Available online at www.sciencedirect.com
SCIENCE DIRECT®
Advances in Water Resources 29 (2006) 792–807
Advances in Water Resources
www.elsevier.com/locate/advwatres

Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design

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 13 April 2005; received in revised form 18 July 2005; accepted 19 July 2005
Available online 8 September 2005

Abstract

This study compares the performances of four state-of-the-art evolutionary multi-objective optimization (EMO) algorithms: the Non-Dominated Sorted Genetic Algorithm II (NSGAII), the Epsilon-Dominance Non-Dominated Sorted Genetic Algorithm II (ε-NSGAII), the Epsilon-Dominance Multi-Objective Evolutionary Algorithm (εMOEA), and the Strength Pareto Evolutionary Algorithm 2 (SPEA2), on a four-objective long-term groundwater monitoring (LTM) design test case. The LTM test case objectives include: (i) minimize sampling cost, (ii) minimize contaminant concentration estimation error, (iii) minimize contaminant concentration estimation uncertainty, and (iv) minimize contaminant mass estimation error. The 25-well LTM design problem was enumerated to provide the true Pareto-optimal solution set to facilitate rigorous testing of the EMO algorithms. The performances of the four algorithms are assessed and compared using three runtime performance metrics (convergence, diversity, and ε-performance), two unary metrics (the hypervolume indicator and unary ε-indicator) and the first-order empirical attainment function. Results of the analyses indicate that the ε-NSGAII greatly exceeds the performance of the NSGAII and the εMOEA. The ε-NSGAII also achieves superior performance relative to the SPEA2 in terms of search effectiveness and efficiency. In addition, the ε-NSGAII's simplified parameterization and its ability to adaptively size its population and automatically terminate results in an algorithm which is efficient, reliable, and easy-to-use for water resources applications.

Keywords: Long-term groundwater monitoring; Evolutionary algorithms; Multi-objective optimization; Performance metrics

1. Introduction

This study demonstrates the effectiveness of a modified version of Deb's Non-Dominated Sorted Genetic Algorithm II (NSGAII) [1], which the authors have named the Epsilon-Dominance Non-Dominated Sorted Genetic Algorithm II (ε-NSGAII) [1–4], at solving a four-objective long-term groundwater monitoring design test case. The ε-NSGAII incorporates prior competent evolutionary algorithm (EA) design concepts [2] and epsilon-dominance archiving [5] to improve the original NSGAII's efficiency, reliability, and ease-of-use. This algorithm eliminates much of the traditional trial-and-error parameterization associated with evolutionary multi-objective optimization (EMO) through epsilon-dominance archiving [5,6], dynamic population sizing [7], and automatic termination. The effectiveness and reliability of the new algorithm is compared to the original NSGAII [1] as well as two other benchmark multi-objective evolutionary algorithms (MOEAs), the Epsilon-Dominance Multi-Objective Evolutionary Algorithm (εMOEA) [6] and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [8]. Each of the MOEAs selected have been demonstrated to be highly effective

* Corresponding author.
E-mail addresses: juk124@psu.edu (J.B. Kollat), pmr11@psu.edu, preed@engr.psu.edu (P.M. Reed).

0309-1708/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advwatres.2005.07.010

at solving numerous multi-objective test problems and applications [1,4,6,8–11].

When decision making in water resources problems is characterized by conflicting objectives, optimality must be defined in the context of the application's tradeoffs. More formally, these tradeoffs characterize the Pareto front [12], which is composed of the set of solutions where improvement in one objective degrades performance in at least one other objective [13]. For instance, when optimizing well placement in long-term groundwater monitoring (LTM) design, minimizing both sampling uncertainty and sampling costs results in a tradeoff in which a design's uncertainty can only be decreased if the decision maker is willing to increase costs by sampling from more locations. The determination of the Pareto-optimal solutions for multiple conflicting objectives is referred to as multi-objective optimization [13] and the determination of these solutions can aid in the design of LTM and other water resources systems. MOEAs, which evolve solutions through a process analogous to Darwinian natural selection [14], have become an increasingly popular optimization technique in recent years. Since many water resources problems are characterized by multiple conflicting objectives and huge decision spaces, the challenge of developing MOEAs that can efficiently search these spaces and provide a sufficient approximation to the true tradeoffs is a problem of significant importance.

This paper will demonstrate the ε-NSGAII's performance using a four-objective LTM application. Formally, LTM can be defined as the sampling of groundwater quality over long time-scales to provide "sufficient and appropriate information" to assess if current mitigation or contaminant control measures are performing adequately to be protective of human and ecological health [15]. The LTM problem has garnered significant interest within the water resources community over the past decade due to the tremendous expense and complexity of characterizing groundwater contamination sites over long time periods (for reviews see [15–17]). For example, a recent LTM task committee report [15] highlights that projected federal expenditures on LTM for the decade beginning in the year 2000 will be more than five-billion US dollars. Prior work has demonstrated that LTM design is an extremely challenging optimization problem with multiple conflicting objectives and very large discrete decision spaces [18–27].

Schaffer [28] developed one of the first EMO algorithms termed the vector evaluated genetic algorithm (VEGA), which was designed to search decision spaces for the optimal tradeoffs among a vector of objectives. Subsequent innovations in EMO have resulted in a rapidly growing field with a variety of solution methods that have been used successfully in a wide range of applications (as reviewed by [13,29–31]). These solution methods have garnered increased attention over the past decade and have been applied successfully in a variety of water resources and environmental applications [18–20,32–36]. More recently, Muleta and Nicklow [37] utilized MOEAs as decision support tools for management of non-point source pollution in watersheds. Keedwell and Khu [38] recently highlighted the ability of MOEAs to aid in optimal design of water distribution networks. Farmani et al. [39] have conducted a comparative study of the performance of several MOEAs at providing optimal tradeoffs for water distribution network design. Many additional applications exist in the literature with these being the most recent, indicating that MOEAs are important decision support tools for many aspects of water resources science and engineering.

The MOEA comparison study presented in this paper proceeds as follows. Section 2 details the underlying methodology of this study and describes the LTM test case. Section 3 details each of the algorithms which are compared. The metrics used to assess MOEA performance are described in Section 4. Section 5 provides a detailed description of the study's computational experiment and the parameterization of each algorithm. Section 6 presents the results of the study using runtime visualizations of the performances of each algorithm and additional end-of-run performance metric results intended to provide rigorous algorithm assessments. Section 7 presents a discussion regarding each algorithm's suitability to the LTM test case and the implications of this study for other water resources applications. Conclusions of the study are presented in Section 8.

2. Methodology

2.1. Multi-objective search and optimization in LTM design

The goal of multi-objective optimization is to identify the Pareto-optimal tradeoffs between an application's design 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 [12]. The Pareto-optimal front is obtained by plotting these solutions according to their objective values yielding an $M-1$ dimensional surface where M is equal to the total number of objectives. The term high-order Pareto surfaces is used to describe those surfaces that result from three or more conflicting objectives. Reed and Minsker [20] recently demonstrated high-order Pareto-optimization on a LTM test case containing 29 wells for four-objectives using quantile kriging and the NSGAII. The algorithm successfully identified 1,156 non-dominated designs approximating the Pareto-optimal set using 450,000 design evaluations out of a total decision space consisting of over 500 million possible designs. Ultimately, Reed

and Minsker [20] demonstrated how to exploit their approximation to the Pareto front to identify one compromise solution. This application demonstrated that MOEAs are capable of high-order Pareto optimization of water resources problems characterized by three or more objectives. The reader should note however that in this prior study, the true Pareto-optimal tradeoffs were not known, making it difficult to assess true algorithm performance.

2.2. 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. PCE concentration data are provided in mg m^{-3} at 47 hypothetical sampling locations in a 25 well monitoring network for a snapshot in time eight years following the initial release of contaminant. Each well can be sampled from one to three times along its vertical axis. 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 (see Fig. 1). The test case contaminant concentration data are both highly skewed and highly variable with well sampling locations clustered in an ad hoc manner within the body and at the source of the contaminant plume, making this test case representative of many real-world sites (see Reed et al. [41] for additional details).

In order to accurately assess the performance of each MOEA at solving the LTM test case, the true four-objective Pareto-optimal solution set was enumerated for the 25 well LTM test case. In this study, it is assumed that if a well is sampled, then all locations along its vertical axis are sampled, which as a result limits the decision space of the problem to 2^{25} (over 33-million) possible sampling schemes. The size of the test case was selected to limit the time required to enumerate all possible solutions to six days of continuous computing on a Pentium IV 3.0 GHz processor running Microsoft® Windows XP. The reader should note that the test case was generated by eliminating the four least important wells from the larger 29 well case analyzed previously by Reed and Minsker [20]. A geostatistical analysis of the 25 well subset revealed no significant changes in the variogram structure. The true four-objective Pareto front for the 25 well LTM test case is presented in Fig. 2. The sampling cost, concentration estimation error, and local uncertainty objectives (which will be explained in greater detail in Section 2.4) are represented by the x-, y-, and z-coordinates and the fourth objective, mass estimation error, is represented by the color of the

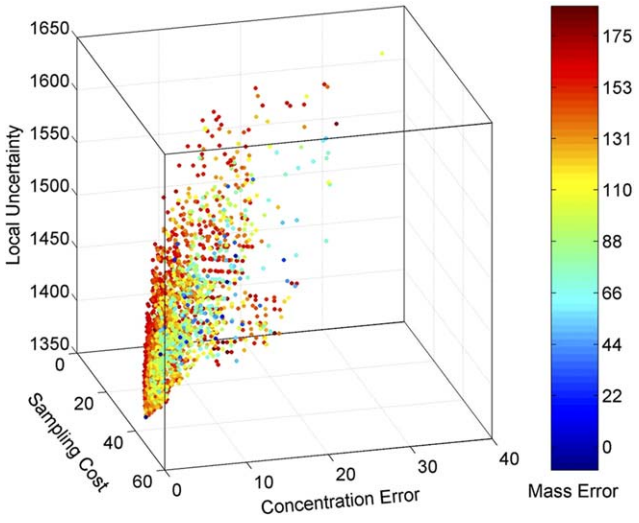

Fig. 2. Visualization of the true four-objective Pareto front for the 25 well LTM test case. The sampling cost, concentration estimation error, and concentration estimation uncertainty objectives are represented by the x-, y-, and z-coordinates and the contaminant mass estimation error objective is represented by marker color.

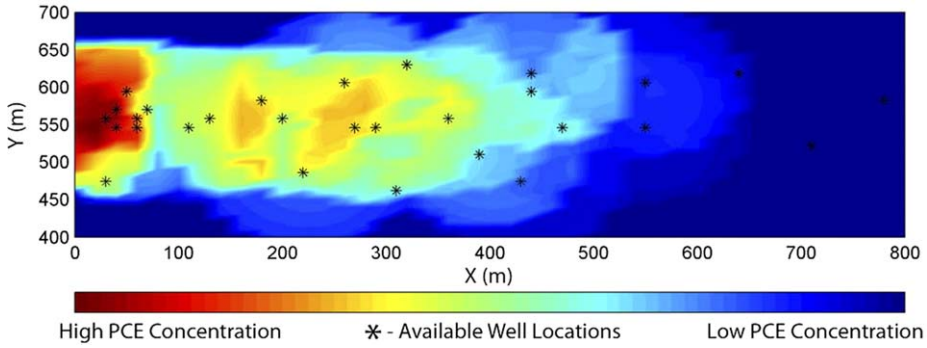

Fig. 1. A representative cross-sectional slice of the simulated PCE contamination plume used in this study. There are 25 well locations available for sampling (represented by stars) and each well has one to three sampling locations along its vertical axes.

markers. The interested reader is invited to explore the electronic version of this article which provides a full color illustration of Fig. 2 as well as a movie which illustrates the rotation of the figure. There are a total of 2439 Pareto optimal designs for the 25 well LTM test case and out of the 33,554,432 potential sampling schemes, 45.6% of them are infeasible based on the objective formulation presented in Section 2.4.

2.3. Spatial interpolation

Spatial interpolation of the contamination plume was conducted using quantile kriging 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]. Quantile kriging extends ordinary kriging (OK) by transforming the sample values to quantile space (or rank transform space) using Eq. (1), where m_i is the rank of each sample value i and N is the number of samples.

$$c(\bar{x}) = \frac{m_i}{N+1} \tag{1}$$

The quantile values represent the probability that a sample is less than or equal to its value, or more commonly, the empirical cumulative distribution function (CDF), 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 search neighborhoods are used to estimate the expected value at each location [42]. Reed et al. [41] found that quantile kriging 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 three-dimensional kriging library written in Fortran as part of the GSLIB software package [42]. A spherical variogram structure with nugget = 0.005 and range = 18 m was used. The interpolation grid was defined by 34 blocks in x, 7 blocks in y, and 7 blocks in z, resulting in 1666 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 of a point, ensuring that clustered data points did not bias interpolation estimates. For a more thorough description of each of the kriging parameters used in this study, the interested reader should refer to the books by Deutsch and Journel [42] and Goovaerts [45] as well as the study conducted by Reed et al. [41].

2.4. Objective formulation

Four-objectives are to be minimized for the LTM test case described in Section 2.2: (i) sampling cost, (ii) relative error of local contaminant concentration estimates, (iii) local contaminant concentration estimation uncertainty, and (iv) contaminant mass estimation error. Eq. (2) represents the objective formulation where $\mathbf{F}(\mathbf{x}_\kappa)$ is a vector valued performance function in which the four-objectives: cost, concentration error, local uncertainty, and mass estimation error are minimized. Eq. (3) subjects $\mathbf{F}(\mathbf{x}_\kappa)$ to the constraint that the number of unestimated points, $U(\mathbf{x}_\kappa)$, in the interpolation domain is zero. This means that if quantile kriging of a particular sampling plan results in any of the 1666 grid points being unestimated, then that particular sampling plan is considered infeasible.

Minimize

$$\mathbf{F}(\mathbf{x}_\kappa) = (f_{\text{cost}}(\mathbf{x}_\kappa), f_{\text{conc}}(\mathbf{x}_\kappa), f_{\text{uncert}}(\mathbf{x}_\kappa), f_{\text{mass}}(\mathbf{x}_\kappa)), \quad \forall \kappa \in \mathbf{\Omega} \tag{2}$$

$$\text{Subject to} \quad U(\mathbf{x}_\kappa) = 0 \tag{3}$$

The objectives are all a function of the vector $\mathbf{x}_\kappa$ representing the κth sampling plan in the decision space Ω. Each component i of a sampling plan κ is determined from Eq. (4) which results in a string of binary digits indicating whether or not a well is sampled.

$$x_{\kappa,i} = \begin{cases} 1, & \text{if the } i\text{th well is sampled} \\ 0, & \text{otherwise} \end{cases} \quad \forall \kappa, i \tag{4}$$

The sampling cost objective quantifies the monitoring cost of a particular sampling scheme using Eq. (5). The coefficient C_S defines the cost per sample (normalized to one in this study). As described in Section 2.2, the maximum number of monitoring wells, n_{wells}, that can be sampled is 25. Since the monitoring wells have a range of one to three potential sampling points on their vertical axis, the sampling cost coefficients range from 1 to 3. Additionally, if a well is sampled, it is assumed that all locations along its vertical axis are sampled. The cost objective is quantified by summing the cost coefficients of each of the wells in a particular scheme resulting in a normalized cost ranging from 0 to 47 for the 25 well test case.

$$f_{\text{cost}}(\mathbf{x}_\kappa) = \sum_{i=1}^{n_{\text{well}}} C_S(i) x_{\kappa,i} \tag{5}$$

The relative error of local contaminant concentration estimates objective measures how the kriged picture of the plume using the κth sampling plan differs from that obtained by sampling all well locations. Eq. (6) quantifies the concentration error objective by summing the squared differences between the concentration estimates at each of the $n_{\text{est}} = 1666$ grid locations $\mathbf{u}_j$ obtained utilizing all available well sampling locations, $c_{\text{all}}(\mathbf{u}_j)$, and

the concentration estimates at each of the 1666 grid locations obtained using the κth sampling plan, $c_\kappa(\mathbf{u}_j)$.

$$f_{\text{conc}}(\mathbf{x}_\kappa) = \sum_{j=1}^{n_{\text{est}}} \left(c_{\text{all}}(\mathbf{u}_j) - c_\kappa(\mathbf{u}_j)\right)^2 \tag{6}$$

Local contaminant concentration estimation uncertainty is quantified by summing the estimation standard deviations obtained from kriging at each of the $n_{\text{est}} = 1666$ grid locations $\mathbf{u}_j$ using Eq. (7). 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.

$$f_{\text{uncert}}(\mathbf{x}_\kappa) = \sum_{j=1}^{n_{\text{est}}} A_j \sigma(\mathbf{u}_j) \tag{7}$$

The contaminant mass estimation error objective quantifies the relative error between the total mass of dissolved contaminant estimated using all well locations, Mass_{all}, and the contaminant mass estimated from the κth sampling plan, Mass_κ. The contaminant data at each sampling location were defined in terms of dissolved mass of contaminant per volume aquifer to ensure additivity. All mass estimates were computed as integrals representing the zeroth spatial moment of the contaminant plume. Eq. (8) expresses the relative mass estimation error in terms of a percentage.

$$f_{\text{mass}}(\mathbf{x}_\kappa) = \left|\frac{\text{Mass}_{\text{all}} - \text{Mass}_\kappa}{\text{Mass}_{\text{all}}}\right| \cdot 100\% \tag{8}$$

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 unestimated points in the interpolation domain (violating the constraint described by Eq. (3)). In this case, the objectives are penalized to ensure that infeasible sampling schemes are eliminated from consideration. Eq. (9) 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) which greatly reduces their chances of surviving the evolutionary process.

$$\mathbf{F}_{\text{penalty}}(\mathbf{x}_\kappa) = \begin{cases} f_{\text{cost}}^{\text{penalty}} = f_{\text{cost}} + f_{\text{cost}}^{\max} \\ f_{\text{conc}}^{\text{penalty}} = f_{\text{conc}} + n_{\text{est}} + U(\mathbf{x}_\kappa) + f_{\text{cost}}^{\max} \\ f_{\text{uncert}}^{\text{penalty}} = f_{\text{uncert}} + n_{\text{est}} + U(\mathbf{x}_\kappa) + f_{\text{cost}}^{\max} \\ f_{\text{mass}}^{\text{penalty}} = f_{\text{mass}} + n_{\text{est}} + U(\mathbf{x}_\kappa) + f_{\text{cost}}^{\max} \end{cases} \tag{9}$$

Eq. (9) penalizes the objective functions based on the maximum cost of a sampling scheme, in this case 47 (based on all available sampling locations), the total number of estimation points in the grid, in this case 1666 (based on the sizing of the interpolation grid), and the total number of unestimated points, $U(\mathbf{x}_\kappa)$, in the infeasible sampling plan. For example, if a particular sampling plan results 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. (9) is defined so that all infeasible solutions will have costs that exceed the maximum feasible cost (i.e., 47). The exact ranges of the other objectives were not known a priori, so 1723 is a conservative penalty for the concentration error, uncertainty, and mass error objectives ensuring 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 further evolve into feasible designs.

3. Algorithm comparison

In this study, the performances of the NSGAII [1], the ε-NSGAII [1–4], the εMOEA [6], and the SPEA2 [46,47] are compared using the true Pareto optimal solution set of the four-objective LTM test case. All of the algorithms share similarities in that they use real parameter simulated binary crossover (SBX) [48], polynomial mutation [13], and elitism [13]. Key differences between the algorithms are highlighted in the following sections. The reader should note that this paper assumes a basic prior knowledge of MOEAs. Readers interested in introductions to multi-objective optimization and EMO tools should refer to the texts by Deb [13] and Coello Coello et al. [31].

3.1. NSGAII

The NSGAII is a second generation MOEA developed by Deb et al. [1] which made significant improvements to the original NSGA by (i) using a more efficient non-domination sorting scheme, (ii) eliminating the sharing parameter, and (iii) adding an implicitly elitist selection method that greatly aids in capturing Pareto surfaces. In addition, the NSGAII can handle both real and binary representations. The NSGAII was chosen for comparison in this study because it has been successfully employed in prior LTM studies [20], and it is the original algorithm from which the authors of this study developed the ε-NSGAII. For the LTM test case, all of the algorithms evaluate potential designs in terms of a vector of objectives. The concept of Pareto-dominance is used to assign fitness values to the sampling designs.

For example, a design $\mathbf{x}_1$ dominates another design $\mathbf{x}_2$ if and only if it performs as well as $\mathbf{x}_2$ in all objectives and better in at least one objective. The fast non-domination sorting approach of the NSGAII ranks each design according to the number of designs that dominate it. Once fitness is assigned, two-step crowded binary tournament selection is performed. In cases where two designs have different ranks, the individual with the lower rank is preferred (i.e., the design that is dominated by fewer other designs). Alternatively, if both designs possess the same rank, then the design with the larger crowding distance is preferred (where crowding distance is the average Euclidean distance between an individual design and those designs within the population that have been assigned the same rank). Designs with higher crowding distances add more diversity to the design population, which helps to ensure that the NSGAII will find solutions along the full extent of the Pareto surface. The interested reader should refer to Deb et al. [1,49] for additional details. Zitzler et al. [8] and Deb et al. [49] have shown that the NSGAII performed as well as or better than other second-generation MOEAs on difficult multi-objective problems.

3.2. ε-NSGAII

The ε-NSGAII builds on the NSGAII, by adding ε-dominance archiving [5,6], adaptive population sizing [7], and automatic termination to minimize the need for extensive parameter calibration as demonstrated by Reed et al. [2]. The concept of ε-dominance allows the user to specify the precision with which they want to quantify each objective in a multi-objective problem. Fig. 3 demonstrates the concept of ε-dominance using a three step approach. First, a user specified ε grid is applied to the search space of the problem based on their precision goals. Larger ε values result in a courser grid (and ultimately fewer solutions) while smaller ε values produce a finer grid. Grid blocks containing multiple solutions preserve the solution closest to the lower left-hand corner of the block (assuming minimization of all objectives). In the second step, non-domination sorting based on the grid blocks is conducted. For example, the solution located in the leftmost column four rows from the bottom dominates the shaded region of grid blocks above and to the right. Thus, the solution above it is dominated in terms of the required precision, and hence eliminated. This results in a "thinning" of solutions (step 3) and promotes a more even search of the objective space. It is important to note that the addition of ε-dominance archiving does not add additional parameters to the algorithm. Rather, it allows the user to define the precision requirements that make sense for their particular application. The interested reader can refer to prior work by Laumanns et al. [5] and Deb et al. [6] for a more detailed description of ε-dominance.

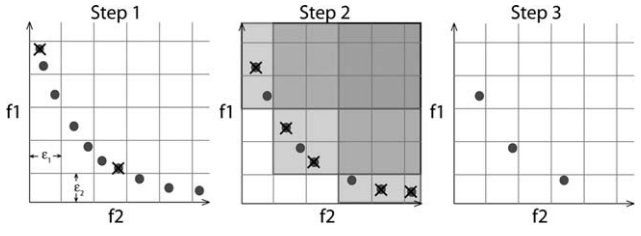

Fig. 3. Illustration of the ε-dominance concept.

The ε-NSGAII uses a series of "connected runs" where small populations are initially exploited to precondition search and automatically adapt population size commensurate with problem difficulty (see Fig. 4). As the search progresses, the population size is automatically adapted based on the number of ε-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 using a 25% injection scheme. In the injection scheme, 25% of the subsequent population will be composed of the ε-non-dominated archive solutions and the other 75% will be generated randomly. This assists the search in two ways: (i) by directing the search using previously evolved solutions and (ii) by adding new solutions to encourage the exploration of additional regions of the search space. This injection scheme bounds the population size to four times the number of solutions that exist at the user specified ε resolution. Theoretically, this approach allows the MOEA's population size to increase or decrease, and in the limit when the ε-dominance archive size stabilizes, the ε-NSGAII's "connected runs" are equivalent to a diversity-based EA search enhancement recommended by Goldberg [50] 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 Δ% across two successive runs.

The primary goal in the development of the ε-NSGAII was to provide a highly reliable and efficient MOEA which minimizes the need for traditional EA parameterization and allows the user to focus on problem specific search quality goals. Computational savings can be viewed in three contexts: (i) the use of minimal population sizes, (ii) the elimination of trial-and-error application runs to determine search parameters, and (iii) the elimination of random seed analysis. 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. Moreover, this approach minimizes unnecessary runtime by terminating search based on user defined precision goals.

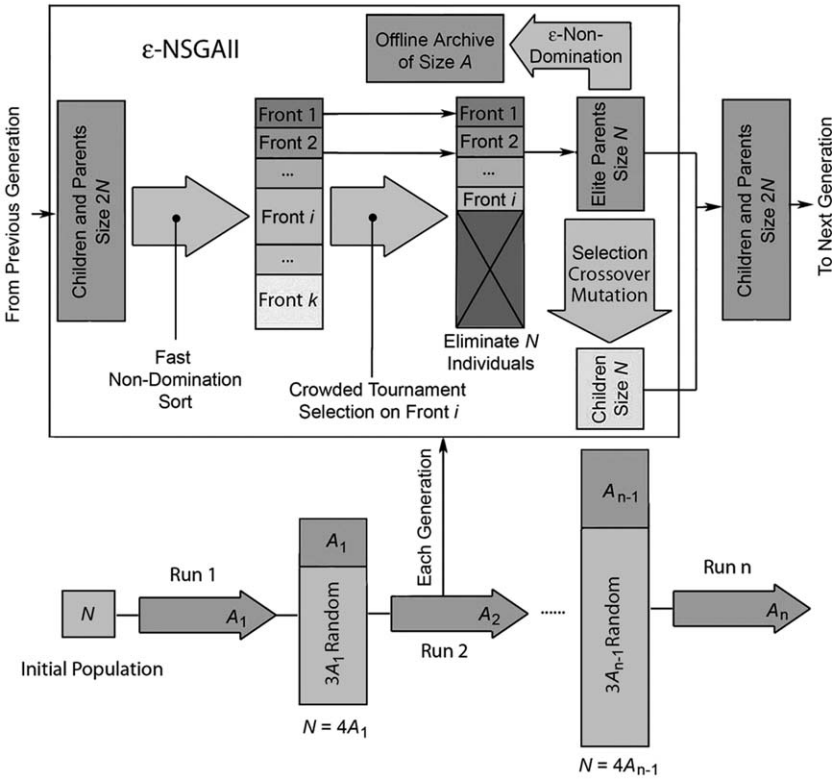

Fig. 4. Schematic diagram of the ε-NSGAII illustrated using the notation of Deb et al. [1]. This figure shows the "connected runs" and dynamic population sizing concepts of the ε-NSGAII with a blow-up diagram of what occurs during each generation of each run. In the figure, N represents population size and A represents ε-dominance archive size.

3.3. εMOEA

The εMOEA [6] is a steady-state MOEA which, similarly to the ε-NSGAII, uses the concept of ε-dominance to promote solution diversity and provide the user with the capability to specify desired objective precision. This algorithm evolves both an EA population and an archive population simultaneously. Initially, a population is generated at random based on a user specified size, and an archive is generated based on the ε-non-dominated solutions from this initial population. Next, an individual design is chosen from the population and the archive for mating. The design chosen from the population is based on random selection of two designs, and the one which dominates is chosen for mating, otherwise if they are non-dominated, one is chosen at random. The design chosen from the archive is simply chosen at random. The two designs are then combined to produce one new design. To determine if the new design is included in either the population or the archive, several tests ensue. For inclusion in the population, three scenarios exist: (i) if the new design dominates any designs which already exist, then it replaces one at random, (ii) if it is dominated by any existing designs, it is not accepted into the population, and (iii) if it is non-dominated with respect to the existing designs, it replaces a random member of the population. For inclusion in the archive, three scenarios also exist: (i) if the new design is ε-dominated by any design in the archive, it is not accepted, (ii) if it ε-dominates any member of the archive, it randomly replaces a dominated design, and (iii) if the new design is ε-non-dominated, and if it does not occur within any of the archive designs' ε hyperboxes, then it is accepted. Otherwise, the two designs occurring in the same hyperbox are compared and the best design with respect to domination in the traditional sense is accepted. The size of the archive is inherently bounded by the user specified ε resolution of the objectives [6]. Termination of the εMOEA is based on a user specified maximum runtime. Deb et al. [6] have shown that the εMOEA performs as well as or better than various other second-generation MOEAs on difficult multi-objective problems.

3.4. SPEA2

The SPEA2 [8] was selected for comparison in this study because it is a benchmark MOEA that has proven quite effective at solving numerous high-dimensional problems while maintaining excellent solution diversity [1,8–11]. Initially, a random population of user specified size is generated and the non-dominated individuals are placed into a fixed size archive (the size of which is also

user specified). Design fitness is then assigned to each archive solution by assigning a strength value which represents the number of designs which the solution dominates, then changing the strength to raw fitness by summing the strengths of all designs from the population and archive which dominate the solution, and finally by addition of a density value which is based on the kth nearest neighbor method [51] where density is a decreasing function of the distance to the kth nearest design. It is important to note that raw fitness is to be minimized, meaning that a raw fitness of zero indicates a non-dominated design and a high raw fitness value means the design is dominated by many individuals. Next, environmental selection ensues in which all non-dominated individuals from the population and archive are copied to the next archive. However, since the archive size is fixed, two additional scenarios exist beyond the non-dominated solutions exactly filling the archive. The first scenario occurs when the number of non-dominated solutions is smaller than the archive size. In this case, all non-dominated solutions are copied and the best dominated solutions are copied to fill the remainder of the archive. The second scenario is when there are more non-dominated solutions than the archive can store. In this case, archive truncation occurs in which solutions with minimum distance to the kth nearest neighbor are removed. Termination of the SPEA2 is based on a user specified maximum runtime.

4. Metrics of performance

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. This analysis can be extremely time consuming if not impossible for computationally intensive water resources applications. The goals of this comparative analysis is to identify which of the algorithms: (i) attain very close approximations to the true Pareto front (i.e., convergence), (ii) attain solutions along the full extent of the Pareto front (i.e., diversity), (iii) maximize the rate of search progress (i.e., computational efficiency), and (iv) show the least sensitivity to random seed effects (i.e., search reliability).

To aid in assessing the performance of each algorithm based on these criteria, several performance metrics which assign a measure of quality to the algorithms' solutions are used. Performance metrics can be used to assess the quality of the end result, or to visualize the dynamics of the runtime performance of the algorithms. This is particularly useful when comparing multiple algorithms. When the end results of the algorithms do not differ substantially, the way in which they achieve these results throughout their run may provide more information regarding performance. In this study, three runtime performance metrics are used: convergence, diversity (both previously published by Deb and Jain [11]), and ε-performance (a metric recently proposed by the authors [4]). Two unary metrics, the hypervolume indicator metric proposed by Zitzler and Thiele [52], and the unary ε-indicator metric proposed by Zitzler et al. [53], are used to evaluate the average final performances of the algorithms. In addition, the first-order empirical attainment function proposed by Fonseca et al. [54] is used to assess each of the algorithms' abilities to attain solutions on two-objective subsets of the full four-objective Pareto front.

Many of the performance metrics used in this study require a reference solution set for comparison purposes. The reference set can represent the true Pareto-optimal solution set or the best known approximation to the Pareto-optimal set attained through previous algorithm runs or by other means. In this study, if a metric required a reference set, the true four-objective Pareto-optimal set for the LTM test case was used.

4.1. Runtime convergence and diversity

The runtime convergence and diversity metrics used in this study were originally proposed by Deb and Jain [11]. The runtime convergence metric measures the average Euclidean distance between an approximation set (i.e., the set of solutions found by the algorithm [53]) and a reference set (i.e., the known Pareto front obtained through enumeration of the 25 well test case). The convergence metric when normalized ranges from zero (indicating perfect performance) to one. The runtime diversity metric measures the ''spread'' of solutions along the full extent of the tradeoff comparing the distribution of the approximation set with respect to a reference set. In calculating the metric, the approximation set and reference set are projected onto an $M-1$ dimensional hyperplane where M is the number of objectives. The projected approximation set is then compared to the projected reference set in terms of distribution across the objective space. The metric value is determined by favouring well distributed solutions with a high weight and assigning low weights to clustered solutions. The values of this metric range from zero to one (indicating perfect diversity).

4.2. ε-Performance

The ε-performance metric, recently proposed by Kollat and Reed [4] assigns a measure of performance by accounting for the proportion of solutions that fall within a user specified ε distance of a reference set. First, the

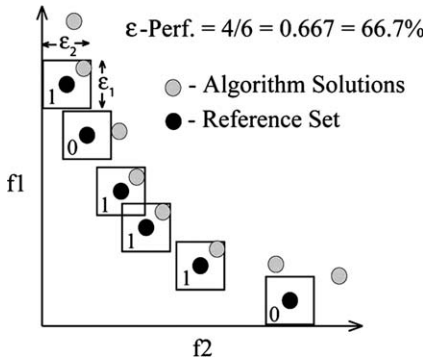

Fig. 5. Example calculation of the ε-performance metric.

ε-dominance concept is applied to the reference set according to user specified values (see Fig. 3). The proportion of solutions found within an ε hypercube of each reference set solution is measured by matching solutions from the approximation set to the reference set. Reference set solutions with a matching approximation set solution receive a score of one while those with no matching solution receive no score (see Fig. 5 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 ε hyperboxes (one case of this is shown in Fig. 5). This metric inherently provides a measure of both convergence and diversity because it accounts for the distance of the solutions from the reference set (i.e., convergence), while only allowing for one solution to be matched with each ε hyperbox surrounding the reference points, thus preventing clustered solutions from having additional weight in the calculation of the metric. This metric ranges from zero to one where a metric value of one would indicate 100% convergence to within ε of the reference set. The interested reader should note that unlike the diversity metric which becomes impractical for higher-dimensional problems (see Deb and Jain [11] for details), the ε-performance metric can easily be extended to any number of objectives.

4.3. Hypervolume indicator and ε-indicator

The hypervolume indicator metric [52] represents the volume of the objective space that is dominated by a solution set. Since the true solution was enumerated for the LTM test case used in this study, the metric was calculated as the difference between the volume of the objective space dominated by the true Pareto-optimal set and the volume of the objective space dominated by an approximation set (i.e., solutions generated by the algorithms). Hence, smaller indicator values are desired as this indicates a smaller difference between the reference set and the approximation set (assuming minimization of objectives). The unary ε-indicator metric [53] represents the smallest distance that an approximation set must be translated in order to completely dominate the reference set. Again, smaller values of this metric are desirable as this indicates a closer approximation to the reference set. For additional details on the hypervolume indicator and the unary ε-indicator metrics, see Zitzler and Thiele [52] and Zitzler et al. [53].

4.4. First-order empirical attainment function

The first-order empirical attainment function [55] can be used to represent the probabilistic performance of MOEAs by measuring the "attainment" of a reference or true Pareto-optimal set of solutions by the distribution of approximation sets generated using multiple random seed MOEA runs. For example, the minimum attainment surface represents the poorest performance of the algorithm across all runs and the maximum attainment surface, the best performance across all runs. The average attainment surface is representative of the surface which was attained in 50% of the runs. The attainment function becomes extremely computationally intensive and difficult to compute beyond two-dimensions, so in this study we demonstrate the attainment metric using representative results for two-objective subsets of the four-objective space. Differences in attainment functions across algorithms can be assessed using a Kolmogorov–Smirnov test which checks for significant differences between cumulative distribution functions.

5. Computational experiment

In this study, the NSGAII, the εMOEA, and the SPEA2 were parameterized according to the most commonly recommended settings from the EMO literature. It should be noted that all of the tested algorithms used simulated binary crossover (SBX) [48], polynomial mutation [13], and elitism [13]. All of the algorithms utilized the same probabilities of crossover and mutation ($p_c = 1.0$ and $p_m = 1/N$, respectively, where N is the population size), and the same crossover and mutation distribution indices ($\eta_c = 15$ and $\eta_m = 20$, respectively) associated with each of these operators. The NSGAII, the εMOEA, and the SPEA2 were assigned a population size of 100 individuals based on recommendations in prior literature [1,6,9,11,49]. The ε-NSGAII used adaptive population sizing and automatic termination, thus requiring only a starting population size (for this study, $N_i = 10$ was chosen) and termination criteria that required at least a 10% improvement in the ε-non-dominated archive. The relevant parameterization of each of the algorithms is summarized in Table 1.

Table 1
Summary of algorithm parameters used in this study

	NSGAII	ε-NSGAII	εMOEA	SPEA2
Population size	$N = 100$	Dynamic starting with $N = 10$	$N = 100$	$N = 100$
Termination criteria	192,400 Evaluations	<10% Improvement	192,400 Evaluations	192,400 Evaluations
Probability of crossover	1.0	1.0	1.0	1.0
Crossover dist. index	15	15	15	15
Probability of mutation	$1/N$	$1/N$	$1/N$	$1/N$
Mutation dist. index	20	20	20	20
$[\varepsilon_{cost}\ \varepsilon_{conc}\ \varepsilon_{uncert}\ \varepsilon_{mass}]$	NA	$[10^0\ 10^{-5}\ 10^{-2}\ 10^{-6}]$	$[10^0\ 10^{-5}\ 10^{-2}\ 10^{-6}]$	NA
Variable representation	Real	Real	Real	Real

In order to accurately assess the reliability of each algorithm, 50 random seeds were chosen resulting in 50 random seed trial runs for each algorithm. The reader should note that identical random seeds were specified for the NSGAII, the ε-NSGAII, and the εMOEA since they all use the same random number generator. The random number generator used by the SPEA2 differed from the other algorithms making the choice of identical random seeds impossible. The impacts of random number generator differences were minimized in this study by using 50 trial runs for statistical performance assessment of each MOEA. In order to facilitate a fair performance comparison (since the ε-NSGAII automatically terminates based on user defined accuracy goals), the ε-NSGAII was run for 50 random seed trial runs and the average number of design evaluations that it required to automatically terminate was used as a basis for parameterizing the runtime of the NSGAII, the εMOEA, and the SPEA2 for the same random seeds (this resulted in approximately 192,400 design evaluations). Parameterizing the runtime of the other algorithms in this manner gave each algorithm the same opportunity (on average) to generate the Pareto front for the LTM test case. However, the reader should note that this maximum runtime for the NSGAII, the εMOEA, and the SPEA2 would not be known in advance, requiring the user to estimate the runtime needed to sufficiently solve their problem using trial-and-error analysis.

Epsilon resolution settings for each of the four-objectives (ε_{cost}, ε_{conc}, ε_{uncert}, and ε_{mass}) were chosen based on reasonable precision requirements and were set to 10^0, 10^{-5}, 10^{-2}, and 10^{-6}, respectively, for each the ε-NSGAII and the εMOEA. For example, ε_{cost} was set to 10^0 because the cost objective was formulated to represent the number of sampling points in a potential design, and the ε_{mass} was set to 10^{-6} because values for this design objective ranged from 10^{-6} to 10^3, thus requiring high precision to capture low objective values. The ε resolution settings limit the archive size to the number of solutions that exist at that resolution and are solely based on what the practitioner considers an acceptable or publishable solution precision. Reed et al. [56] demonstrated that interactive refinement of search precision requirements could reduce the number of design evaluations required to solve a similar LTM application by 90% while maintaining a high quality representation of the objective tradeoffs. The enumeration of the 25 well LTM test case revealed a total of 2439 Pareto-optimal solutions. At the ε resolution setting used in this study, there are 2411 ε-non-dominated solutions. To make the SPEA2 comparable to the other algorithms, its maximum archive size was set to 2411 solutions so that archive truncation would occur if the algorithm found more than this quantity of solutions. Again, the reader should note that archive size specification significantly impacts the SPEA2's performance and that the algorithm's archive is typically sized using trial-and-error analysis. In order to make the NSGAII comparable to the other algorithms, an offline archive which used traditional non-domination sorting was added to the algorithm and updated after each generation.

Other relevant parameters include a diversity metric grid specification of 12 blocks for each coordinate axis of the projected solutions (see Deb and Jain [11] for additional information) and ε resolutions settings (ε_{cost}, ε_{conc}, ε_{uncert}, and ε_{mass}) for the ε-performance metric of 10^0, 10^{-1}, 10^0, and 10^{-2}, respectively. A lower resolution was used in the calculation of the ε-performance metric so all of the algorithms' results could be visualized and compared. In addition, since the εMOEA and the SPEA2 are real-coded algorithms, the inherent 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 then 0.5, it was changed to a 0.0 and variables greater than or equal to 0.5 were changed to 1.0. Interestingly, this approach improved the performances of both the NSGAII and the ε-NSGAII relative to their binary implementations.

6. Results

Fig. 6 presents runtime results of ε-performance, convergence, and diversity, versus total design evaluations for each of the algorithms compared. The results of all 50 random seed trial runs are shown in the figure with

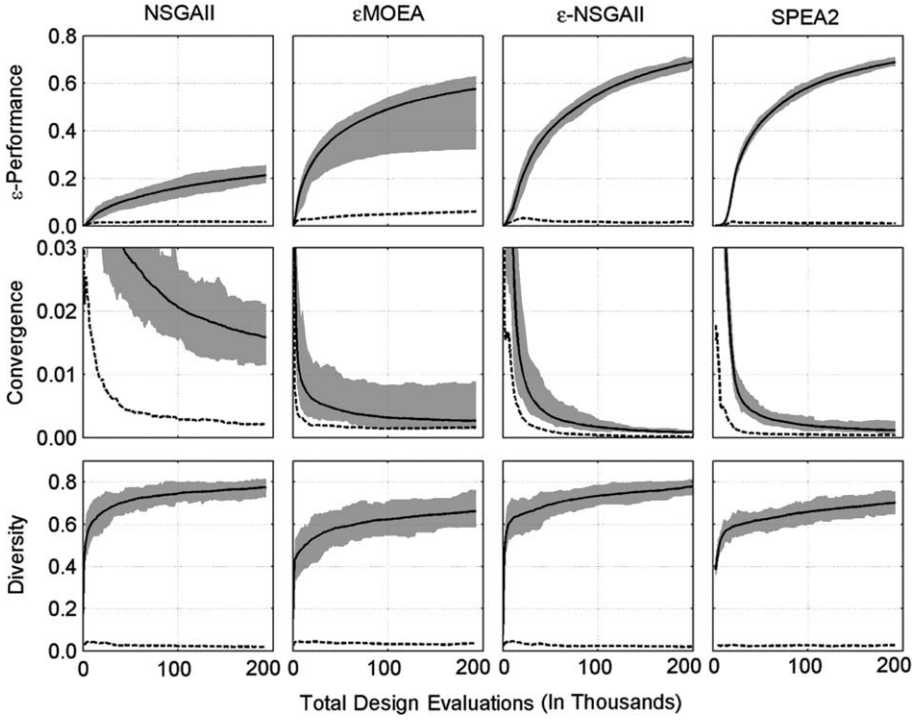

Fig. 6. Dynamic performance plots of ε-performance, convergence, and diversity versus total design evaluations for the NSGAII, the ε-NSGAII, the εMOEA, and the SPEA2 for 50 random seed trial runs. Mean performance is indicated by a solid line, the standard deviation by a dashed line, and the range of performance by the shaded region.

the mean performance indicated by a solid line, the standard deviation by a dashed line, and the range of random seed performance indicated by the shaded region. Visualizing the results in this manner allows for comparison between the dynamics and reliability (i.e., larger shaded regions indicate lower random seed reliability) of each algorithm. The first row of plots portraying runtime ε-performance reveal that the NSGAII achieves an end-of-run ε-performance of 21% on average. The εMOEA achieves 59% ε-performance, but its reliability is low as is indicated by its range of performance. The ε-NSGAII and SPEA2 attain the highest levels of average ε-performance at 68% and 69%, respectively, and both algorithms are highly reliable with the SPEA2 exhibiting slightly higher reliability than the ε-NSGAII. In fact, the ε-performance of the ε-NSGAII and the SPEA2 is consistently two to three times that achieved by the NSGAII. Key performance differences between the ε-NSGAII and the SPEA2 are highlighted in Fig. 7. This figure is formatted similarly to Fig. 6 except that it focuses on key areas where performance differs between the two algorithms. The first row of plots in this figure shows the early runtime ε-performance (i.e., less than 20,000 design evaluations) of the ε-NSGAII and the SPEA2. In these plots, the lower bound of the ε-NSGAIIs performance approximately matches the upper bound performance of the SPEA2 indicating that the ε-NSGAII is highly efficient early in its runs. This also indicates that the use of initially small population sizes are greatly improving early search progress. Differences in reliability between the algorithms can be explained by the ε-NSGAIIs use of initially small populations, in this case 10 individuals, versus the SPEA2s use of 100 individuals to begin search. This result also indicates the ε-NSGAII would be more effective at conducting computationally expensive runs to obtain a rough approximation of the Pareto front.

Runtime convergence metric results are shown in the second row of Fig. 6. The runtime convergence metric displays a similar trend in performance differences between the algorithms. The NSGAII performs the poorest in terms of convergence and has a low reliability. The εMOEA performs much better than the NSGAII but achieves a low level of reliability later in its runs with its distribution of random seed trial runs weighted towards lower convergence values (as is indicated by the mean performance line). The ε-NSGAII attains the best final measure of convergence and maintains the this superiority throughout its entire run when compared to the NSGAII and the εMOEA. Comparing the ε-NSGAII and the SPEA2 reveals that the larger search population used by the SPEA2 yields a higher reliability near the beginning of its runs while the ε-NSGAII attains a higher reliability towards the end of its runs due to its adaptive population sizing. Focusing on the later portion of the convergence plot (greater than

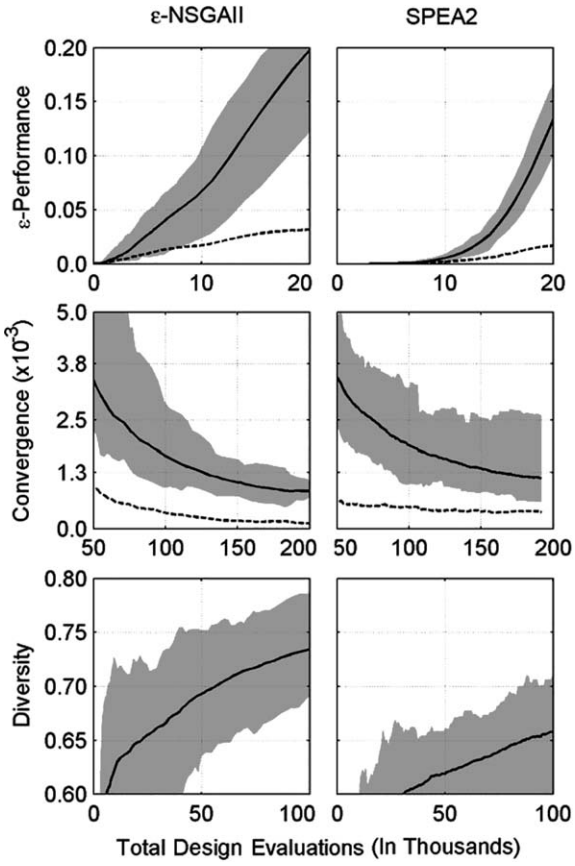

Fig. 7. Dynamic performance plots of ε-performance, convergence, and diversity versus total design evaluations for the ε-NSGAII and the SPEA2 for 50 random seed trial runs. These plots focus on interesting regions showing disparate performance between the two algorithms. The top panels focus on the ε-performance results of the algorithms at less than 20,000 design evaluations. The middle panels focus on convergence values less than 0.005 at greater than 50,000 design evaluations. The bottom panels focus on diversity values greater than 0.6 at less than 100,000 design evaluations.

50,000 design evaluations) in Fig. 7 reveals a significant difference in performance between the two algorithms. Namely, the ultimate average convergence of the ε-NSGAII (indicated by the solid line) is consistently better than the SPEA2s average convergence and the envelope of reliability consistently improves throughout the duration of the ε-NSGAIIs runs.

The diversity metric reveals that both the NSGAII and the ε-NSGAII perform comparatively well indicating that even though the NSGAII is achieving low average convergence and ε-performance measures, the solutions that it is finding are representative of the full extent of the Pareto front. Both algorithms ultimately perform slightly better in diversity than the εMOEA and the SPEA2 and they maintain this superiority throughout their entire runs. Fig. 7 highlights the differences in diversity between the ε-NSGAII and the SPEA2 in the third row of plots for early portions of the runs (less than 100,000 design evaluations). These plots reveal that the ε-NSGAII achieves a higher level of diversity than the SPEA2 early in its runs. In fact, the lower bound diversity performance of the ε-NSGAIIs approximates the upper bound performance of the SPEA2.

Final performance metric results are summarized in Table 2. The average final ε-performance of the ε-NSGAII is superior to the NSGAII and the εMOEA, and comparable to the SPEA2. The average final convergence of the ε-NSGAII is superior to all other algorithms compared. The average final diversity of the ε-NSGAII is superior to the εMOEA and the SPEA2 and equivalent to the NSGAII. The results shown in Table 2 emphasize the need to assess dynamics of performance rather than just end-of-run results.

Hypervolume indicator and ε-indicator metric results are shown in Table 3. The hypervolume indicator measure represents the average difference in dominated hypervolume between the true Pareto-optimal set and the approximation sets generated by the algorithm runs. The average hypervolume indicator results reveal that the hypervolume measure achieved by the ε-NSGAII is an order of magnitude lower than that achieved by the other algorithms indicating superior performance. The εMOEA performs the poorest in terms of its hypervolume measure. In addition, the ε-NSGAII achieves the lowest standard deviation of all algorithms in this measure. Interestingly, the NSGAII performs better in this measure than the SPEA2 indicating that even though the NSGAII is finding fewer solutions, the spread of the solutions throughout the objective space are dominating a volume greater than that of the SPEA2. The ε-indicator metric represents the smallest distance on average that an algorithm's approximation sets must be translated to completely dominate the true

Table 2
Mean and standard deviations of ultimate algorithm performance across 50 random seeds

	ε-Performance mean (std. dev.)	Convergence mean (std. dev.)	Diversity mean (std. dev.)
NSGAII	0.21 (0.016)	0.0160 (0.0022)	0.77 (0.019)
εMOEA	0.59 (0.068)	0.0028 (0.0021)	0.65 (0.037)
ε-NSGAII	0.68 (0.022)	0.0008 (0.0001)	0.77 (0.023)
SPEA2	0.69 (0.010)	0.0012 (0.0004)	0.70 (0.027)

Table 3
Mean and standard deviations of the hypervolume indicator and unary ε-indicator metric results for algorithm performance across 50 random seed trial runs

	Hypervolume ($\times 10^{-6}$) mean (std. dev.)	ε-Indicator mean (std. dev.)
NSGAII	12.5 (2.23)	2.07 (0.19)
εMOEA	41.1 (20.0)	2.86 (0.71)
ε-NSGAII	1.33 (0.57)	1.11 (0.09)
SPEA2	17.4 (8.96)	1.96 (0.27)

Pareto-optimal set. The results of this metric indicate that the ε-NSGAII requires the smallest average translation distance and that the εMOEA requires the largest translation distance on average. In addition, the ε-NSGAII achieves the smallest standard deviation in this measure compared to the other algorithms. The Kruskal–Wallis non-parametric statistical test for significant differences between multiple independent samples [57] was used to determine if the hypervolume and ε-indicator metric results differed significantly between algorithms. The test revealed that all differences were significant at the 99% confidence level.

Reed and Minsker [20] proposed a LTM design methodology for high-order Pareto optimization whereby conflicting objectives which are subsets of the larger multi-objective problem are analyzed to aid in choosing a single compromise solution from the full approximation set. For the LTM test case used in this study, the conflicting pairs of objectives analyzed include sampling cost and concentration estimation error (cost–conc) and sampling cost and concentration estimation uncertainty (cost–uncert). Plots of the two-objective tradeoffs drawn from the true four-objective Pareto surface are shown in Fig. 8. These tradeoffs are subsequently used to better understand the relationships between the LTM design objectives. When using this methodology, the ability of the MOEA to effectively find close approximations to the two-objective conflicting tradeoffs is important.

The first-order empirical attainment function can be used to determine the range of algorithm performance in obtaining the two-objective tradeoffs and to determine if the abilities of the algorithms to attain these tradeoffs differ significantly. The first-order attainment surfaces for the conflicting objective pairs, cost–conc and cost–uncert are shown in Fig. 9. In the figure, the NSGAII and the ε-NSGAII appear to be the most reliable algorithms for consistently finding solutions close to or matching the two-objective tradeoffs as is indicated by their small range of performance. The εMOEA has the most difficulty finding solutions on the two-objective tradeoffs for its runs with 50% attainment differing substantially from the true tradeoffs for cost levels 7 through 11. The SPEA2 performs slightly better than the εMOEA in that its 50% attainment levels for the cost–conc tradeoff only differ significantly for cost levels 7 through 10 and for the cost–uncert tradeoffs for cost levels 8 and 9. The SPEA2s range of performance is significantly poorer than the NSGAII and the ε-NSGAII. The ε-NSGAII is clearly the superior algorithm for consistently finding the two-objective tradeoffs as the range of performance for its runs is quite small for both tradeoffs. All differences between attainment surfaces achieved by each algorithm are significant to greater than 99% confidence using a generalization of the multi-variate two-sided Kolmogorov–Smirnov test for two independent samples [55,58].

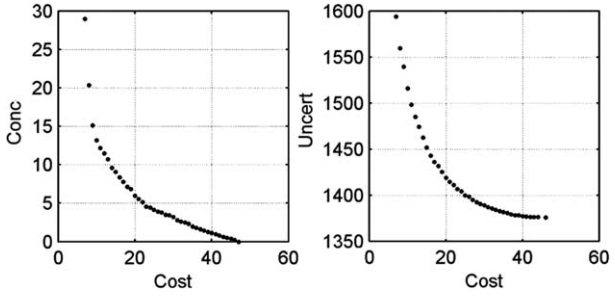

Fig. 8. Two-objective tradeoffs drawn from the true four-objective Pareto surface. The cost–conc tradeoff contains 42 solutions and the cost–uncert tradeoff contains 37 solutions.

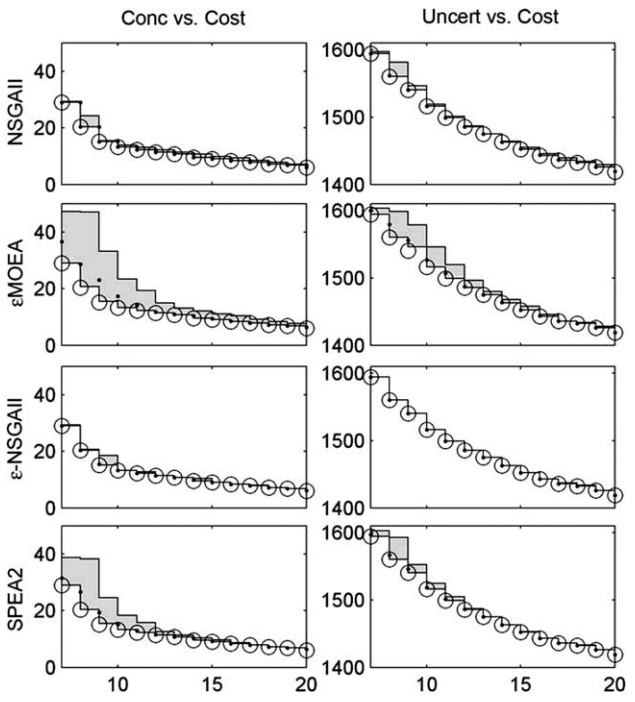

Fig. 9. First-order attainment surfaces of the conflicting two-objective pairs, cost–conc and cost–uncert. The minimum and maximum attainment surfaces for all runs is represented by the shaded region. The mean or 50% attainment surface is represented by dots and the true tradeoff solutions are represented by circles. Cost levels 7 through 20 are shown for clarity.

7. Discussion

The results presented in this study indicate superior performance of the ε-NSGAII in terms of the hypervolume indicator, ε-indicator, and first-order empirical attainment function metrics. In addition, the runtime metric results indicate that the ε-performance and convergence dynamics of the ε-NSGAII are competitive to superior relative to the SPEA2, with both algorithms greatly outperforming the NSGAII and εMOEA in terms of these metrics. Dynamic diversity results indi-

cate superior performance of the ε-NSGAII relative to the SPEA2. The improvements in performance of the ε-NSGAII over its parent algorithm the NSGAII demonstrate that the application of ε-dominance archiving, dynamic population sizing with archive injection, and automatic termination greatly improve algorithm efficiency and reliability. In addition, the usability of the algorithm is improved through the elimination of the population sizing parameter, the replacement of runtime specification by a more intuitive termination criteria, and the addition of ε-dominance archiving which eliminates large costs associated with computations at unnecessary levels of precision.

Although the ε-NSGAII and the SPEA2 use the same mating and mutation operators, the key factor leading to differences in their performances results from the ε-NSGAIIs use of dynamic population sizing and injection. The original NSGAII uses a selection process (see Fig. 4) where the best N solutions are selected from a combined pool of N elite parents and N children generated from the elite parents. In essence, this represents a form of truncation selection where the top N solutions from a pool of $2N$ solutions are selected every generation. This truncation selection makes the original NSGAII very sensitive to population size [2] because increasing the population size results in an increase in selection pressure (i.e., the probability of selecting good solutions increases). The ability of the ε-NSGAII to dynamically change its population size allows the algorithm to increase or decrease selection pressure commensurate with search progress. When the algorithm is performing well, selection pressure is increased (i.e., the population size is increased) and the algorithm quickly proceeds towards favourable regions of the search space (i.e., injected archive solutions). However, if the algorithm is having difficulty finding highly fit solutions, selection pressure is either maintained or decreased and the algorithm continues exploring the search space. Initially small population sizes are utilized by the ε-NSGAII to direct search at a low computational cost until the algorithm begins finding highly fit solutions. At this point, the algorithm immediately increases its selection pressure, and quickly proceeds towards favourable regions of the search space.

Prior knowledge of the decision and objective space characteristics of high-dimensional water resources problems is usually limited. Hence, choosing optimal MOEA parameters to achieve efficient and reliable algorithm performance is necessary to ensure that the computation time required to solve water resources problems is not wasted on trial-and-error analysis. The ε-NSGAIIs use of dynamic population sizing makes this lack of prior knowledge less relevant by allowing the algorithm to dynamically adjust its population size commensurate with problem difficulty and resolution requirements. In addition, providing automatic termination capabilities eliminates the need to parameterize algorithm runtime by providing a more intuitive method whereby the user can specify the percentage change in solution quantity and quality required to continue search. It is important to note that the other algorithms analyzed in this study used information gained from the ε-NSGAII in terms of required runtime. In addition, the SPEA2s archive size was parameterized based on the number of true Pareto optimal solutions found by enumerating the LTM test case. This information would not typically be known when solving water resources applications and trial-and-error analysis would be necessary to ensure that the archive size can accommodate a sufficient approximation to the true Pareto front.

It has been demonstrated that ε-dominance archiving will preserve a good representation of the Pareto front for both low and high resolution settings [5,6,56]. The reader should also note that the addition of ε-dominance archiving to the ε-NSGAII does not add additional parameters, but rather adds flexibility in that the user has the option to control precision requirements and hence algorithm runtime. Reed et al. [56] have shown that decreasing objective precision requirements of the ε-NSGAII does in fact result in decreased runtime (i.e., approximations are accepted to save computation time). More formally, from evolutionary algorithm theory [19,59], the population size required by MOEAs to solve multi-objective problems is directly related to the size of the Pareto-optimal set. ε-dominance archiving allows water resources users to directly reduce the ε-non-dominated set size and the ultimate computational requirements of their applications. Moreover, in the limit when the ε-dominance archive size stabilizes, the ε-NSGAIIs "connected runs" are equivalent to a diversity based EA search enhancement recommended by Goldberg [50] termed "time continuation". The ε-NSGAII shows great potential as an efficient, reliable, and easy-to-use MOEA for water resources applications.

8. Conclusions

This study compared the performance of four state-of-the-art evolutionary multi-objective optimization algorithms (the NSGAII, the ε-NSGAII, the εMOEA, and the SPEA2) on a four-objective LTM test case. The LTM test case objectives included: (i) minimize sampling cost, (ii) minimize contaminant concentration estimation error, (iii) minimize contaminant concentration estimation uncertainty, and (iv) minimize contaminant mass estimation error. The 25-well LTM test case was enumerated to provide the true four-objective Pareto-optimal solution set to facilitate rigorous testing of the EMO algorithms. The performances of the four algorithms were assessed and compared using three runtime performance metrics (convergence, diversity,

and ε-performance), two unary metrics (the hypervolume indicator and ε-indicator) and the first-order empirical attainment function. The use of ε-dominance archiving, dynamic population sizing with archive injection, and automatic termination in the ε-NSGAII have resulted in a robust algorithm that is easier to implement than traditional MOEAs. The results of this study indicate that the ε-NSGAII is greatly improved over its parent algorithm, the NSGAII, its performance exceeds that of the εMOEA, and its performance is competitive to superior relative to the SPEA2. However, the key difference between the ε-NSGAII and the SPEA2 is that the ε-NSGAII eliminates much of the traditional trial-and-error parameterization required by the SPEA2 (i.e., water resources applications are easier to solve efficiently and reliably). This study demonstrates a rigorous methodology for testing new MOEA tools and highlights key performance issues that impact water resources applications. Overall, the ε-NSGAII developed as part of this study shows great potential as an efficient, reliable, and easy-to-use MOEA for water resources applications.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.advwatres.2005.07.010.

References

[1] 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.

[2] Reed P, Minsker BS, Goldberg DE. Simplifying multiobjective optimization: an automated design methodology for the nondominated sorted genetic algorithm-II. Water Resour Res 2003;39(7):1196. doi:10.1029/2002WR001483.

[3] Reed P, Devireddy V. Groundwater monitoring design: a case study combining epsilon-dominance archiving and automatic parameterization for the NSGA-II. In: Coello-Coello C, editor. Applications of multi-objective evolutionary algorithms. Advances in natural computation series, vol. 1. New York: World Scientific; 2004. p. 79–100.

[4] Kollat JB, Reed PM. The value of online adaptive search: a performance comparison of NSGA-II, ϵ-NSGAII, and ϵMOEA. In: Coello CC, Aguirre AH, Zitzler E, editors. The Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Lecture Notes in Computer Science, vol. 3410. Guanajuato, Mexico: Springer Verlag; 2005. p. 386–98.

[5] Laumanns M, Thiele L, Deb K, Zitzler E. Combining convergence and diversity in evolutionary multiobjective optimization. Evol Comput 2002;10(3):263–82.

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

[7] Harik GR, Lobo FG. A parameter-less genetic algorithm. Tech Rep IlliGAL 99009, University of Illinois at Urbana-Champaign, good review of onemax, noisy onemax, and bounded deceptive problems. Talks about concept of genetic drift. Includes concepts on automatic paramterization and population double, 1999.

[8] Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. Tech Rep TIK-103, Department of Electrical Engineering, Swiss Federal Institute of Technology, 2001.

[9] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 2000; 8(2):125–48.

[10] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable multi-objective optimization test problems. In: Proceedings of the Congress on Evolutionary Computation (CEC-2002), 2002. p. 825–30.

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

[12] Pareto V, Cours D'Economie Politique, vols. 1 and 2, Rouge, Lausanne, 1896.

[13] Deb K. Multi-objective optimization using evolutionary algorithms. New York, NY: John Wiley & Sons Ltd; 2001.

[14] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley Publishing Company; 1989.

[15] Task Committee on Long Term Groundwater Monitoring. On long-term groundwater monitoring design, long-term groundwater monitoring: the state of the art. Tech Rep, American Society of Civil Engineers, 2003.

[16] Loaiciga H, Charbeneau RJ, Everett LG, Fogg GE, Hobbs BF, Rouhani S. Review of ground-water quality monitoring network design. J Hydraulic Eng 1992;118(1):11–37.

[17] Reed P, Minsker BS, Valocchi AJ. Cost effective long-term groundwater monitoring design using a genetic algorithm and global mass interpolation. Water Resour Res 2000;36(12): 3731–41.

[18] Cieniawski SE, Eheart JW, Ranjithan SR. Using genetic algorithms to solve a multiobjective groundwater monitoring problem. Water Resour Res 1995;31(2):399–409.

[19] 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.

[20] Reed P, Minsker BS. Striking the balance: long-term groundwater monitoring design for conflicting objectives. J Water Resour Plann Manage 2004;130(2):140–9.

[21] 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.

[22] 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.

[23] 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.

[24] Meyer PD, Valocchi AJ, Eheart JW. Monitoring network design to provide initial detection of groundwater contamination. Water Resour Res 1994;30(9):2647–59.

[25] 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.

[26] Sun N-Z. Inverse problems in groundwater modeling. Theory and applications of transport in porous media, vol. 6. New York, NY: Kluwer Academic Publishers; 1994.

[27] Wagner BJ. Sampling design methods for groundwater modeling under uncertainty. Water Resour Res 1995;31(10):2581–91.

[28] Schaffer JD. Some experiments in machine learning using vector evaluated genetic algorithms, Doctoral thesis, Vanderbilt University, 1984.

[29] Fonseca CM, Fleming PJ. An overview of evolutionary algorithms in multiobjective optimization. Evol Comput 1995;3(1): 1–16.
[30] Van Veldhuizen DA. Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Doctoral thesis, Air Force Institute of Technology, 1999.
[31] Coello CC, Van Veldhuizen DA, Lamont GB. Evolutionary algorithms for solving multi-objective problems. New York, NY: Kluwer Academic Publishers; 2002.
[32] Horn J, Nafpliotis F. Multiobjective optimization using the niched Pareto genetic algorithm. Tech Rep IlliGAL Report No. 93005, University of Illinois, 1993.
[33] 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.
[34] Halhal D, Walters GA, Ouazar D, Savic DA. Water network rehabilitation with structured messy genetic algorithm. J Water Resour Plann Manage 1997;123(3):137–46.
[35] 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.
[36] 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.
[37] Muleta MK, Nicklow JW. Decision support for watershed management using evolutionary algorithms. J Water Resour Plann Manage-Asce 2005;131(1):35–44.
[38] Keedwell E, Khu ST. Hybrid genetic algorithms for multiobjective optimisation of water distribution networks. Genetic and Evolutionary Computation Gecco 2004, Pt 2, Proceedings. Lecture Notes in Computer Science, vol. 3103. Springer-Verlag; 2004. p. 1042–53.
[39] Farmani R, Savic DA, Walters GA. Evolutionary multi-objective optimization in water distribution network design. Eng Optim 2005;37(2):167–83.
[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] Goovaerts P. Geostatistics for natural resources evaluation. New York, NY: Oxford University Press; 1997.
[46] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 1999;3(4):257–71.
[47] Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. In: Evolutionary methods for design, optimisation, and control, Barcelona, Spain, 2002. p. 95–100.
[48] Deb K, Agrawal RB. Simulated binary crossover for continuous search space. Tech Rep IITK/ME/SMD-94027, Indian Institute of Technology, Kanpur, 1994.
[49] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable test problems for evolutionary multi-objective optimization. Tech Rep KanGAL 2001001, Indian Institute of Technology Kanpur, 2001.
[50] Goldberg DE. The design of innovation: lessons from and for competent genetic algorithms. Norwell, MA: Kluwer Academic Publishers; 2002.
[51] Silverman BW. Density estimation for statistics and data analysis. Chapman and Hall; 1986.
[52] Zitzler E, Thiele L. Multiobjective optimization using evolutionary algorithms—a comparative case study. In: Eiben A, Back T, Schoenauer M, Schwefel H-P, editors. Parallel problem solving from nature (PPSN V). Lecture notes in computer science. Berlin, Amsterdam, The Netherlands: Springer-Verlag; 1998. p. 292–301.
[53] Zitzler E, Thiele L, Laumanns M, Fonseca CM, da Fonseca VG. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput 2003;7(2):117–32.
[54] Fonseca VGd, Fonseca CM, Hall AO. Inferential performance assessment of stochastic optimisers and the attainment function. In: Zitzler E, Deb K, Thiele L, Coelle CC, Corne D, editors. Evolutionary Multi-Criterion Optimization: First International Conference, EMO 2001. Lecture Notes in Computer Science, vol. 1993. Berlin, Zurich, Switzerland: Springer-Verlag; 2001.
[55] Fonseca CM, Fonseca VGd, Paquete L. Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In: Coello CC, Aguirre AH, Zitzler E, editors. Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005. Lecture Notes in Computer Science, vol. 3410. Berlin, Guanajuato, Mexico: Springer-Verlag; 2005. p. 250–64.
[56] Reed P, Kollat JB, Devireddy V. Using interactive archives in evolutionary multiobjective optimization: case studies for long-term groundwater monitoring design. Environ Model Software.
[57] Conover W. Practical nonparametric statistics. 3rd ed. Wiley series in probability and statistics. Applied probability and statistics section. New York: Wiley; 1999.
[58] Shaw K, Nortcliffe A, Thompson M, Love J, Fleming P. Assessing the performance of multiobjective genetic algorithms for optimization of a batch process scheduling problem. In: Proceedings of the Congress on Evolutionary Computation (CEC99), vol. 1, Washington DC, 1999. p. 37–45.
[59] Khan N. Bayesian optimization algorithms for multiobjective and hierarchically difficult problems. Ph.D. thesis, University of Illinois at Urbana-Champaign, 2003.