Kollat JB, Reed PM. The value of online adaptive search: a performance comparison of NSGA-II,  $\epsilon$ -NSGAI, and  $\epsilon$ MOEA. In: Coello CC, Aguirre AH, Zitzler E, editors. *The third international conference on evolutionary multi-criterion optimization (EMO 2005)*. Lecture notes in computer science, 3410. Guanajuato, Mexico: Springer Verlag; 2005. p. 386–98.
- [3] Farmani R, Savic DA, Walters GA. Evolutionary multi-objective optimization in water distribution network design. *Eng Optimiz* 2005;37(2):167–83.
- [4] Task Committee on Long-Term Groundwater Monitoring Design. *Long-term groundwater monitoring: the state of the art*. Reston, VA: American Society of Civil Engineers; 2003.
- [5] Knopman DS, Voss CI. Multiobjective sampling design for parameter estimation and model discrimination in groundwater solute transport. *Water Resour Res* 1989;25(10):2245–58.
- [6] Meyer PD, Valocchi AJ, Eheart JW. Monitoring network design to provide initial detection of groundwater contamination. *Water Resour Res* 1994;30(9):2647–59.
- [7] James BR, Gorelick SM. When enough is enough: the worth of monitoring data in aquifer remediation design. *Water Resour Res* 1994;30(12):3499–513.
- [8] Sun N. *Inverse problems in groundwater modeling. Theory and applications of transport in porous media*, vol. 6. New York, NY: Kluwer Academic Publishers; 1994.
- [9] Cieniawski SE, Eheart JW, Ranjithan SR. Using genetic algorithms to solve a multiobjective groundwater monitoring problem. *Water Resour Res* 1995;31(2):399–409.
- [10] Wagner BJ. Sampling design methods for groundwater modeling under uncertainty. *Water Resour Res* 1995;31(10):2581–91.
- [11] Storck P, Eheart JW, Valocchi AJ. A method for the optimal location of monitoring wells for detection of groundwater contamination in three-dimensional aquifers. *Water Resour Res* 1997;33(9):2081–8.
- [12] Reed P, Minsker BS, Goldberg DE. A multiobjective approach to cost effective long-term groundwater monitoring using an elitist nondominated sorted genetic algorithm with historical data. *J Hydroinform* 2001;3(2):71–90.
- [13] Reed P, Minsker BS. Striking the balance: long-term groundwater monitoring design for conflicting objectives. *J Water Resour Plan Manage* 2004;130(2):140–9.
- [14] Lettenmaier DP. Dimensionality problems in water quality network design. *Water Resour Res* 1979;13(6):1692–700.
- [15] Knopman DS, Voss CI, Garabedian SP. Sampling design for groundwater solute transport: tests methods and analysis of cape cod tracer test data. *Water Resour Res* 1991;27(5):925–49.
- [16] Fleming PJ, Purshouse RC, Lygoe RJ. Many-objective optimization: an engineering design perspective. In: Coello CC, Hernandez A, Zitzler E (Eds.), *Evolutionary multi-criterion optimization*, Springer

- lecture notes in computer science, vol. 3410, Guanajuato, Mexico, 2005. p. 14–32.
- [17] Farina M, Amato P. On the optimal solution definition for many-criteria optimization problems. In: Keller J, Nasraoui O. (Eds.), *Proceedings of the 2002 NAFIPS-FLINT international conference*, IEEE Service Center, Piscataway, New Jersey, 2002. p. 233–8.
- [18] Pareto V. *Cours D'Economie Politique*, vols. 1 and 2, Rouge, Lausanne, 1896.
- [19] Back T, Fogel D, Michalewicz Z. *Handbook of evolutionary computation*, Bristol, UK, 2000.
- [20] Salomon R. Evolutionary algorithms and gradient search: similarities and differences. *IEEE Trans Evol Comput* 1998;2(2):45–55.
- [21] Goldberg DE. *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley Publishing Company; 1989.
- [22] Keeney RL, Raiffa H. *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge, UK: Cambridge University Press; 1993.
- [23] Chankong V, Haimes Y. *Multiobjective decision making: theory and methodology*. North-Holland series in system science and engineering. New York, NY: North-Holland; 1983.
- [24] Haimes Y. *Risk modeling, assessment, and management*. Wiley series in systems engineering. New York, NY: John Wiley & Sons, Inc.; 1998.
- [25] Coello CC, Van Veldhuizen DA, Lamont GB. *Evolutionary algorithms for solving multi-objective problems*. New York, NY: Kluwer Academic Publishers; 2002.
- [26] Ritzel BJ, Eheart JW, Ranjithan SR. Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. *Water Resour Res* 1994;30(5):1589–603.
- [27] Horn J, Nafpliotis F. Multiobjective optimization using the niched Pareto genetic algorithm. Tech rep IlliGAL report no. 93005, University of Illinois, 1993.
- [28] Halhal D, Walters GA, Ouazar D, Savic DA. Water network rehabilitation with structured messy genetic algorithm. *J Water Resour Plan Manage* 1997;123(3):137–46.
- [29] Loughlin DH, Ranjithan SR, Baugh Jr JW, Brill Jr ED. Application of genetic algorithms for the design of ozone control strategies. *J Air Waste Manage Assoc* 2000;50:1050–63.
- [30] Erickson MA, Mayer A, Horn J. Multi-objective optimal design of groundwater remediation systems: application of the niched Pareto genetic algorithm (NPGA). *Adv Water Resour* 2002;25(1):51–6.
- [31] Deb K, Mohan M, Mishra S. A fast multi-objective evolutionary algorithm for finding well-spread Pareto-optimal solutions. Tech rep KanGAL report no. 2003002, Indian Institute of Technology, 2003.
- [32] Deb K. *Multi-objective optimization using evolutionary algorithms*. New York, NY: John Wiley & Sons LTD; 2001.
- [33] Kumar SV, Ranjithan SR. Evaluation of the constraint method-based multiobjective evolutionary algorithm (CMEA) for a three-objective optimization problem. In: Cantu-Paz E, Mathias K, Roy R, Davis D, Poli R, Balakrishnan K, et al., editors. *Proceedings of the genetic and evolutionary computation conference (GECCO 2002)*. New York, NY: Morgan Kaufmann; 2002. p. 431–8.
- [34] Muleta M, Nicklow J. Decision support for watershed management using evolutionary algorithms. *J Water Resour Plan Manage* 2005;131(1):35–44.
- [35] Farmani R, Walters GA, Savic DA. Trade-off between total cost and reliability for anytown water distribution network. *J Water Resour Plan Manage* 2005;131(5):161–71.
- [36] Ren X, Minsker BS. Which groundwater remediation objective is better: a realistic one or a simple one? *J Water Resour Plan Manage* 2005;131(5):351–61.
- [37] McPhee J, Yeh WG. Multiobjective optimization for sustainable groundwater management in semiarid regions. *J Water Resour Plan Manage* 2004;130(6):490–7.
- [38] Labadie J. Optimal operation of multireservoir systems: state-of-the-art review. *J Water Resour Plan Manage* 2004;130(2):93–111.
- [39] Vrugt J, Gupta HV, Bastidas LA, Bouten W, Sorooshian S. Effective and efficient algorithm for multiobjective optimization of hydrologic models. *Water Resour Res* 2003;39(8):1214. doi:10.1029/2002WR001746.
- [40] Maxwell R, Carle FS, Tompson FB. Contamination, risk, and heterogeneity: on the effectiveness of aquifer remediation. Tech rep UCRL-JC-139664, Lawrence Livermore National Laboratory, 2000.
- [41] Reed P, Ellsworth T, Minsker BS. Spatial interpolation methods for nonstationary plume data. *Ground Water* 2004;42(2):190–202.
- [42] Deutsch CV, Journel AG. *GSLIB: geostatistical software library and user's guide*. New York, NY: Oxford University Press; 1998.
- [43] Cooper RM, Istok JD. Geostatistics applied to groundwater contamination. I: Methodology. *J Environ Eng* 1988;114(2):270–86.
- [44] Cooper RM, Istok JD. Geostatistics applied to groundwater contamination. II: Application. *J Environ Eng* 1988;114(2):287–99.
- [45] Deb K, Agrawal RB. Simulated binary crossover for continuous search space. Tech rep IITK/ME/SMD-94027, Indian Institute of Technology, Kanpur, 1994.
- [46] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 2002;6(2):182–97.
- [47] Laumanns M, Thiele L, Deb K, Zitzler E. Combining convergence and diversity in evolutionary multiobjective optimization. *Evol Comput* 2002;10(3):263–82.
- [48] Harik GR, Lobo FG. A parameter-less genetic algorithm. Tech rep IlliGAL 99009, University of Illinois at Urbana-Champaign, 1999.
- [49] Goldberg DE. *The design of innovation: lessons from and for competent genetic algorithms*. Norwell, MA: Kluwer Academic Publishers; 2002.