Hydrol. Earth Syst. Sci., 10, 289–307, 2006
www.hydrol-earth-syst-sci.net/10/289/2006/


Hydrology and Earth System Sciences

# How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration?

**Y. Tang, P. Reed, and T. Wagener**

Department of Civil and Environmental Engineering, The Pennsylvania State University, University Park, Pennsylvania, USA

Received: 3 November 2005 – Published in Hydrol. Earth Syst. Sci. Discuss.: 29 November 2005
Revised: 13 February 2006 – Accepted: 6 March 2006 – Published: 8 May 2006

**Abstract.** This study provides a comprehensive assessment of state-of-the-art evolutionary multiobjective optimization (EMO) tools' relative effectiveness in calibrating hydrologic models. The relative computational efficiency, accuracy, and ease-of-use of the following EMO algorithms are tested: Epsilon Dominance Nondominated Sorted Genetic Algorithm-II ($\varepsilon$-NSGAII), the Multiobjective Shuffled Complex Evolution Metropolis algorithm (MOSCEM-UA), and the Strength Pareto Evolutionary Algorithm 2 (SPEA2). This study uses three test cases to compare the algorithms' performances: (1) a standardized test function suite from the computer science literature, (2) a benchmark hydrologic calibration test case for the Leaf River near Collins, Mississippi, and (3) a computationally intensive integrated surface-subsurface model application in the Shale Hills watershed in Pennsylvania. One challenge and contribution of this work is the development of a methodology for comprehensively comparing EMO algorithms that have different search operators and randomization techniques. Overall, SPEA2 attained competitive to superior results for most of the problems tested in this study. The primary strengths of the SPEA2 algorithm lie in its search reliability and its diversity preservation operator. The biggest challenge in maximizing the performance of SPEA2 lies in specifying an effective archive size without a priori knowledge of the Pareto set. In practice, this would require significant trial-and-error analysis, which is problematic for more complex, computationally intensive calibration applications. $\varepsilon$-NSGAII appears to be superior to MOSCEM-UA and competitive with SPEA2 for hydrologic model calibration. $\varepsilon$-NSGAII's primary strength lies in its ease-of-use due to its dynamic population sizing and archiving which lead to rapid convergence to very high quality solutions with minimal user input. MOSCEM-UA is best suited for hydrologic model calibration applications that have small parameter sets and small model evaluation times. In general, it would be expected that MOSCEM-UA's performance would be met or exceeded by either SPEA2 or $\varepsilon$-NSGAII.

*Correspondence to:* P. Reed
(preed@engr.psu.edu)

## 1 Introduction

The hydrological behavior of a watershed can be conceptualized as a collection of spatially distributed and highly interrelated water, energy and vegetation processes. Any computer-based model of watershed behavior must, therefore, implement this conceptualization using appropriately coupled systems of parametric mathematical functions; with parameters allowing for the ability to adapt the model to different (but conceptually similar) watersheds. These parameterizations can be of varying complexity, but are, by definition, much simpler than nature itself. Model parameters therefore often become effective parameters that are related to, but not identical with measurable watershed characteristics and have to be estimated by calibrating the model to observed watershed behavior (e.g. streamflow) to account for this discrepancy. Traditional manual calibration methods use trial-and-error based analyses, which are time consuming and difficult to implement for multiple performance objectives (e.g., capturing high flow, average flow, and low flow simultaneously). There is a large body of recent water resources literature analyzing alternative tools and strategies for automatic calibration using simulation-optimization frameworks (Duan et al., 1992; Gan and Biftu, 1996; Yapo et al., 1996, 1998; Kuczera, 1997; Gupta et al., 1998; Boyle et al., 2000; Madsen, 2000; Madsen et al., 2002). Early studies (Duan et al., 1992) have highlighted that in the context of optimization, the calibration problem is ill-posed, often highly nonlinear, non-convex, and multimodal (i.e., numerous local optima exist). These problem properties have motivated several prior studies to use heuristic-based optimization, and in particular evolutionary algorithms because they have been shown to work well on

Published by Copernicus GmbH on behalf of the European Geosciences Union.

nonlinear, nonconvex, and multimodal problems (Goldberg, 1989; Duan et al., 1992; Schwefel, 1995).

Advances in computational capabilities have led to more complex hydrologic models often predicting multiple hydrologic fluxes simultaneously (e.g. surface and subsurface flows, energy). In addition, the use of an identification framework based on a single objective function is based on the erroneous assumption that all the available information regarding one hydrologic variable can be summarized (in a recoverable form) using a single aggregate measure of model performance, leading unavoidably to the loss of information and therefore poor discriminative power (Wagener and Gupta, 2005). These issues have led to an increasing interest in multi-objective optimization frameworks. The growing body of research in the area of multiobjective calibration (Gupta et al., 1998; Boyle et al., 2000; Madsen, 2000, 2003; Seibert, 2000; Wagener et al., 2001; Madsen et al., 2002; Vrugt et al., 2003a) has shown that the multiobjective approach is practical, relatively simple to implement, and can provide insights into parameter uncertainty as well as the limitations of a model (Gupta et al., 1998). Although a majority of prior studies have focused on conceptual rainfall-runoff applications, there are an increasing number of recent studies focusing on developing multiobjective calibration strategies for distributed hydrologic models (Madsen, 2003; Ajami et al., 2004; Muleta and Nicklow, 2005a, b; Vrugt et al., 2005). Calibrating a distributed hydrologic model remains a challenging problem because distributed hydrologic models have more complex structures and significantly larger parameter sets that must be specified. Moreover, distributed models are computationally expensive, causing automatic calibration to be subject to severe computational time constraints.

There is also a hidden cost in using evolutionary algorithms for hydrologic model calibration that has not been well addressed in the water resources literature. For increasingly complex models with larger parameter sets a single evolutionary multiobjective optimization (EMO) algorithm trial run may take several days or longer. Users must carefully consider how EMO algorithms' search parameters (i.e., population size, run length, random seed, etc.) impact their performance. Moreover, all of the algorithms perform stochastic searches that can attain significantly different results depending on the seeds specified in their random number generators. When a single EMO trial run takes several days, trial-and-error analysis of the performance impacts of EMO algorithm parameters or running the algorithm for a distribution of random trials can take weeks, months, or even years of computation. The increasing size and complexity of calibration problems being considered within the water resources literature necessitate rapid and reliable search.

The purpose of this study is to comprehensively assess the efficiency, effectiveness, reliability, and ease-of-use of current EMO tools for hydrologic model calibration. The following EMO algorithms are tested: Epsilon Dominance Nondominated Sorted Genetic Algorithm-II ($\varepsilon$-NSGAII) (Kollat and Reed, 2005b), the Multiobjective Shuffled Complex Evolution Metropolis algorithm (MOSCEM-UA) (Vrugt et al., 2003a), and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler et al., 2001). $\varepsilon$-NSGAII is a new algorithm developed by Kollat and Reed (2005a) that has been shown to be capable of attaining superior performance relative to other state-of-the-art EMO algorithms, including SPEA2 and $\varepsilon$-NSGAII's parent algorithm NSGAII developed by Deb et al. (2002). The performance of $\varepsilon$-NSGAII is being tested relative to MOSCEM-UA and SPEA2 because these algorithms provide performance benchmarks within the fields of water resources and computer science, respectively. This study contributes a rigorous statistical assessment of the performances of these three evolutionary multiobjective algorithms using a formal metrics-based methodology.

This study bridges multiobjective calibration hydrologic research where MOSCEM-UA (Vrugt et al., 2003a, b) represents a benchmark algorithm and EMO research where SPEA2 (Coello Coello et al., 2002) is a benchmark algorithm. Three test cases are used to compare the algorithms' performances. The first test case is composed of a standardized suite of computer science test problems (Zitzler et al., 2000; Deb, 2001; Coello Coello et al., 2002) that are used to validate the algorithms' abilities to perform global search effectively, efficiently, and reliably for a broad range of problem types. This is the first study to test MOSCEM-UA on this suite of problems. The second test case is a benchmark hydrologic calibration problem in which the Sacramento soil moisture accounting model (SAC-SMA) is calibrated for the Leaf River watershed located close to Collins, Mississippi. The Leaf River case study has been used in the development of both single and multi-objective calibration tools and specifically MOSCEM-UA (Duan et al., 1992; Yapo et al., 1998; Boyle et al., 2000; Wagener et al., 2001; Vrugt et al., 2003a, b). The third test case assesses the algorithms' performances for a computationally intensive integrated hydrologic model calibration application for the Shale Hills watershed located in the Susquehanna River Basin in north central Pennsylvania. The Shale Hills test case demonstrates the computational challenges posed by the paradigmatic shift in environmental and water resources simulation tools towards highly nonlinear physical models that seek to holistically simulate the water cycle. A challenge and contribution of this work is the development of a methodology for comprehensively comparing EMO algorithms that have different search operators and randomization techniques.

## 2 Multiobjective optimization: terms and tools

### 2.1 Multiobjective optimization terminology

There is a growing body of water resources literature (Horn and Nafpliotis, 1993; Ritzel et al., 1994; Cieniawski et al.,

1995; Halhal et al., 1997; Loughlin et al., 2000; Reed et al., 2001; Erickson et al., 2002; Reed and Minsker, 2004) demonstrating the importance of multiobjective problems (MOPs) and evolutionary multiobjective solution tools. A key characteristic of MOPs is that optimization cannot consider a single objective because performance in other objectives may suffer. Optimality in the context of multiobjective global optimization was originally defined by and named after Vilfredo Pareto (Pareto, 1896). A solution $X^*$ is classified as Pareto optimal when there is no feasible solution $X$ that will improve some objective values without degrading performance in at least one other objective. More formally, solution $X^* \in \Omega$ is Pareto optimal if for each $X \in \Omega$ and $I = \{1, 2, ..., n\}$, either

$$f_i(X) \geq f_i(X^*) \quad (\forall i \in I) \tag{1}$$

or, there is at least one $i \in I$ so that

$$f_i(X^*) < f_i(X) \tag{2}$$

where $I$ is a set of integers that range from one to the number of total objectives $n$, $X$ and $X^*$ are vectors of decision variables, $\Omega$ is the decision space, and $f_i$ is the value of a specific objective function. The definition here is based on the assumption that the optimization problem is formulated to minimize all objective values.

Equations (1) and (2) state that a Pareto optimal solution $X^*$ has at least one smaller (better) objective value compared to any other feasible solution $X$ in the decision space while performing as well or worse than $X$ in all remaining objectives. As the name implies, Pareto set is the set of Pareto optimal solutions. The Pareto front ($PF^*$) is the mapping of Pareto optimal set from the decision space to the objective space. In other words, the Pareto front is composed of a set of objective vectors which are not dominated by any other objective vectors in the objective space.

## 2.2 Evolution-based multiobjective search

Schaffer (1984) 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 (for a detailed review see Coello Coello et al., 2002). This study contributes the first comprehensive comparative analysis of these algorithms' strengths and weaknesses in the context of hydrologic model calibration. The next sections give a brief overview of each tested algorithm as well as a discussion of their similarities and differences. For detailed descriptions, readers should reference the algorithms' original published descriptions (Zitzler et al., 2001; Vrugt et al., 2003a, b; Kollat and Reed, 2005b).

### 2.2.1 Epsilon Dominance NSGAII ($\varepsilon$-NSGAII)

The $\varepsilon$-NSGAII exploits $\varepsilon$-dominance archiving (Laumanns et al., 2002; Deb et al., 2003) in combination with automatic parameterization (Reed et al., 2003) for the NSGA-II (Deb et al., 2002) to accomplish the following: (1) enhance the algorithm's ability to maintain diverse solutions, (2) automatically adapt population size commensurate with problem difficulty, and (3) allow users to sufficiently capture tradeoffs using a minimum number of design evaluations. A sufficiently quantified trade-off can be defined as a subset of Pareto optimal solutions that provide an adequate representation of the Pareto frontier that can be used to inform decision making. Kollat and Reed (2005b) performed a comprehensive comparison of the NSGA-II, SPEA2, and their proposed $\varepsilon$-NSGAII on a 4-objective groundwater monitoring application, where the $\varepsilon$-NSGAII was easier to use, more reliable, and provided more diverse representations of tradeoffs.

As an extension to NSGA-II (Deb et al., 2002), $\varepsilon$-NSGAII adds the concepts of $\varepsilon$-dominance (Laumanns et al., 2002), adaptive population sizing, and a self termination scheme to reduce the need for parameter specification (Reed et al., 2003). The values of $\varepsilon$, specified by the users represent the publishable precision or error tolerances for each objective. A high precision approximation of the Pareto optimal set can be captured by specifying very small precision tolerances $\varepsilon$. The goal of employing $\varepsilon$-dominance is to enhance the coverage of nondominated solutions along the full extent of an application's tradeoffs, or in other words, to maintain the diversity of solutions. $\varepsilon$-NSGAII is binary coded and real coded. In this application, the real coded version of the $\varepsilon$-NSGAII proposed by Kollat and Reed (2005b) is employed. The $\varepsilon$-NSGAII uses a series of "connected runs" where small populations are exploited to pre-condition the search with successively adapted population sizes. Pre-conditioning occurs by injecting current solutions within the epsilon-dominance archive into the initial generations of larger population runs. This scheme bounds the maximum size of the population to four times the number of solutions that exist at the user specified $\varepsilon$ resolution. Theoretically, this approach allows population sizes to increase or decrease, and in the limit when the epsilon dominance archive size stabilizes, the $\varepsilon$-NSGAII's "connected runs" are equivalent to time continuation (Goldberg, 2002). (i.e., injecting random solutions when search progress slows down). For more details about $\varepsilon$-dominance or the $\varepsilon$-NSGAII, please refer to the following studies (Laumanns et al., 2002; Kollat and Reed, 2005a, b).

There are 4 major parameters that need to be specified for $\varepsilon$-NSGAII (1) the probability of mating, (2) the probability of mutation, (3) the maximum run time, and (4) the initial population size. The mating and mutation operators and parameters are discussed in more detail in Sect. 2.2.4. The maximum run time is defined as the upper limit on the time the user is willing to invest in search. Although epsilons must be specified for every objective, these values are defined by the

properties of the application not the evolutionary algorithm. In any optimization application, it is recommended that the user specify the publishable precision or error tolerances for their objectives to avoid wasting computational resources on unjustifiably precise results.

#### 2.2.2 The Strength Pareto Evolutionary Algorithm 2 (SPEA2)

SPEA2 represents an improvement from the original Strength Pareto Evolutionary Algorithm (Zitzler and Thiele, 1999; Zitzler et al., 2001). SPEA2 overcomes limitations of the original version of the algorithm by using an improved fitness assignment, bounded archiving, and a comprehensive assessment of diversity using k-means clustering. SPEA2 requires users to specify the upper bound on the number of nondominated solutions that are archived. If the number of non-dominated solutions found by the algorithm is less than the user-specified bound then they are copied to the archive and the best dominated individuals from the previous generation are used to fill up the archive. If the size of non-dominated set is larger than the archive size, a k-means clustering algorithm comprehensively assesses the distances between archive members. A truncation scheme promotes diversity by iteratively removing the individual that has the minimum distance from its neighbouring solutions. The archive update strategy in SPEA2 helps to preserve boundary (outer) solutions and guide the search using solution density information. SPEA2 has 5 primary parameters that control the algorithm's performance: (1) population size, (2) archive size, (3) the probability of mating, (4) the probability of mutation, and (5) the maximum run time. For a more detailed description, see the work of Zitzler et al. (Zitzler and Thiele, 1999; Zitzler et al., 2001)

#### 2.2.3 Multiobjective Shuffled Complex Evolution Metropolis (MOSCEM-UA)

MOSCEM-UA was developed by Vrugt et al. (2003a). The algorithm combines a Markov Chain Monte Carlo sampler with the Shuffle Complex Evolutionary algorithm (SCE-UA) algorithm (Duan et al., 1992), while seeking Pareto optimal solutions using an improved fitness assignment approach based on the original SPEA (Zitzler and Thiele, 1999). It modifies the fitness assignment strategy of SPEA to overcome the drawback that individuals dominated by the same archive members are assigned the same fitness values (Zitzler et al., 2001; Vrugt et al., 2003a). MOSCEM-UA combines the complex shuffling of the SCE-UA (Duan et al., 1992, 1993) with the probabilistic covariance-annealing process of the Shuffle Complex Evolution Metropolis-UA algorithm (Vrugt et al., 2003b). Firstly, a uniformly distributed initial population is divided into complexes within which parallel sequences are created after sorting the population based on fitness values. Secondly, the sequences are evolved iteratively toward a multivariate normally distributed set of solutions. The moments (mean and covariance matrix) of the multivariate distribution change dynamically because they are calculated using the information from the current evolution stage of sequences and associated complexes. Finally, the complexes are reshuffled before the next sequence of evolution. For a detailed introduction to the algorithm, please refer to the research of Vrugt et al. (2003a, b).

Based on the findings of Vrugt et al. (2003a) and our own analysis, MOSCEM-UA's performance is most sensitive to three parameters: (1) population size, (2) run length, and (3) the number of complexes/sequences. All of the remaining parameters (i.e., reshuffling and scaling) were set to the default values in a C source version of the algorithm we received from Vrugt in June 2004. Readers should also note that while MOSCEM-UA and SCE-UA use some of the same underlying search operators, their algorithmic structures and implementations are very different. The analysis and conclusions of this study apply only to the MOSCEM-UA algorithm.

#### 2.2.4 Similarities and differences between the algorithms

$\varepsilon$-NSGAII, SPEA2, and MOSCEM-UA all seek the Pareto optimal set instead of a single solution. Although these algorithms employ different methodologies, ultimately they all seek to balance rapid convergence to the Pareto front with maintaining a diverse set of solutions along the full extent of an application's tradeoffs. Diversity preservation is also important for limiting premature-convergence to poor approximations of the true Pareto set. The primary factors controlling diversity are population sizing, fitness assignment schemes that account for both Pareto dominance and diversity, and variational operators for generating new solutions in unexplored regions of a problem space.

A key characteristic of $\varepsilon$-NSGAII is the algorithm's ability to adapt population size commensurate to problem difficulty and promote diversity using "time continuation" (i.e., injecting random solutions when search progress slows). Both SPEA2 and MOSCEM-UA are impacted by population size, but currently trial-and-error analysis is necessary to determine an appropriate search population size. With respect to the fitness assignment, these three algorithms all use the Pareto dominance concept. Both MOSCEM-UA and SPEA2 use the fitness assignment method based on the original fitness assignment approach employed in SPEA. MOSCEM-UA improves the original method by adding Pareto rank when assigning fitness values to dominated individuals in the population. SPEA2 considers both dominated and nondominated individuals as well as their density information when applying fitness assignment. The density function is used to differentiate individuals with the same raw fitness values by calculating the distance from current point being considered to a predefined nearest point (Zitzler et al., 2001). $\varepsilon$-NSGAII

**Table 1.** Suite of test functions.

| Names of Test Functions | Number of Decision Variables and Parameter Ranges | Main Features of the Pareto optimal front |
|---|---|---|
| $T_1$ | m=30; [0, 1] | Convex |
| $T_2$ | m=30; [0, 1] | Non-Convex counterpart to $T_1$ |
| $T_3$ | m=30; [0, 1] | Discreteness: Multiple non-contiguous convex parts |
| $T_4$ | m=10; [0, 1] for the first variable, $[-5, 5]$ for others | Multimodality: $21^9$ local fronts |
| $T_6$ | m=10; [0, 1] | Solutions are non-uniformly distributed; Solution density is lowest near the front and highest away from the front |

adopts the $\varepsilon$-dominance grid based approach for fitness assignment and diversity preservation (Laumanns et al., 2002).

Regarding the whole evolution process, MOSCEM-UA is significantly different from SPEA2 and $\varepsilon$-NSGAII although all of them randomly initialize their search populations. As discussed above, MOSCEM-UA uses the complex shuffling method and the Metropolis-Hastings algorithm to conduct search. Offspring are generated using a multivariate normal distribution developed utilizing information from the current draw of the parallel sequence within a complex. The acceptance of a new generated candidate solution is decided according to the scaled ratio of candidate solution's fitness to current draw's fitness of the sequence. Complex shuffling helps communication between different complexes and promotes solution diversity.

Comparatively, SPEA2 and $\varepsilon$-NSGAII adopt the traditional evolutionary operators (e.g. selection, crossover and mutation) in searching. They both use binary tournament selection, simulated binary crossover (SBX), and polynomial mutation. And both of them maintain external archives which store the best solutions found from the random initial generation to final termination generation. However, these two algorithms are different in many aspects. After population initialization, SPEA2 assigns fitness to each individual in the population and the archive. Nondominated sorting is conducted on all these individuals and then the nondominated solutions are copied to the archive of next generation. Because the archive is fixed in size, either a truncation scheme must be implemented or the best dominated solutions must be used to fill up the archive. Then binary tournament selection with replacement is applied to select individuals for a mating pool. The new population in SPEA2's next generation is created by applying crossover and mutation operators to the mating pool. The process is repeated until a user specified termination criterion is met.

$\varepsilon$-NSGAII initiates the search with an arbitrarily small number of individuals (e.g., 10-individuals). Binary tournament selection, SBX crossover, and mutation operators are implemented to generate the first child population. Pareto ranks are assigned to the individuals from the parent and children populations. Solutions are selected preferentially based on their non-domination rank. Crowding distances (i.e., Euclidean norms for measuring distance from neighbour solutions in objective space) are used to distinguish between the individuals with the same non-domination rank (i.e., larger crowding distances are picked preferentially to promote diversity). At the end of each generation, the external archive is updated with the $\varepsilon$-non-dominated solutions. The archive size and population size change dynamically based on the total number of $\varepsilon$ non-dominated solutions stored. In this study, a single termination criterion based on the maximum number of function evaluations was used for all of the algorithms (i.e., they all had identical numbers of function evaluations) to ensure a fair comparison.

## 3 Case studies

### 3.1 Case study 1: the test function suite

The first test case is composed of a standardized suite of computer science test problems (Zitzler et al., 2000; Deb, 2001; Coello Coello et al., 2002) that are used to validate the algorithms' abilities to perform global search effectively, efficiently, and reliably for a broad range of problem types. This is the first study to test MOSCEM-UA on this suite of problems. The test function suite has been developed collaboratively by the EMO community (Coello Coello et al., 2002; Deb et al., 2002) as standardized performance tests where new algorithms must meet or exceed the performance of current benchmark algorithms such as SPEA2.

Since these test functions have been used very broadly in the EMO literature (Zitzler et al., 2000; Deb, 2001; Coello Coello et al., 2002; Kollat and Reed, 2005a), their detailed formulations will not be presented here. Table 1 provides an overview of the number of decision variables used, their ranges, and the problems' characteristics. The test functions

are labeled T1, T2, T3, T4, and T6 following the naming convention of Zitzler et al. (2000). All of the test functions have been implemented in the standard forms used in the EMO literature. Generally, T1 and T2 are considered relatively straightforward convex and non-convex test problems. T3 tests algorithms' abilities to find discontinuous convex sets of solutions. T4 and T6 are the most challenging of the test functions requiring algorithms to overcome large numbers of local fronts and non-uniformly distributed solution spaces, respectively.

### 3.2 Case study 2: Leaf River watershed

The Leaf River SAC-SMA test case represents a benchmark problem within the water resources literature that has been used extensively for developing tools and strategies for improving hydrologic model calibration (Duan et al., 1992; Yapo et al., 1998; Boyle et al., 2000; Wagener et al., 2001; Vrugt et al., 2003a, b). Readers interested in the full details of the Leaf River case study's dataset should reference earlier works (e.g., Sorooshian et al., 1993). The Leaf River case study used in this paper has been developed based on the original studies used to develop and demonstrate MOSCEM-UA (Vrugt et al., 2003a, b). The Sacramento Soil Moisture Accounting model is a 16 parameter lumped conceptual watershed model used for operational river forecasting by the National Weather Service throughout the US (see Burnash, 1995, for more details on the model). All three algorithms searched the same 13 SAC-SMA parameters (3 parameters are commonly fixed a priori) and parameter ranges as were specified by Vrugt et al. (2003a). The algorithms were tested on their ability to quantify a 2-objective tradeoff based on a root-mean square error (RMSE) problem formulation. The first objective was formulated using a Box-Cox transformation of the hydrograph ($z=[(y+1)^{\lambda}-1]/\lambda$ where $\lambda=0.3$) as recommended by Misirli et al. (2003) to reduce the impacts of heteroscedasticity in the RMSE calculations (also increasing the influence of low flow periods). The second objective was the non-transformed RMSE objective, which is largely dominated by peak flow prediction errors due to the use of squared residuals. The best known approximation set generated for this problem is discussed in more detail in the results of this study (see Fig. 5a).

A 65-day warm-up period was used based on the methodological recommendations of Vrugt et al. (2003a). A two-year calibration period was used from 1 October 1952 to 30 September 1954. The calibration period was shortened for this study to control the computational demands posed by rigorously assessing the EMO algorithms. A total of 150 EMO algorithm trial runs were used to characterize the algorithms (i.e., 50 trials per algorithm). Each EMO algorithm trial run utilized 100 000 SAC-SMA model evaluations, yielding a total of 15 000 000 SAC-SMA model evaluations used in our Leaf River case study analysis. Reducing the calibration period improved the computational tractability of our analysis. The focus of this study is on assessing the performances of the three EMO algorithms that are captured in the 2 year calibration period. In actual operational use of the SAC-SMA for the Leaf River 8 to 10 year calibration periods are used to account for climatic variation between years (Boyle et al., 2000).

### 3.3 Case study 3: Shale Hills watershed

The Shale Hills experimental watershed was established in 1961 and is located in the north of Huntington County, Pennsylvania. It is located within the Valley and Ridge province of the Susquehanna River Basin in north central Pennsylvania. The data used in this study was supplied by a comprehensive hydrologic experiment conducted in 1970 on a 19.8 acre sub-watershed of the Shale Hill experimental site. The experiment was led by Jim Lynch of the Pennsylvania State University's Forestry group with the purpose of exploring the physical mechanisms of the formation of stream-flow at the upland forested catchment and to evaluate the impacts of antecedent soil moisture on both the volume and timing of the runoff (see Duffy, 1996). The experiment was composed of an extensive below canopy irrigation network for simulating rainfall events as well as a comprehensive piezometer network, 40 soil moisture neutron access tubes and 4 weirs for measuring flow in the ephemeral channel. Parameterization of the integrated surface-subsurface model for the Shale Hills was also supported by more recent site investigations, where Lin et al. (2005) extensively characterized the soil and groundwater properties of the site using in-situ observations and ground penetrating radar investigations.

#### 3.3.1 Integrated surface-subsurface model description

The hydrologic model being calibrated in this study is a semi-distributed version of the integrated hydrologic model being developed by Duffy et al. (1996, 2004), Qu (2004). This model integrates watershed processes within the terrestrial hydrologic cycle over a wide range of time scales. It couples surface, subsurface and channel states within the hillslope and watershed. The model strategy is to transform partial differential equations (PDEs) to ordinary differential equations (ODEs), using the semi-discrete finite volume method (SD_FVM) (Duffy, 2004). Specifically, the spatial domain is decomposed into different zones (response units). Different ODEs are created to simulate different hydrologic processes within each zone. The ODE system within each zone is termed a "Model Kernel". An overall ODE system is created by combining all of the model kernels. The ODE system is solved using an implicit Runge-Kutta ODE solver (RADAU IIA) of order 5 (Hairer and Wanner, 1996). As noted by Duffy (1996, 2004), by taking advantage of the finite volume method, the model strategy has the capability of capturing the "dynamics" in different processes while maintaining the water balance (Qu, 2004). The model also has the

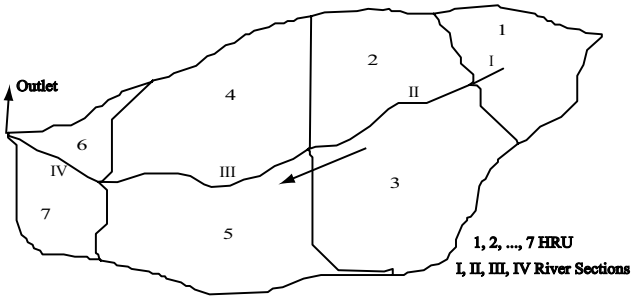


**Fig. 1.** Domain decomposition of the Shale Hills test case.

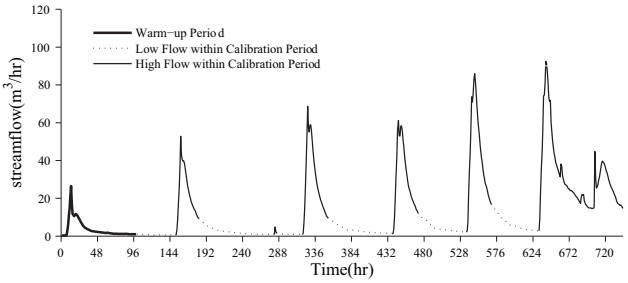


**Fig. 2.** Illustration of the Shale Hills calibration period where a 100 h warm up period was used. High flow and low flow classifications were made based on the points of inflection within the hydrograph.

flexibility of easily adding/eliminating (switching on/off) the key hydrologic processes for a system.

As discussed above, the water budget is computed using a global model kernel composed of ODEs representing each of the watershed zones or river sections. The number of ODEs increases linearly with the number of decomposed spatial zones within the watershed. In the Shale Hills application, the watershed is decomposed into 7 zones and 4 river sections connected to each other between the zones. The decomposed domain and the topology of the zones and the river sections are shown in Fig. 1. The domain decomposition results in 32 ODEs solved implicitly using a solver that has been proven to be highly effective for ODE systems (Guglielmi and Hairer, 2001). The model simulation time is substantial for this study given that the EMO algorithms will have to evaluate thousands of simulations while automatically calibrating model parameters. On a Pentium 4 Linux workstation with a 3 gigahertz processor and 2 gigabytes of RAM, a one month simulation of Shale Hills using a 1 h output time interval requires 120 s of computing time. If 5000 model evaluations are used to optimize model parameters, then a single EMO run will take almost 7 days. This study highlights how trial-and-error analysis of EMO algorithm performance can have a tremendous cost in both user and computational time.

### 3.3.2 Problem formulation

Multiobjective calibration uses multiple performance measures to improve model predictions of distinctly different responses within a watershed's hydrograph simultaneously (e.g., high flow, low flow, average flow). For the Shale Hills case study, the calibration objectives were formulated to generate alternative model parameter groups that capture high flow, average flow, and low flow conditions for the Shale Hills test case using the three search objectives given in Eqs. (3)–(5). The problem formulations used in this study build on prior research using RMSE and the heteroscedastic maximum likelihood estimator (HMLE) measures (Sorooshian and Dracup, 1980; Yapo et al., 1996, 1998; Gupta et al., 1998; Boyle et al., 2000; Madsen, 2003; Ajami et al., 2004).

$$\text{Average RMSE}: f_1(\theta) = \left[\frac{1}{N}\sum_{i=1}^{N} w_{1i}\left[Q_{\text{obs,i}} - Q_{\text{sim,i}}(\theta)\right]^2\right]^{1/2} \quad (3)$$

$$\text{High flow RMSE}: f_2(\theta) = \left[\frac{1}{\sum_{j=1}^{M_p} n_j}\sum_{j=1}^{M_p}\sum_{i=1}^{n_j} w_{2i}\left[Q_{\text{obs,i}} - Q_{\text{sim,i}}(\theta)\right]^2\right]^{1/2} \quad (4)$$

$$\text{Low flow RMSE}: f_3(\theta) = \left[\frac{1}{\sum_{j=1}^{M_l} n_j}\sum_{j=1}^{M_l}\sum_{i=1}^{n_j} w_{3i}\left[Q_{\text{obs,i}} - Q_{\text{sim,i}}(\theta)\right]^2\right]^{1/2} \quad (5)$$

where $Q_{\text{obs,i}}$ is the observed discharge at time $i$; $Q_{\text{sim,i}}(\theta)$ is the simulated discharge; $N$ is the total number of time steps in the calibration period; $M_p$ is the number of peak flow events; $M_l$ is the number of low flow events; $n_j$ is the number of time steps in peak/low flow event number $j$; $w_1$, $w_2$ and $w_3$ are the weighting coefficients; $\theta$ is the set of model parameters to be calibrated.

In this study, the weighting coefficients for high flow and low flow are adapted forms of the HMLE statistics (Yapo et al., 1996). The weights for high flow errors are set to the square of the observed discharges to emphasize peak discharge values. The weights for low flow are set to give prominence to low flow impacts on the estimation errors. The weighting coefficient for average flow is set to 1 and thus the error metric for average flow is the standard RMSE statistic. Equation (6) provides the weighting coefficients used to differentiate different hydrologic responses.

$$w_1 = 1 \quad w_2 = Q_{\text{obs}}^2 \quad w_3 = \left(\frac{1}{Q_{\text{obs}}^2}\right)^{1/\sum_{j=1}^{M_l} n_j} \quad (6)$$

Preliminary sensitivity analysis showed that the model was very sensitive to the initial surface storage, but the impacts

of the initial surface storage were attenuated within the first 100 h. Figure 2 illustrates the Shale Hills calibration period including a 100 h warm up period to reduce the impacts of the initial conditions. High flow and low flow classifications were made based on points of inflection within the hydrograph. Table 2 overviews the parameters being calibrated for the Shale Hills case study. For overland flow, the convergence time scale of a hill slope $\eta$ cannot be estimated analytically so the parameter was selected for calibration. The saturated soil hydraulic conductivity $K_s$ is calibrated as well as the empirical constants ($\alpha$, $\beta$) in the van Genuchten soil functions. In our preliminary sensitivity analysis, Manning's coefficient ($n$) and the saturated hydraulic conductivity of river reaches were identified to significantly impact river routing and groundwater-stream interactions. Both of these parameters are calibrated. In the Shale Hills case study, a total of 36 parameters are being calibrated (7 spatial zones * 4 parameters + 4 river sections * 2 parameters). The parameter ranges were specified based on both field surveys (Qu, 2004; Lin et al., 2005) and recommendations from literature (Carsel, 1988; Dingman, 2002).

## 4 Description of the computational experiment

### 4.1 Algorithm configurations and parameterizations

In an effort to ensure a fair comparison between $\varepsilon$-NSGAII and each of the other algorithms, significant effort has been focused on seeking optimal configurations and parameterizations for SPEA2 and MOSCEM-UA using trial-and-error analysis and prior literature. The broadest analysis of the impacts of alternative algorithm configurations was done for the test function suite, since this test case has the smallest computational demands. The algorithms were allotted 15 000 function evaluations for each trial run when solving each problem within the test function suite based on the recommendations and results of prior studies (Zitzler et al., 2001; Kollat and Reed, 2005a). For each problem in the test function suite a total of 350 trial runs were performed (i.e., 1 configuration for $\varepsilon$-NSGAII tested for 50 random seeds, 4 MOSCEM-UA configurations tested for 50 random seeds each yielding 200 trial runs, and 2 SPEA2 configurations tested for 50 random seeds each yielding 100 trial runs).

Since $\varepsilon$-NSGAII and SPEA2 use the same mating and mutation operators, the algorithms' probabilities of mating where set equal 1.0 and their probabilities of mutation were set equal to 1/L where L is the number of decision variables as has been recommended extensively in the literature (Zitzler et al., 2000, 2001; Deb, 2001; Coello Coello et al., 2002). $\varepsilon$-NSGAII utilized an initial population size of 10 individuals. For the test function suite SPEA2's two configurations both used an archive size of 100 based on prior studies (Zitzler et al., 2000, 2001; Deb, 2001; Coello Coello et al., 2002) and two different population sizes ($N$=100) and ($N$=250). MOSCEM-UA's configurations tested the impacts of increasing population sizes $N$ and increasing the numbers of complexes $C$: ($N$=100, $C$=2), ($N$=250, $C$=2), ($N$=250, $C$=5) and ($N$=1000, $C$=5). The largest population size and number of complexes tested for MOSCEM-UA were based on a personal communication with Jasper Vrugt, the algorithm's creator.

$\varepsilon$-NSGAII utilized the same configuration as was used for the test function suite on the Leaf River and Shale Hills case studies in an effort to test the algorithms' robustness in the absence of trial-and-error analysis. Based on the SPEA2's performance on the test function suite and trial-and-error analysis the algorithm's population size was set equal to 100 for both the Leaf River and Shale Hills test cases. The key challenge in maximizing the performance of SPEA2 lies in specifying an effective archive size without a priori knowledge of the Pareto set. SPEA2's performance is very sensitive to archive size. Trial-and-error analysis revealed that if the algorithm's archive is too small then its overall performance suffered. Moreover, setting the SPEA2 archive to be very large also reduced the algorithm's search effectiveness because its diversity enhancing clustering operator is under utilized. For the Leaf River and Shale Hills case studies, SPEA2's performance was maximized by setting the archive size equal to 500 and 100, respectively, based on the average archive sizes attained by the $\varepsilon$-NSGAII. Note $\varepsilon$-NSGAII automatically sizes its archive based on the number of $\varepsilon$-nondominated solutions that have been found.

For the Leaf River case study, MOSCEM-UA utilized a population size of 500 individuals and 10 complexes as was used by Vrugt et al. (2003a) in the original development and demonstration of the algorithm. As will be discussed in the results presented in Sect. 5 increasing the population size and number of complexes used by MOSCEM-UA has a very large impact on the algorithm's solution time, which significantly impacted our analysis of the Shale Hills test case. For the Shale Hills case study, MOSCEM-UA was tested for a population size of 250 with 2 or 5 complexes to ensure that a single run would complete in 7 days based on the maximum run times allotted for the LION-XO computing cluster. The computational constraints limiting our ability to use larger population sizes and more complexes in the Shale Hills trial runs for MOSCEM-UA are discussed in greater detail in Sect. 5.

### 4.2 Performance metrics

The performances of all of the EMO algorithms tested in this study were assessed using metrics designed to answer two questions: (1) how good are the approximation sets found by the EMO algorithms? and (2) which of the solution sets are better than the others? Deb and Jain (2002), stress that EMO performance assessments must account for two separate and often conflicting approximation set properties: (1) convergence – the distance from the reference set of opti-

**Table 2.** Parameters being optimized in the Shale Hills case study.

| Parameters | Description | Units | Min. | Max. | Kernel |
|---|---|---|---|---|---|
| $K_s$ | Saturated hydraulic conductivity | m/h | 0.000035 | 0.15 | Zone |
| $\eta$ | Surface time scale | 1/h | 0.08 | 1 | Zone |
| $\alpha$ | Empirical constant | 1/m | 0 | 7 | Zone |
| $\beta$ | Empirical constant | | 1.1 | 2 | Zone |
| $n$ | Manning's coefficient | | 0.02 | 0.08 | River Section |
| Ksr | Saturated hydraulic conductivity of river section | m/h | 0.000035 | 0.3 | River Section |

mal solutions, and (2) diversity – how well the evolved set of solutions represents the full extent of the tradeoffs that exist between an application's objectives. Performance metrics that measure these properties are termed unary indicators because their values are calculated using one solution set and they reveal specific aspects of solution quality (Zitzler et al., 2003).

Two unary metrics, the $\varepsilon$-indicator (Zitzler et al., 2003) and the hypervolume indicator (Zitzler and Thiele, 1999) were selected to assess the performances of the algorithms. The unary $\varepsilon$-indicator measures how well the algorithms converge to the true Pareto set or the best known approximation to the Pareto set. The unary $\varepsilon$-indicator represents the smallest distance that an approximation set must be translated to dominate the reference set, so smaller indicator values are preferred. For example, in Fig. 3, the approximation set has to be translated a distance of $\varepsilon$ so that it dominates the reference set. The unary hypervolume metric measures how well the algorithms performed in identifying solutions along the full extent of the Pareto surface or its best known approximation (i.e., solution diversity). The unary hypervolume metric was computed as the difference between the volume of the objective space dominated by the true Pareto set and volume of the objective space dominated by the approximation set. For example, the blue shaded area in Fig. 3 represents the hypvervolume metric of the approximation set. Ideally, the hypervolume metric should be equal to zero. For more details about the descriptions and usages of these metrics, see Zitzler and Thiele (1999); Zitzler et al. (2003); Kollat and Reed (2005b).

In addition to the unary metrics discussed above, performance was also assessed using the binary metric. The binary metric was implemented by combining the unary $\varepsilon$-indicator metric with an interpretation function. Zitzler et al. (2003) formulated the interpretation function to directly compare two approximation solution sets and conclude which set is better or if they are incomparable. The term "binary" refers to the metric's emphasis on comparing the quality of two approximation sets. The $\varepsilon$-indicator and the interpretation function are formulated as shown in Eqs. (7) and (8) separately:

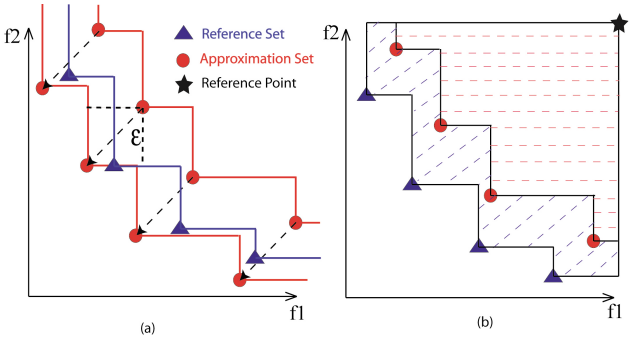


**Fig. 3. (a)** Example illustration of the $\varepsilon$-indicator metric. **(b)** Example illustration of the hypervolume metric. The shaded area with blue color represents the hypervolume value. Adapted from (Fonseca et al., 2005).

$$I_\varepsilon(A, B) = \max_{\mathbf{f}^2 \in B} \min_{\mathbf{f}^1 \in A} \max_{1 \le i \le n} \frac{f_i^1}{f_i^2} \tag{7}$$

$$F = (I_\varepsilon(A, B) \le 1 \wedge I_\varepsilon(B, A) > 1) \tag{8}$$

Where $\mathbf{f}^1 = \{f_1^1, f_2^1, ..., f_n^1\} \in A$ and $\mathbf{f}^2 = \{f_1^2, f_2^2, ..., f_n^2\} \in B$ are objective vectors; $A$ and $B$ are two approximation sets; $F$ is an interpretation function. If $A$ is not better than $B$ and $B$ is not better than $A$, then the sets are incomparable. When $F$ is true, it indicates that $A$ is better than $B$. Similarly, changing the order of $A$ and $B$, the decision about whether $B$ is better than $A$ can be made.

The binary $\varepsilon$-indicator metric provides a direct way of ranking the quality of approximation sets generated using different initial random populations and/or different algorithm configurations. The results of each trial run are compared to the results of all other trial runs in the comparison pool. Each trial run is given a rank according to the number of trial runs that exceed its performance in terms of the binary $\varepsilon$-indicator metric. The best trial runs are assigned a rank of one, while a rank of two is assigned to the trial runs that have the second best results. The process is repeated until every trial run is assigned a rank. The trial runs in the same rank

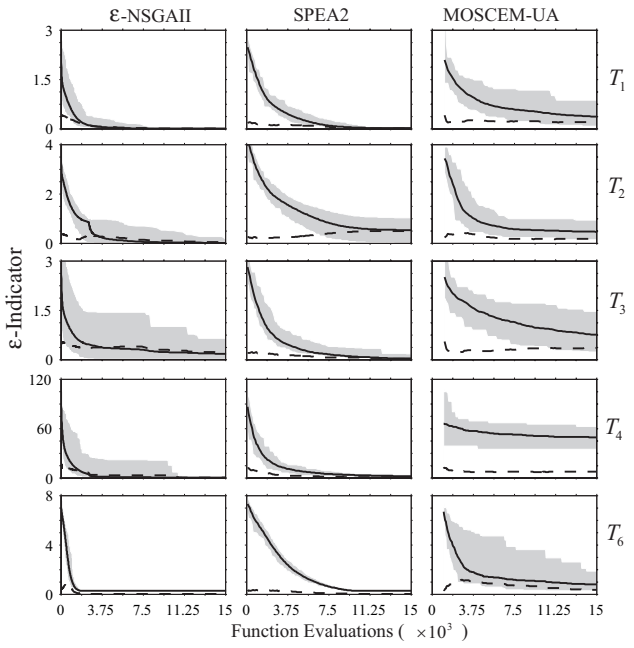


**Fig. 4.** Dynamic performance plot for the unary $\varepsilon$-indicator distance metric versus total design evaluations for the best performing configurations of the $\varepsilon$-NSGAII, SPEA2, and MOSCEM. Mean performance is indicated by a solid line, the standard deviation by a dashed line, and the range of performance by the shaded region. The plots were generated using 50 trials for each algorithm.

are incomparable to one another. In this study, the binary $\varepsilon$-indicator ranking results are presented in terms of the ratio of trial runs that attain top ranks (i.e., ranks of 1 or 2).

## 5 Results

### 5.1 Optimization results for case study 1: test function suite

As described in Sect. 4.2, the binary $\varepsilon$-indicator metric provides performance rankings for alternative algorithm configurations and cross-algorithm performance. For each test problem a total of 350 trial runs were performed (i.e., 1 configuration for $\varepsilon$-NSGAII tested for 50 random seeds, 4 MOSCEM-UA configurations yielding 200 random seed trials, and 2 SPEA2 configurations yielding 100 random seed trials). After ranking the trial runs, we present the ratio of the number of top ranking runs out of the 50 trials used to test each of the algorithms' configurations (see Table 3).

The best configurations for SPEA2 and MOSCEM-UA are ($N$=100) and ($N$=1000, $C$=5), respectively. The $\varepsilon$-NSGAII has the best overall binary $\varepsilon$-indicator metric rankings for the test function suite.

The unary hypervolume and $\varepsilon$-indicator metrics measure solution diversity and algorithm convergence to the true Pareto fronts, respectively. These unary metrics provide a more detailed understanding of the dynamic performances of the algorithms in terms of efficiency, effectiveness, and reliability. The means and standard deviations of the final optimization results for the best configurations ($\varepsilon$-NSGAII has only one configuration) are summarized in Table 4.

Recall that the unary $\varepsilon$-indicator represents the smallest distance that an approximation set must be translated to dominate the reference set so smaller indicator values are preferred. Likewise, the unary hypervolume metric is the difference between the volume of the objective space dominated by the true Pareto set and volume of the objective space dominated by the approximation set. Ideally, the hypervolume metric should be equal to zero.

In Table 4, the $\varepsilon$-NSGAII has the best overall average performance in both metrics for the test functions. In addition, the relatively small standard deviations reveal that $\varepsilon$-NSGAII is reliable in solving the test functions. SPEA2 is also effective and reliable in solving the test functions. Both $\varepsilon$-NSGAII and SPEA2 are superior to MOSCEM-UA. Figure 4 illustrates the variability in the algorithms' performances by presenting runtime results for the $\varepsilon$-indicator distance metric.

The plots show the results of all 50 random seed trials with 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.

Figure 4 confirms that $\varepsilon$-NSGAII was both the most efficient and effective of the algorithms attaining very close approximations of the true Pareto sets in under 2500 evaluations. SPEA2 typically requires 7500 evaluations to attain equivalent metric values relative to $\varepsilon$-NSGAII. MOSCEM is the least reliable and efficient of the algorithms for the test function suite, failing to attain competitive results in 15 000 evaluations. Dynamic plots of the hypervolume metric showed very similar results to the runtime unary $\varepsilon$-indicator results shown in Fig. 4. The most significant performance differences between the algorithms resulted for the multi-modal T4 problem. The performance rankings in Table 3 show that MOSCEM-UA generally failed to converge to the Pareto front for T4. SPEA2's dynamic search results for T4 (see Fig. 4) are much better than MOSCEM-UA but its final solution set is still far away from the Pareto front as evidenced by its poor ranking results in Table 3. Only $\varepsilon$-NSGAII successfully converges to the true Pareto front for T4 with high reliability. In terms of elapsed computational time, the $\varepsilon$-NSGAII is an order of magnitude faster than that of SPEA2, and the elapsed computational time of SPEA2 is an order of magnitude faster than MOSCEM-UA. For example, in solving T1, the average computational times required by $\varepsilon$-NSGAII, SPEA2 and MOSCEM-UA are 1.90 s, 21.75 s, and 397.42 s, respectively. Note this difference in computational efficiency had dramatic impacts on the com-

**Table 3.** Test function results for the ratios of top trial runs for each configuration of the algorithms based on the binary $\varepsilon$-indicator metric ranking. The values highlighted by bold font are the best values among the configurations within a specific algorithm, the values indicated by bold font with underscore are the best values across algorithms.

| MOEA | Configurations | Top Ranking Ratios | | | | |
|---|---|---|---|---|---|---|
| | | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_6$ |
| $\varepsilon$-NSGAII | ($N$=10) | **50/50** | **50/50** | **50/50** | **50/50** | **50/50** |
| SPEA2 | ($N$=100) | **50/50** | **9/50** | **50/50** | **1/50** | **47/50** |
| | ($N$=250) | 50/50 | 6/50 | 45/50 | 0/50 | 27/50 |
| MOSCEM-UA | ($N$=100, $C$=2) | 0/50 | 0/50 | 0/50 | 0/50 | 6/50 |
| | ($N$=250, $C$=2) | 1/50 | 0/50 | 0/50 | 0/50 | 12/50 |
| | ($N$=250, $C$=5) | 0/50 | 0/50 | 0/50 | 0/50 | 14/50 |
| | ($N$=1000, $C$=5) | **11/50** | 0/50 | 0/50 | 0/50 | **20/50** |

**Table 4.** Averages and standard deviations of the unary metrics for each algorithm's best configuration. AVG stands for mean, STD stands for standard deviation, and bolded entries highlight the best value attained.

| | MOEA | Hypervolume | | $\varepsilon$-Indicator | | Time (s) | |
|---|---|---|---|---|---|---|---|
| | | AVG | STD | AVG | STD | AVG | STD |
| $T_1$ | $\varepsilon$-NSGAII | **1.43E-4** | 7.50E-5 | **4.12E-3** | 2.02E-3 | 1.90E+0 | 1.22E+0 |
| | SPEA2 | 1.31E-2 | 2.12E-3 | 1.61E-2 | 2.40E-3 | 2.18E+1 | 8.40E-1 |
| | MOSCEM-UA | 6.69E-1 | 4.73E-1 | 3.62E-1 | 1.97E-1 | 3.97E+2 | 1.66E+2 |
| $T_2$ | $\varepsilon$-NSGAII | **2.91E-4** | 1.58E-3 | **9.85E-3** | 2.22E-2 | 1.14E+0 | 8.60E-1 |
| | SPEA2 | 5.30E-1 | 1.63E-1 | 5.30E-1 | 4.95E-1 | 1.13E+1 | 7.90E-1 |
| | MOSCEM-UA | 5.11E-1 | 2.22E-1 | 4.60E-1 | 1.81E-1 | 2.96E+2 | 1.85E+1 |
| $T_3$ | $\varepsilon$-NSGAII | 3.78E-2 | 5.52E-2 | 1.71E-1 | 2.10E-1 | 1.70E+0 | 1.18E+0 |
| | SPEA2 | **2.61E-2** | 9.60E-3 | **3.08E-2** | 2.28E-2 | 2.12E+1 | 6.00E-1 |
| | MOSCEM-UA | 1.08E0 | 4.31E-1 | 7.51E-1 | 3.49E-1 | 3.07E+2 | 2.21E+1 |
| $T_4$ | $\varepsilon$-NSGAII | **1.73E-2** | 4.23E-2 | **2.33E-2** | 4.83E-2 | 2.34E+0 | 1.61E+0 |
| | SPEA2 | 1.65E+0 | 6.06E-1 | 1.93E+0 | 6.59E-1 | 2.34E+1 | 6.40E-1 |
| | MOSCEM-UA | 5.10E+1 | 6.69E+0 | 4.94E+1 | 7.29E+0 | 7.33E+2 | 8.96E+1 |
| $T_6$ | $\varepsilon$-NSGAII | **1.51E-2** | 1.57E-3 | 2.81E-1 | 1.68E-4 | 1.42E+0 | 7.90E-1 |
| | SPEA2 | 4.23E-2 | 4.42E-3 | **2.81E-1** | 0.00E+0 | 2.62E+1 | 2.90E+0 |
| | MOSCEM-UA | 1.48E+0 | 1.07E+0 | 7.84E-1 | 3.39E-1 | 5.52E+2 | 1.67E+2 |

putational times required for our test function analysis, where several days were required for MOSCEM-UA, several hours for SPEA2, and several minutes for $\varepsilon$-NSGAII.

Averaged performance metrics are meaningful only in cases when the EMO algorithms' metric distributions are significantly different from one another. In this study, the Mann-Whitney test (Conover, 1999) was used to validate that the algorithms attained statistically significant performance differences. The null hypothesis for the tests assumed that metric distributions for any two algorithms are the same. The Mann-Whitney test showed a greater than 99% confidence that performance metric scores for the $\varepsilon$-NSGAII are significantly different from those of MOSCEM-UA for all of the test functions. When comparing SPEA2 and MOSCEM-UA it was found that the algorithms' performance differences on T2 are not statistically significant. On all of the remaining test functions SPEA2's superior performance relative to MOSCEM-UA was validated at greater than a 99% confidence level. The $\varepsilon$-NSGAII's performance was statistically superior to SPEA2 at the 99% confidence level for all of the test functions except for T3. $\varepsilon$-NSGAII and SPEA2 did not attain a statistically meaningful performance difference on T3.

### 5.2 Optimization results for case study 2: leaf river watershed

The performance metrics utilized in this study require a reference Pareto set or the best known approximation to the Pareto optimal set. The best known approximation set was gener-

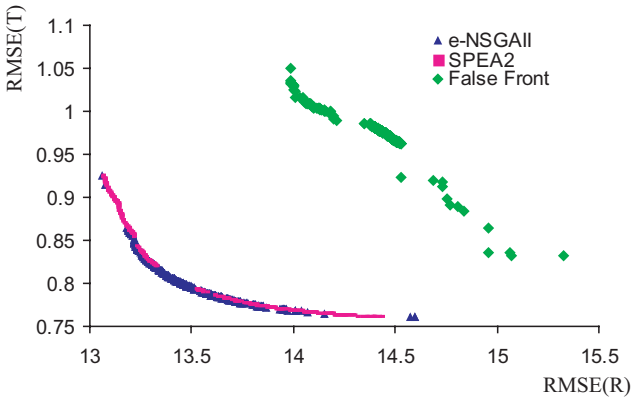


**Fig. 5.** **(a)** Reference set generated for the Leaf River test case where RMSE(T) are the errors for the Box-Cox transform of the hydrograph and RMSE(R) are the errors for the raw hydrograph. The figure also shows a false front that often trapped the algorithms. **(b)** The percentage of the reference set contributed by $\varepsilon$-NSGAII, SPEA2, and MOSCEM-UA.

**Table 5.** Leaf River case study's ratios of top trial runs for each configuration of the algorithms based on the binary $\varepsilon$-indicator metric ranking. The best performing algorithm is highlighted in bold.

| MOEA | Configurations | Top Ranking Ratios |
|---|---|---|
| $\varepsilon$-NSGAII | ($N$=10) | 23/50 |
| SPEA2 | ($N$=100) | **42/50** |
| MOSCEM-UA | ($N$=500, $C$=10) | 13/50 |

ated by collecting all of the nondominated solutions generated from the 150 trial runs used for this case study (i.e., 50 trial runs per algorithm). Figure 5 shows the solutions contributed by each algorithm for the 2-objective tradeoff between the Box-Cox transformed RMSE metric and the standard RMSE metric.

$\varepsilon$-NSGAII found 58% of the reference set and the remaining 42% of the reference set was generated by SPEA2. MOSCEM-UA was unable to contribute to the best solutions that compose the reference set. Table 5 shows that SPEA2 was able to attain the best binary $\varepsilon$-indicator metric rankings followed by $\varepsilon$-NSGAII and lastly MOSCEM-UA.

Table 6 shows that SPEA2 had the best average performance in terms of both the $\varepsilon$-indicator and hypervolume unary metrics. The Mann-Whitney test validated that SPEA2's results were different from both MOSCEM-UA and $\varepsilon$-NSGAII at the 99% confidence level.

The results of Table 6 demonstrate that average performance metrics can be misleading without statistical testing. Although MOSCEM-UA has superior mean hypervolume and $\varepsilon$-indicator distance values relative to $\varepsilon$-NSGAII, performance differences between the algorithms were not statistically significant (i.e., the null hypothesis in the Mann-Whitney test could not be rejected). In fact, all three algorithms had significant ranges of performance for this test case because of the presence of a large false front (i.e., the locally nondominated front shown in Fig. 5) that caused some of the algorithms' runs to miss the best known front. Figure 6 illustrates the variability in the algorithms' performances by presenting runtime results for the $\varepsilon$-indicator distance metric.

Figure 6 verifies that SPEA2 has the best mean performance over the full duration of the run. The figure also shows that SPEA2 was slightly more reliable relative to $\varepsilon$-NSGAII and MOSCEM-UA. Dynamic plots for hypervolume showed similar runtime distributions for the three algorithms. Figure 7 illustrates dynamic results for the best trial runs for each of the algorithms. The best trial runs were selected based on the algorithms' best unary metrics scores.

The plot shows that $\varepsilon$-NSGAII is able to attain superior hypervolume (diversity) and $\varepsilon$-indicator distance (convergence) metrics in less than 5000 model evaluations. SPEA2 and MOSCEM-UA required between 12 000 and 25 000 model evaluations to attain equivalent performance metric values. Overall SPEA2 had superior performance for this test case while MOSCEM-UA and $\varepsilon$-NSGAII had comparable performances.

### 5.3 Optimization results for case study 3: Shale Hills watershed

For the Shale Hills test case, MOSCEM-UA's parameters were challenging to set given the computational expense of the integrated hydrologic model. As discussed in Sect. 3.3.1, the Shale Hills test case poses a tremendous computational challenge where a single algorithm trial run requires approximately a week of computation. Given the magnitude of simulation evaluation times, the computational time spent in algorithmic search for both $\varepsilon$-NSGAII and SPEA2 is negligible. Unfortunately, MOSCEM-UA's algorithmic time is not negligible for increasing population sizes and increasing numbers of complexes because the algorithm utilizes a matrix inversion as part of its stochastic search operators. The severity of MOSCEM-UA's algorithmic inefficiency is highlighted in the test function analysis where $\varepsilon$-NSGAII was able to solve the test function suite for 50 random seeds in times on the order of minutes, MOSCEM-UA required days for population sizes greater than 250. For the Shale Hills case study, MOSCEM-UA was tested for a population size of 250 with 2 or 5 complexes because increasing these parameters caused a single run to exceed the 7 day maximum run times allotted for the LION-XO computing cluster. The severe computational demands of this test case required that we assess the algorithms using 15 random seed trials. If the 60 trial runs (i.e., 4 algorithm configurations * 15 random seed trials) were run on a single Pentium 4 Linux workstation with a 3 gigahertz processor and 2 gigabytes of RAM this test case would have required approximately 420 days of continuous computation.

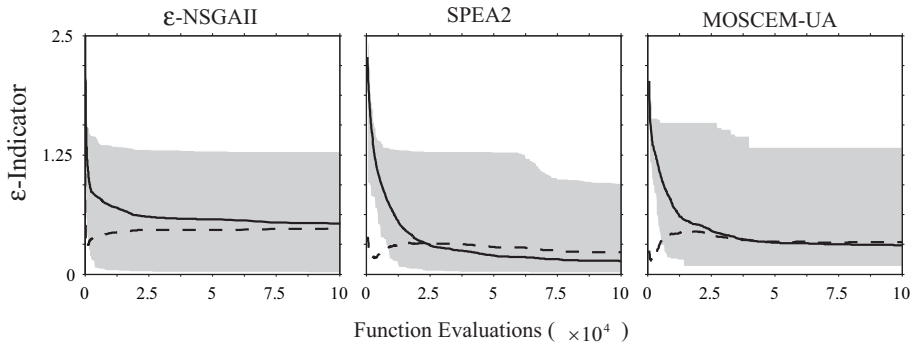


**Fig. 6.** Leaf River test case dynamic performance results for the unary $\varepsilon$-indicator distance metric versus total design evaluations. Mean performance is indicated by a solid line, the standard deviation by a dashed line, and the range of performance by the shaded region. The plots were generated using 50 trial runs for each algorithm.

**Table 6.** Leaf River case study's results for the averages and standard deviations of the unary metrics for each algorithm configuration. AVG stands for mean, STD stands for standard deviation, and bolded entries highlight the best value attained.

| MOEA | Hypervolume | | $\varepsilon$-Indicator | | Time (s) | |
|---|---|---|---|---|---|---|
| | AVG | STD | AVG | STD | AVG | STD |
| $\varepsilon$-NSGAII | 1.11E+0 | 1.04E+0 | 5.31E-1 | 4.78E-1 | **8.29E+2** | 3.58E+1 |
| SPEA2 | **2.96E-1** | 4.32E-1 | **1.39E-1** | 2.30E-1 | 8.33E+2 | 1.82E+1 |
| MOSCEM-UA | 5.49E-1 | 6.49E-1 | 3.05E-1 | 3.34E-1 | 1.24E+3 | 5.95E+1 |

**Table 7.** Shale Hills case study's ratios of top trial runs for each configuration of the algorithms based on the binary $\varepsilon$-indicator metric ranking. The best performing algorithm is highlighted in bold.

| MOEA | Configurations | Top Ranking Ratios |
|---|---|---|
| $\varepsilon$-NSGAII | ($N$=10) | 14/15 |
| SPEA2 | ($N$=100) | **15/15** |
| MOSCEM-UA | ($N$=250, $C$=2) | 4/15 |
| | ($N$=250, $C$=5) | 6/15 |

**Table 8.** Shale Hills case study's results for the averages and standard deviations of the unary metrics for each algorithm configuration. AVG stands for mean, STD stands for standard deviation, and bolded entries highlight the best value attained.

| MOEA | Hypervolume | | $\varepsilon$-Indicator | |
|---|---|---|---|---|
| | AVG | STD | AVG | STD |
| $\varepsilon$-NSGAII | 2.09E+04 | 1.82E+04 | 1.18E+0 | 1.95E-1 |
| SPEA2 | **1.63E+04** | 7.17E+03 | **1.12E+0** | 4.46E-2 |
| MOSCEM-UA | 4.71E+04 | 1.93E+04 | 1.38E+0 | 2.22E-1 |

The best known approximation set was generated by collecting the nondominated solutions from the 60 trial runs used for this case study. Figure 8a shows the best known solution set in the 3-objective solution space defined for this test case. Figure 8b projects the solution set onto the 2-objective planes to better illustrate the tradeoffs that exist between low, average, and peak flow calibration errors.

Figure 9 shows that $\varepsilon$-NSGAII found 94% of the reference set and the remaining 6% of the reference set was generated by SPEA2. MOSCEM did not contribute to the best solutions that compose the reference set.

Table 7 shows that SPEA2 was able to attain slightly better binary $\varepsilon$-indicator metric rankings relative to the $\varepsilon$-NSGAII. As indicated by Fig. 9 and Table 7 MOSCEM had difficultly in generating highly ranked runs for this test case. Although Table 8 shows that SPEA2 had the best average performance in terms of the $\varepsilon$-indicator and hypervolume unary metrics, the Mann-Whitney test showed that SPEA2's results were not statistically different from $\varepsilon$-NSGAII. Relative to MOSCEM-UA, SPEA2 and $\varepsilon$-NSGAII attained superior results that were confirmed to be statistically different at the 99% confidence level.

Figures 10 and 11 show the dynamic results for the full distribution of trials and for the best single runs for the three algorithms, respectively. The best trial runs were selected

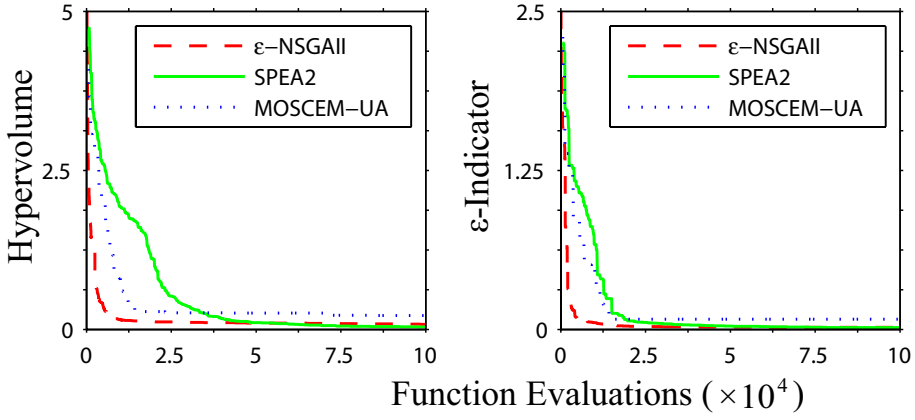


**Fig. 7.** Dynamic performance plots showing the best performing Leaf River trial runs for each algorithm.

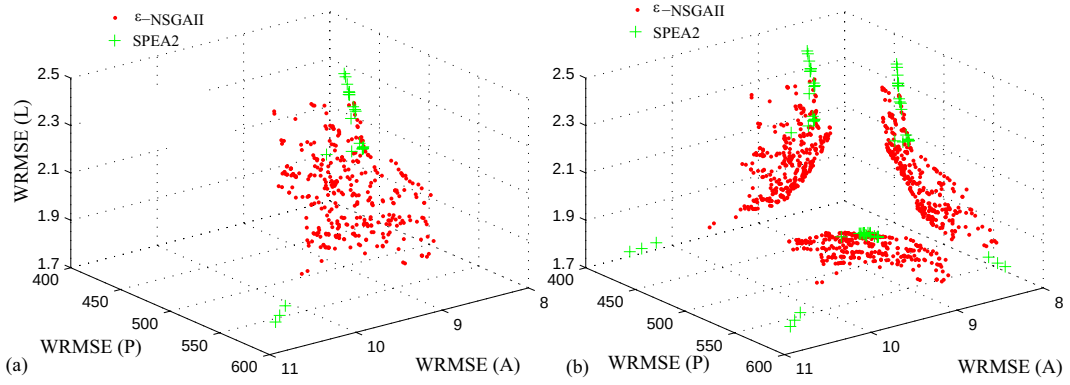


**Fig. 8. (a)** Reference set for the Shale Hills test case **(b)** projections of the reference set onto the 2-objective planes to highlight the tradeoffs between the objectives.

based on the algorithms' best unary metrics scores. Performance metric differences between SPEA2 and $\varepsilon$-NSGAII resulted from a single trial run. As shown in Table 7 a single $\varepsilon$-NSGAII run failed to attain a top binary ranking, which is reflected in the upper bound of the shaded region in Fig. 10. This single run highly biased both the mean and standard deviations for the unary metrics given in Table 8 for $\varepsilon$-NSGAII. The Mann-Whitney test validates that the remaining $\varepsilon$-NSGAII trial runs were not statistically different from SPEA2. For MOSCEM-UA, Table 7 in combination with Fig. 10 show that more than 60 percent of the algorithm's trial runs failed to solve this test case. Figures 9 and 11 show that $\varepsilon$-NSGAII's best runs were superior relative to the other algorithms' results, generating nearly all of the reference set.

As was noted for the Leaf River case study, SPEA2's performance for the Shale Hills test case is heavily impacted by its archive size. It has been widely recognized (Coello Coello et al., 2002) that SPEA2's *k*-means clustering diversity operator allows the algorithm to attain highly diverse solution sets for high-order Pareto optimization problems (i.e., problems with 3 or more objectives). This operator is only active in the search process if the archive is sized appropriately, which in typical applications will require trial-and-error analysis. For this test case every trial run would require a week of computing time. It should be noted that $\varepsilon$-NSGAII automatically generates its archive size based on users' precision goals for each objective. Additionally, the algorithm starts with a very small population size, which is automatically adjusted to enhance search effectiveness. The results presented in this study are conservative tests for the $\varepsilon$-NSGAII because SPEA2 and MOSCEM-UA initiate search with at least an order of magnitude advantage in search population.

## 6 Discussion

### 6.1 Relative benefits and limitations of SPEA2

SPEA2 is an excellent benchmark algorithm for multiobjective hydrologic model calibration. Overall SPEA2 attained competitive to superior results for most of the problems tested in this study. The algorithm's poorest performance occurred on the T4 test function, which represents a severely

difficult multimodal problem with $21^9$ local fronts. SPEA2's best overall performance occurred for the Leaf River case study where the algorithm was far more reliable relative to both the $\varepsilon$-NSGAII and MOSCEM-UA. The Leaf River test case is challenging because of its multimodality (see Fig. 5). Our analysis showed that carefully setting the archive size for SPEA2 for this case study enabled the algorithm to fully exploit its k-means clustering diversity operator to spread solutions across the search space and more reliably escape the false nondominated front shown in Fig. 5. For the Shale Hills test case, SPEA2 and $\varepsilon$-NSGAII had statistically equivalent performance metrics, although SPEA2 was slightly more reliable. SPEA2 is generally superior in performance relative to MOSCEM-UA.

The primary strengths of the SPEA2 algorithm lie in the algorithm's search reliability and its diversity preservation operator as has been recognized in other studies. In this study, SPEA2 showed a limited sensitivity to its population sizing and search parameters. Other studies (Zitzler et al., 2001; Coello Coello et al., 2002; Deb et al., 2003) have shown that SPEA2's sensitivity to population size often manifests itself in terms of a performance threshold for very difficult problems where the algorithm fails until the population is made sufficiently large. In this study, SPEA2's poor performance on test function T4 provides an example of this performance threshold. In these cases, it is very difficult to predict how to appropriately size SPEA2's population. Significant trial-and-error analysis is required. The biggest challenge in maximizing the performance of SPEA2 lies in specifying an effective archive size without a priori knowledge of the Pareto set. In practice, this would require significant trial-and-error analysis, which is problematic for more complex, computationally intensive calibration applications.

### 6.2 Relative benefits and limitations of MOSCEM-UA

MOSCEM-UA was the least competitive of the three algorithms tested in this study failing to effectively solve either the standardized test function suite or the Shale Hills test case. MOSCEM-UA attained its best performance on the Leaf River case study, which was used in its development (Vrugt et al., 2003a). On the Leaf River case study, MOSCEM-UA was inferior to SPEA2 and statistically similar to $\varepsilon$-NSGAII. MOSCEM-UA did not contribute to any of the reference sets (i.e., the best overall solutions) for the two hydrologic calibration applications. The algorithm's Markov Chain Monte Carlo sampler in combination with its shuffle complex search operator does not scale well for problems of increasing size and/or difficulty. MOSCEM-UA's binary $\varepsilon$-indicator rankings for all three test cases show that the algorithm is not reliable even with significant increases in population size and the number of complexes.

MOSCEM-UA's primary strength is its estimation of the posterior parameter distributions for hydrologic model parameters (assuming the initial Gaussian assumptions made for hydrologic parameters are acceptable to users). Additionally, the algorithm has a limited number of parameters that need to be specified (i.e., the population size, run length, and number of complexes). MOSCEM-UA is however, critically sensitive to these parameters. The matrix inversion used in the algorithm's stochastic search operators causes MOSCEM-UA's efficiency to dramatically reduce with increases in population size and increases in the number of complexes. The algorithm is best suited for hydrologic model calibration applications that have small parameter sets and small model evaluation times. In general, it would be expected that MOSCEM-UA's performance would be met or exceeded by either SPEA2 or $\varepsilon$-NSGAII.

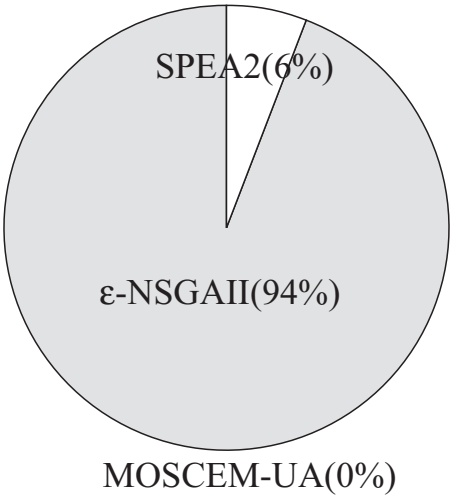


**Fig. 9.** The percentages of the Shale Hills reference set contributed by $\varepsilon$-NSGAII, SPEA2, and MOSCEM-UA.

### 6.3 Relative benefits and limitations of $\varepsilon$-NSGAII

$\varepsilon$-NSGAII attained competitive to superior performance results relative to SPEA2 on the test function suite and the Shale Hills test case. Overall, $\varepsilon$-NSGAII generated the majority the reference sets (i.e., best overall solutions) for both hydrologic model calibration case studies. $\varepsilon$-NSGAII also had the best single run results for both of the calibration case studies as illustrated in Figs. 7 and 11. The algorithm's poorest performance occurred on Leaf River case study, in which its average performance was inferior to SPEA2 and statistically equivalent to MOSCEM-UA.

Although $\varepsilon$-NSGAII generated 58% of the reference set for the Leaf River test, its binary $\varepsilon$-indicator metric rankings (see Table 5) show that the algorithm performed less reliably than SPEA2. This highlights the biggest limitation impacting $\varepsilon$-NSGAII's performance, which is related to its parent algorithm NSGAII's diversity operator (Deb et al., 2002). It has been widely reported (Coello Coello et al., 2002; Deb et al., 2003) that the original NSGAII converges very quickly, but

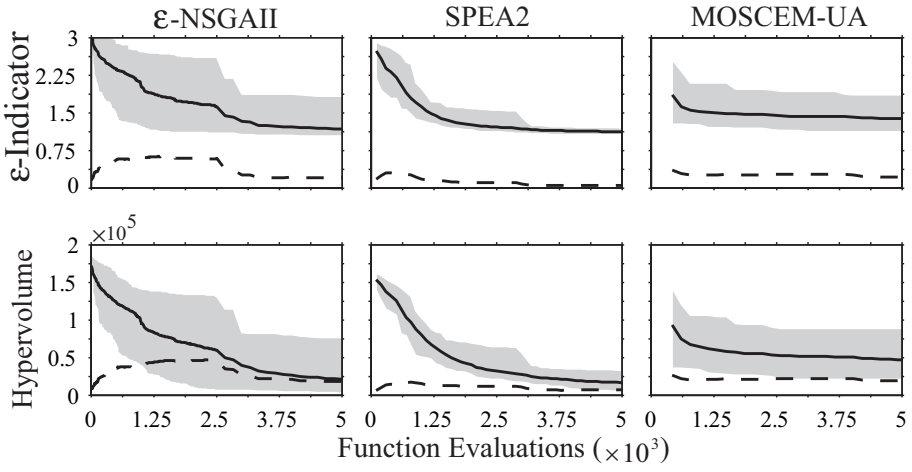


**Fig. 10.** Shale Hills test case dynamic performance results for the unary $\varepsilon$-indicator distance metric versus total design evaluations. Mean performance is indicated by a solid line, the standard deviation by a dashed line, and the range of performance by the shaded region. The plots were generated using 15 trial runs for each algorithm.

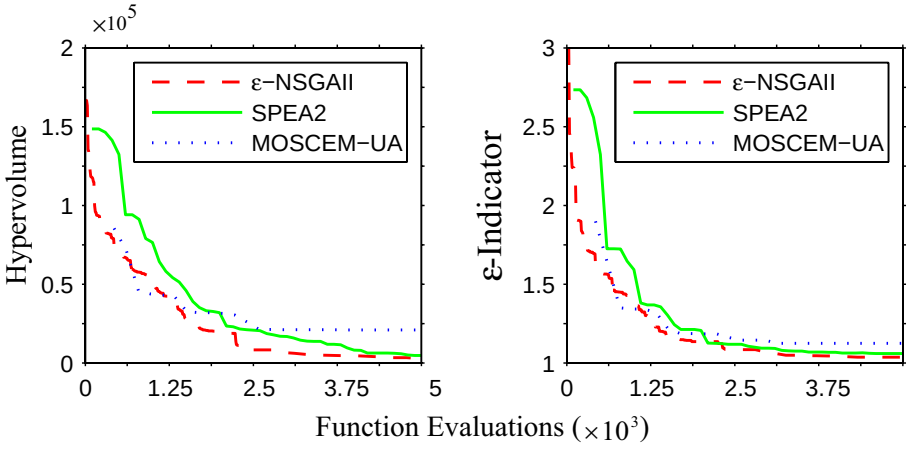


**Fig. 11.** Dynamic performance plots showing the best performing Shale Hills trial runs for each algorithm.

its crowded tournament diversity operator can fail to promote sufficient diversity for some problems. Although Kollat and Reed (2005a, b) have demonstrated $\varepsilon$-NSGAII is statistically superior to the original NSGAII in terms of both convergence and diversity, $\varepsilon$-NSGAII can still be impacted by the limitations associated with the crowded tournament operator. For the Leaf River case study, $\varepsilon$-NSGAII had a reduced reliability relative to SPEA2 because several trial runs failed to create sufficiently diverse solutions that could escape the false local front. As was discussed above, SPEA2's archive was sized carefully to maximize the effectiveness of its k-means clustering diversity operator, which allowed the algorithm to escape the local front. It is interesting to note that for the multimodal T4 test function with $21^9$ local fronts, that $\varepsilon$-NSGAII's performance is far superior to SPEA2. In this instance, $\varepsilon$-NSGAII's was able to escape local fronts because of the random solutions injected into the search population during the algorithm's dynamic changes in population size. In the limit, when the algorithm's $\varepsilon$-dominance archive size stabilizes, the $\varepsilon$-NSGAII's dynamic population sizing and random solution injection is equivalent to a diversity enhancing search operator termed "time continuation" (Goldberg, 2002).

In this study, $\varepsilon$-NSGAII appears to be superior to MOSCEM-UA and competitive with SPEA2 for hydrologic model calibration. $\varepsilon$-NSGAII's primary strength lies in its ease-of-use due to its dynamic population sizing and archiving which lead to rapid convergence to very high quality solutions. Overall $\varepsilon$-NSGAII found a majority of the best known solutions for the calibration problems using less than 5000 model evaluations. $\varepsilon$-NSGAII's dynamic population sizing and archive-based preconditioning of search helps eliminate the need for trial-and-error analysis relative to SPEA2, which is particularly important for computationally intensive applications like the Shale Hills test case.

## 7 Conclusions

This study provides a comprehensive assessment of state-of-the-art evolutionary multiobjective optimization tools' relative effectiveness in calibrating hydrologic models. Three test cases were used to compare the algorithms' performances. The first test case is composed of a standardized suite of computer science test problems, which are used to validate the algorithms' abilities to perform global search effectively, efficiently, and reliably for a broad range of problem types. The $\varepsilon$-NSGAII attained the best overall performance for the test function suite followed by SPEA2. MOSCEM-UA was not able to solve the test function suite reliably. The second test case is a benchmark hydrologic calibration problem in which the Sacramento soil moisture accounting model is calibrated for the Leaf River watershed. SPEA2 attained statistically superior performance for this case study in all metrics at the 99% confidence level. MOSCEM-UA and $\varepsilon$-NSGAII attained results that were competitive with one another for the Leaf River case study. The third test case assesses the algorithms' performances for a computationally intensive integrated hydrologic model calibration application for the Shale Hills watershed located in the Susquehanna River Basin in north central Pennsylvania. For the Shale Hills test case, SPEA2 and $\varepsilon$-NSGAII had statistically equivalent performance metrics, although SPEA2 was slightly more reliable. MOSCEM-UA's performance on the Shale Hills test case was limited by the severe computational costs associated with increasing the algorithm's population size and number of complexes.

Overall, SPEA2 is an excellent benchmark algorithm for multiobjective hydrologic model calibration. SPEA2 attained competitive to superior results for most of the problems tested in this study. The primary strengths of the SPEA2 algorithm lie in its search reliability and its diversity preservation operator. The biggest challenge in maximizing the performance of SPEA2 lies in specifying an effective archive size without a priori knowledge of the Pareto set. In practice, this would require significant trial-and-error analysis, which is problematic for more complex, computationally intensive calibration applications. $\varepsilon$-NSGAII appears to be superior to MOSCEM-UA and competitive with SPEA2 for hydrologic model calibration. $\varepsilon$-NSGAII's primary strength lies in its ease-of-use due to its dynamic population sizing and archiving which lead to rapid convergence to very high quality solutions with minimal user input. MOSCEM-UA is best suited for hydrologic model calibration applications that have small parameter sets and small model evaluation times. In general, it would be expected that MOSCEM-UA's performance would be met or exceeded by either SPEA2 or $\varepsilon$-NSGAII. Future hydrologic calibration studies are needed to test emerging algorithmic innovations combining global multiobjective methods and local search (e.g., see Solomatine, 1998; Solomatine, 1999; Ishibuchi and Narukawa, 2004; Krasnogor and Smith, 2005; Solomatine, 2005).

*Acknowledgements.* This work was partially supported by the National Science Foundation under grants EAR-0310122 and EAR-0418798. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the writers and do not necessarily reflect the views of the National Science Foundation. Additional support was provided by the Pennsylvania Water Resources Center under grant USDI-01HQGR0099. Partial support for the third author was provided by SAHRA under NSF-STC grant EAR-9876800, and the National Weather Service Office of Hydrology under grant numbers NOAA/NA04NWS4620012, UCAR/NOAA/COMET/S0344674, NOAA/DG 133W-03-SE-0916.

Edited by: D. Solomatine

## References

Ajami, N. K., Gupta, H. V., Wagener, T., and Sorooshian, S.: Calibration of a semi-distributed hydrologic model for streamflow estimation along a river system, J. Hydrol., 298, 112–135, 2004.

Boyle, D. P., Gupta, H. V., and Sorooshian, S.: Toward improved calibration of hydrologic models: Combining the strengths of manual and automatic methods, Water Resour. Res., 36, 3663–3674, 2000.

Burnash, R. J. C.: The NWS river forecast system-Catchment model, in: Computer Models of Watershed Hydrology, edited by: Singh, V. P., Water Resources Publications, Highlands Ranch, CO, 1995.

Carsel, R. F.: Developing joint probability distributions of soil water retention characteristics, Water Resour. Res., 24, 755–769, 1988.

Cieniawski, S. E., Eheart, J. W., and Ranjithan, S. R.: Using genetic algorithms to solve a multiobjective groundwater monitoring problem, Water Resour. Res., 31, 399–409, 1995.

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

Conover, W. J.: Practical Nonparametric Statistics , 3rd edition, John Wiley & Sons, New York, 1999.

Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons LTD., New York, 2001.

Deb, K. and Jain, S.: Running Performance Metrics for Evolutionary Multi-Objective Optimization, 2002004, Indian Institute of Technology Kanpur, Kanpur, India, 2002.

Deb, K., Mohan, M., and Mishra, S.: A Fast Multi-objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions, KanGAL Report No. 2003002, Indian Institute of Technology, Kanpur, India, 2003.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE Trans. Evol. Computation, 6, 182-197, 2002.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E.: Scalable multi-objective optimization test problems, in: In Proceedings of the Congress on Evolutionary Computation (CEC-2002), pp. 825–830, 2002.

Dingman, S. L.: Physical Hydrology, Second Edition edition, Prentice Hall, New Jersey, 2002.

Duan, Q., Gupta, V. K., and Sorooshian, S.: Effective and efficient global optimization for conceptual rainfall-runoff models, Water Resour. Res., 28, 1015–1031, 1992.

Duan, Q., Gupta, V. K., and Sorooshian, S.: A shuffled complex evolution approach for effective and efficient global minimization, J. Optimiz. Theory App., 76, 501–521, 1993.

Duffy, C. J.: A two-state integral-balance model for soil moisture and groundwater dynamics in complex terrain, Water Resour. Res., 32, 2421–2434, 1996.

Duffy, C. J.: A Distributed-Dynamical Model for Mountain-Front Recharge & Water Balance Estimation: The Rio Grande of Southern Colorado and New Mexico, in: AGU Monograph: Recharge in Semi-Arid Regions: State of the Art, edited by: Scanlon, H. a., 2004.

Erickson, M. A., Mayer, A., and Horn, J.: Multi-objective optimal design of groundwater remediation systems: application of the niched Pareto genetic algorithm (NPGA), Adv. Water Resour., 25, 51–56, 2002.

Fonseca, C. M., Knowles, J. D., Thiele, L., and Ziztler, E.: A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers, in: Evolutionary Multi Criterion Optimization: Third International Conference (EMO 2005), Guanajuato, Mexico, 2005.

Gan, T. Y. and Biftu, G. F.: Automatic calibration of conceptual rainfall-runoff models: optimization algorithms, catchment conditions, and model structure, Water Resour. Res., 32, 3513–3524, 1996.

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

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

Guglielmi, N. and Hairer, E.: Implementing Radau IIA methods for stiff delay differential equations, Computing, 67, 1–12, 2001.

Gupta, H. V., Sorooshian, S., and Yapo, P. O.: Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information, Water Resour. Res., 34, 751–763, 1998.

Hairer, E. and Wanner, G.: Solving Ordinary Differential Equations II – Stiff and Differential-Algebraic Problems, Second Revised Edition, Springer, Verlag, 1996.

Halhal, D., Walters, G. A., Ouazar, D., and Savic, D. A.: Water network rehabilitation with structured messy genetic algorithm, J. Water Res. Pl., 123, 137–146, 1997.

Horn, J. and Nafpliotis, F.: Multiobjective Optimization using the Niched Pareto Genetic Algorithm, IlliGAL Report No. 93005, University of Illinois, Urbana, IL, 1993.

Ishibuchi, H. and Narukawa, K.: Comparision of local search implementation schemes in hybrid evolutionary multiobjective optimization algorithms, in: Hybrid Intelligent Systems (HIS 2004): Fourth International Conference, IEEE Press, Kitakyushu, Japan, 404–409, 2004.

Kollat, J. B. and Reed, P.: The Value of Online Adaptive Search: A comparison of NSGA-II, $\varepsilon$-NSGAII, and $\varepsilon$ MOEA, in: Evolutionary Multi Criterion Optimization: Third International Conference (EMO 2005), edited by: Coello Coello, C., Hernandez, A., and Zitzler, E., Lecture Notes in Computer Science, Springer-Verlag, Guanajuato, Mexico, 386–398, 2005a.

Kollat, J. B. and Reed, P.: Comparing State-of-the-Art Evolutionary Multi-Objective Algorithms for Long-Term Groundwater Monitoring Design, Adv. Water Resour., 29, 792–807, 2005b.

Krasnogor, N. and Smith, J.: A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues, IEEE Trans. Evol. Comput., 9, 474–488, 2005.

Kuczera, G.: Efficient subspace probabilistic parameter optimization for catchment models, Water Resour. Res., 33, 177–185, 1997.

Laumanns, M., Thiele, L., Deb, K., and Zitzler, E.: Combining Convergence and Diversity in Evolutionary Multiobjective Optimization, Evol. Comput., 10, 263–282, 2002.

Lin, H. S., Kogelmann, W., Walker, C., and Bruns, M. A.: Soil moisture patterns in a forested catchment: A hydropedological perspective, Geoderma, 131, 345–368, 2005.

Loughlin, D. H., Ranjithan, S. R., Baugh Jr., J. W., and Brill Jr., E. D.: Application of Genetic Algorithms for the Design of Ozone Control Strategies, J. Air Waste Manage. Assoc., 50, 1050–1063, 2000.

Madsen, H.: Automatic calibration of a conceptual rainfall-ruoff model using multiple objectives, J. Hydrol., 235, 276–288, 2000.

Madsen, H.: Parameter estimation in distributed hydrological catchment modeling using automatic calibration with multiple objectives, Adv. Water Resour., 26, 205–216, 2003.

Madsen, H., Wilson, G., and Ammentorp, H. C.: Comparison of different automated strategies for calibration of rainfall-runoff models, J. Hydrol., 261, 48–59, 2002.

Misirli, F., Gupta, H. V., Sorooshian, S., and Thiemann, M.: Bayesian recursive estimation of parameter and output uncertainty for watershed models, in: Calibration of Watershed Models, edited by: Duan, Q., Gupta, H. V., Sorooshian, S., et al., American Geophysical Union, Washington, D.C., pp. 113–124, 2003.

Muleta, M. and Nicklow, J.: Decision Support for Watershed Management Using Evolutionary Algorithms, J. Water Res. Pl., 131, 35–44, 2005a.

Muleta, M. K. and Nicklow, J. W.: Sensitivity and uncertainty analysis coupled with automatic calibration for a distributed watershed model, J. Hydrol., 306, 127–145, 2005b.

Pareto, V.: Cours D'Economie Politique, Rouge, Lausanne, 1896.

Qu, Y.: An integrated hydrologic model for multi-process simulation using semi-discrete finite volume approach, PhD dissertation, Penn State University, University Park, PA, 2004.

Reed, P. and Minsker, B. S.: Striking the Balance: Long-Term Groundwater Monitoring Design for Conflicting Objectives, J. Water Res. Pl., 130, 140–149, 2004.

Reed, P., Minsker, B. S., and Goldberg, D. E.: A multiobjective approach to cost effective long-term groundwater monitoring using an Elitist Nondominated Sorted Genetic Algorithm with historical data, J. Hydroinform., 3, 71–90, 2001.

Reed, P., Minsker, B. S., and Goldberg, D. E.: Simplifying Multiobjective Optimization: An Automated Design Methodology for the Nondominated Sorted Genetic Algorithm-II, Water Resour. Res., 39, 1196, doi:1110.1029/2002WR001483, 2003.

Ritzel, B. J., Eheart, J. W., and Ranjithan, S. R.: Using genetic algorithms to solve a multiple objective groundwater pollution containment problem, Water Resour. Res., 30, 1589–1603, 1994.

Schaffer, J. D.: Some experiments in machine learning using vector evaluated genetic algorithms, Doctoral Thesis, Vanderbilt University, Nashville, TN, 1984.

Schwefel, H.-P.: Evolution and Optimum Seeking, Computer Disk Edition edition, John Wiley & Sons, Inc., New York, 1995.

Seibert, J.: Multi-criteria calibration of a conceptual runoff model using a genetic algorithm, Hydrol. Earth Syst. Sci., 4, 215–224, 2000.

Solomatine, D.P.: Genetic and other global optimization algorithms – comparison and use in calibration problems, in: Proc. 3rd International Conference on Hydroinformatics, Copenhagen, Denmark, 1021–1028, 1998.

Solomatine, D.P.: Two strategies of adaptive cluster covering with descent and their comparison to other algorithms, J. Global Optim., 14(1), 55–78, 1999.

Solomatine, D.P.: Adaptive cluster covering and evolutionary approach: comparison, differences and similarities, in: Proc. IEEE Congress on Evolutionary Computation (CEC-2005), Edinburgh, U.K., 1959–1966, 2005.

Sorooshian, S. and Dracup, J. A.: Stochastic parameter estimation procedures for hydrologic rainfall-runoff models: correlated and heteroscedastic error cases, Water Resour. Res., 16, 430–442, 1980.

Sorooshian, S., Duan, Q., and Gupta, H. V.: Calibration of rainfall-runoff models: Application of global optimization to the Sacramento soil moisture accounting model, Water Resour. Res., 29, 1185–1194, 1993.

Vrugt, J., Gupta, H. V., Bastidas, L. A., Bouten, W., and Sorooshian, S.: Effective and efficient algorithm for multiobjective optimization of hydrologic models, Water Resour. Res., 39, 1214, doi:1210.1029/2002WR001746, 2003a.

Vrugt, J., Gupta, H. V., Bouten, W., and Sorooshian, S.: A Shuffled Complex Evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters, Water Resour. Res., 39, 1201, doi:1210.1029/2002WR001642, 2003b.

Vrugt, J. A., Schoups, G., Hopmans, J. W., Young, C., Wallender, W. W., and Harter, T.: Inverse modeling of large-scale spatially distributed vadose zone properties using global optimization, Water Resour. Res., 41, W06003, doi:10.1029/2004WR003698, 2005.

Wagener, T., Boyle, D. P., Lees, M. J., Wheater, H. S., Gupta, H. V., and Sorooshian, S.: A framework for development and application of hydrological models, Hydrol. Earth Syst. Sci., 5, 13–26, 2001.

Wagener, T. and Gupta, H. V.: Model Identification for hydrological forecasting under uncertainty, Stoch. Env. Res. Risk A., 19 (6), 378–387, doi:10.1007/s00477-005-006-5, 2005.

Yapo, P. O., Gupta, H. V., and Sorooshian, S.: Automatic calibration of conceptual rainfall-runoff models: sensitivity to calibration data, J. Hydrol., 181, 23–48, 1996.

Yapo, P. O., Gupta, H. V., and Sorooshian, S.: Multi-objective global optimization for hydrologic models, J. Hydrol., 204, 83–97, 1998.

Zitzler, E., Deb, K., and Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results, Evol. Comput., 8, 125–148, 2000.

Zitzler, E., Laumanns, M., and Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm, TIK-103, Department of Electrical Engineering, Swiss Federal Institute of Technology, Zurich, Switzerland, 2001.

Zitzler, E. and Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, IEEE Trans. Evol. Computation, 3, 257–271, 1999.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Grunert da Fonseca, V.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review, IEEE Trans. Evol. Computation, 7, 117–132, 2003.