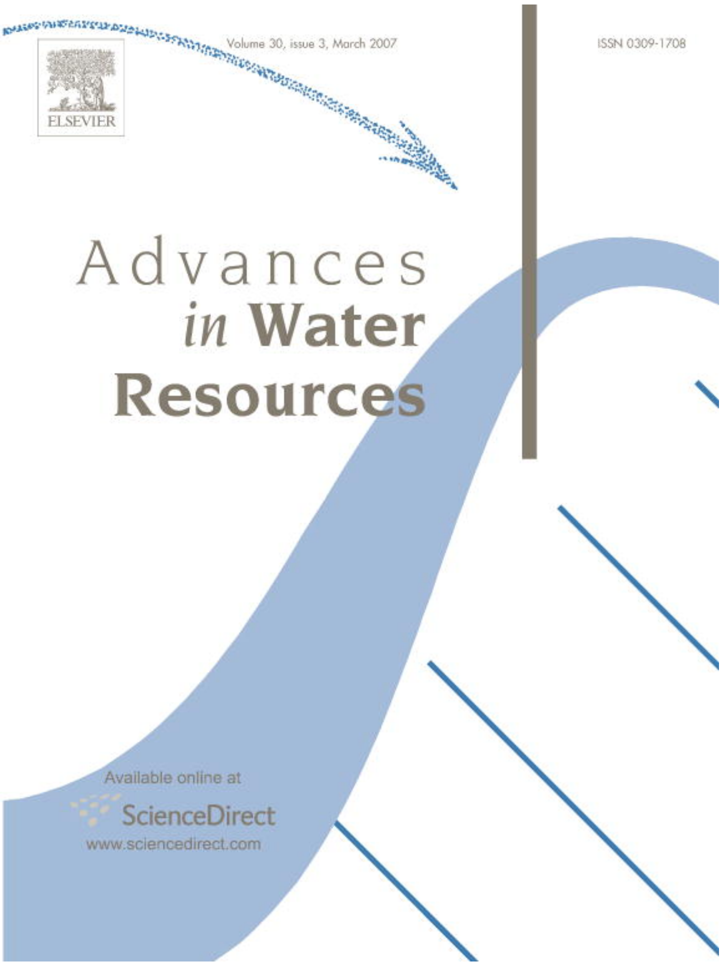
Volume 30, issue 3, March 2007
ISSN 0309-1708
ELSEVIER
Advances in Water Resources
Available online at
ScienceDirect
www.sciencedirect.com

Available online at www.sciencedirect.com

ScienceDirect

Advances in Water Resources 30 (2007) 335–353

Advances in Water Resources

www.elsevier.com/locate/advwatres

Parallelization strategies for rapid and robust evolutionary multiobjective optimization in water resources applications

Y. Tang, P.M. Reed [*], J.B. Kollat

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

Received 29 March 2006; received in revised form 5 June 2006; accepted 5 June 2006
Available online 17 August 2006

Abstract

This study uses a formal metrics-based framework to demonstrate the Master–Slave (MS) and the Multiple-Population (MP) parallelization schemes for the Epsilon-Nondominated Sorted Genetic Algorithm-II (ε-NSGAII). The MS and MP versions of the ε-NSGAII generalize the algorithm's auto-adaptive population sizing, ε-dominance archiving, and time continuation to a distributed processor environment using the Message Passing Interface. This study uses three test cases to compare the MS and MP versions of the ε-NSGAII: (1) an extremely difficult benchmark test function termed DTLZ6 drawn from the computer science literature, (2) an unconstrained, continuous hydrologic model calibration test case for the Leaf River near Collins, Mississippi, and (3) a discrete, constrained four-objective long-term groundwater monitoring (LTM) application. The MP version of the ε-NSGAII is more effective than the MS scheme when solving DTLZ6. Both the Leaf River and the LTM test cases proved to be more appropriately suited to the MS version of the ε-NSGAII. Overall, the MS version of the ε-NSGAII exhibits superior performance on both of the water resources applications, especially when considering its simplicity and ease-of-implementation relative to the MP scheme. A key conclusion of this study is that a simple MS parallelization strategy can exploit time-continuation and parallel speedups to dramatically improve the efficiency and reliability of evolutionary multiobjective algorithms in water resources applications.

Keywords: Evolutionary algorithms; Multiobjective optimization; Parallelization; Hydrologic calibration; Groundwater monitoring

1. Introduction

This study uses a formal metrics-based framework to demonstrate the Master–Slave (MS) and the Multiple-Population (MP) parallelization schemes for the Epsilon-Nondominated Sorted Genetic Algorithm II (ε-NSGAII). The ε-NSGAII was developed by the authors and has been recently demonstrated to be competitive and superior to other state-of-the-art evolutionary multiobjective (EMO) algorithms on a suite of water resources applications (see [1–3]). The MS and MP versions of the ε-NSGAII generalize the algorithm's auto-adaptive population sizing, ε-dominance archiving, and time continuation to a distributed processor environment using the Message Passing Interface (MPI) parallelization libraries [4]. This paper assumes that readers will have an introductory understanding of EMO algorithms (for excellent introductions see [5,6]). A key finding of this work is that time-continuation and parallel speedups can dramatically improve the efficiency and reliability of EMO algorithms in water resources applications. Time continuation [7] is an evolutionary algorithm (EA) search enhancement that promotes solution diversity and allows the ε-NSGAII to maintain effective search for as long as is necessary or is computationally tractable (more details are presented in Section 2.2). This study uses three test cases to compare the MS and MP versions of the ε-NSGAII: (1) an extremely difficult benchmark test function termed DTLZ6 drawn from the computer science literature

[*] Corresponding author. Tel.: +1 814 8632940; fax: +1 814 8637304.
E-mail addresses: yxt132@psu.edu (Y. Tang), pmr11@psu.edu (P.M. Reed), juk124@psu.edu (J.B. Kollat).

0309-1708/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advwatres.2006.06.006

[8], (2) an unconstrained, continuous hydrologic model calibration test case for the Leaf River near Collins, Mississippi [3,9–14], and (3) a discrete, constrained four-objective long-term groundwater monitoring (LTM) application [2,15,16]. These test cases were carefully selected to encompass a broad range of problem properties and clearly demonstrate the importance of problem difficulty when selecting parallelization strategies for multiobjective applications. Specifically, this study highlights that artificially constructed, extremely difficult test problems such as DTLZ6 may wrongly bias water resources professionals towards using more complicated algorithms (i.e., the MP parallelization scheme), when a simpler MS strategy may work as well or better for real-world problems.

There has been a modern confluence of systems analysis research towards approaches that emphasize multiple objectives (see reviews [6,17–19]). This trend is clearly evident in the water resources literature over the past decade [20–28]. Recent applications demonstrate that a growing body of researchers in both the water resources and the broader systems analysis communities are seeking to use EMO algorithms to solve large (in terms of the number of decisions and objectives), computationally intensive optimization problems [5,6,28–32]. Kollat and Reed [16] have recently shown that EMO algorithms potentially have a quadratic computational complexity when solving water resources applications. A quadratic complexity implies that a k-fold increase in the number of decision variables will yield a k^2-fold increase in the number of function evaluations (NFE) required to solve an application. Kollat and Reed's [16] findings motivate the need to effectively design and comprehensively assess EMO parallelization schemes that can help to overcome the computational constraints posed by large water resources problems. There is a dearth of parallel EMO studies in the water resources literature despite the large number of areas where multiobjective applications are prevalent [14,20,21,25,33–37]. The Leaf River hydrologic calibration test case and the LTM test case were selected in this study to represent two distinctly different problem areas within the water resources literature where there is substantial interest in advancing multiobjective methods.

The Leaf River test case is a benchmark hydrologic model calibration problem in which the Sacramento soil moisture accounting model (SAC-SMA) is calibrated for the watershed located close to Collins, Mississippi. The Leaf River case study has been used in the development of both single and multiobjective calibration tools [9–14]. The growing body of research in the area of multiobjective calibration [11,12,14,38–42] 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 [38]. Although a majority of prior studies have focused on conceptual rainfall-runoff applications, there have been an increasing number of recent studies focused on developing multiobjective calibration strategies for computationally intensive distributed hydrologic models [33,41,43–45] where effective EMO parallelization strategies are needed. Readers should also note that our Leaf River analysis is conservative in the sense that the SAC-SMA evaluation times are relatively small, making it more difficult to attain efficient parallel speedups (see Sections 2.3.1 and 2.3.2).

Groundwater monitoring design was one of the first EMO application areas in the water resources literature [21]. In general, groundwater monitoring design has been shown to be a challenging optimization problem with several conflicting objectives and very large discrete decision spaces [25,27,46–53]. Knopman and Voss [47,54] recognized that the groundwater quality network design problem has many mathematical similarities to the classical combinatorial knapsack problem (i.e., discrete decision spaces that grow exponentially with increased problem size). Reed and Minsker [28] used the LTM problem to illustrate that EMO algorithms are capable of solving a new problem class [29,30], which they termed high-order Pareto optimization. The term high-order Pareto optimization is used to describe those applications that seek to quantify optimal tradeoffs between three or more objectives. This paper uses the LTM test case to demonstrate that the parallel versions of the ε-NSGAII can rapidly and reliably solve high-order Pareto optimization problems. More generally, the three test cases used in this study demonstrate that the MS and MP versions of the ε-NSGAII can broaden the size and difficulty of multiobjective water resources applications that can be solved efficiently and reliably.

This paper proceeds as follows. The multiobjective optimization methods, parallelization strategies, test cases, performance metrics, and computational experiment are discussed in Section 2. Section 3 present the results of the study followed by a discussion in Section 4 comparing the MS and MP versions of the ε-NSGAII. Finally, conclusions of the study are presented in Section 5.

2. Methodology

2.1. Evolutionary multi-objective optimization search

Evolutionary multiobjective optimization algorithms are similar to traditional single objective evolutionary algorithms in that all genetic algorithms search complex problem spaces using a process that is analogous to Darwinian natural selection. Evolutionary algorithms use a population-based search in which high quality solutions are evolved using the three basic operators of (1) selection, (2) mating, and (3) mutation. Analogous to natural systems, selection preferentially samples higher fitness solutions and biases the population to converge to the best solutions. The fitness of each solution is determined by how well it satisfies specified objectives and constraints of a given application. Mating occurs by combining the decision variables of high quality "parent" solutions to create "child" solutions. The mating operator in combination with selection allows evolutionary algorithms (EAs) to globally search promising regions of a problem space. Lastly, mutation perturbs the decision variables that compose population members. Selection in combination with mutation allows EAs to locally search the problem space near a given solution. The primary difference between EMO algorithms and single objective EAs lies in how fitness is assigned.

EMO algorithms assign solutions fitness values based on their performance across a vector of objectives. A solution cannot be assessed in terms of its performance in any single objective because it may perform poorly with respect to the remaining objectives. Instead, the concept of Pareto dominance is used to assign fitness values to solutions. A solution x dominates another solution x' if and only if it performs as well as x' in all objectives and better in at least one. Solutions are assigned ranks based on their Pareto dominance where top performing solutions are non-dominated (i.e., no solution exceeds their performance in all objectives). After Pareto ranking, EMO algorithms apply selection to bias search towards top ranking non-dominated solutions. In cases where solutions have equal ranking, selection biases search towards "diverse" solutions that are distant from neighboring solutions. Solution diversity is a key factor in finding solutions along the full extent of an application's tradeoffs.

More formally, the goal of multiobjective optimization is to identify the Pareto-optimal tradeoffs between an application's objectives. These tradeoffs are composed of the set of solutions that are better than all other solutions in at least one objective and are termed non-dominated or Pareto-optimal solutions [55]. The Pareto-optimal front is obtained by plotting these solutions according to their objective values yielding an $M-1$ dimensional surface where M is the total number of objectives. The term high-order Pareto surfaces is used to describe those surfaces that result from three or more objectives. EMO algorithms' population-based search enables them to evolve entire tradeoff (or Pareto) surfaces within a single run for problems with huge decision spaces. These methods can solve highly nonlinear, discrete, and non-convex problems without differentiation [56–58].

2.2. The ε-NSGAII

The ε-NSGAII has been demonstrated in both discrete and continuous water resources applications [2,3]. The ε-NSGAII's performance has been shown to be superior to the original Non-Dominated Sorted Genetic Algorithm II (NSGAII) [59] and the Epsilon-Dominance Multi-Objective Evolutionary Algorithm (εMOEA) [60] and competitive to superior relative to the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [61] on a suite of water resources applications. All of these MOEAs use real parameter simulated binary crossover (SBX), polynomial mutation, and elitism [5,62]. The primary goal in the development of the ε-NSGAII was to provide a highly reliable and efficient EMO algorithm which minimizes the need for traditional EA parameterization and allows the user to focus on problem specific search quality goals. Computational savings can be viewed in two contexts: (i) the use of minimal population sizes and (ii) the elimination of trial-and-error application runs to determine search parameters.

The ε-NSGAII builds on its parent algorithm, the NSGAII [59], by adding ε-dominance archiving [60,63] and adaptive population sizing [64] to minimize the need for extensive parameter calibration as demonstrated by Reed et al. [64]. The concept of ε-dominance allows the user to specify the precision with which they want to quantify each objective in a multiobjective problem. Fig. 1 demonstrates the concept of ε-dominance using a two step approach. First, a user specified ε grid is applied to the search space of the problem.

Larger ε values result in a coarser grid (and ultimately fewer solutions) while smaller ε values produce a finer grid. Grid blocks containing multiple solutions are then examined and only the solution closest to the bottom left corner of the block is kept (assuming minimization of all objectives). In the second step, non-domination sorting based on the grid blocks is then conducted resulting in a "thinning" of solutions and promoting a more even search of the objective space. Epsilon-dominance allows the user to define objective precision requirements that make sense for their particular application. The interested reader can refer to prior work by Laumanns et al. [63] and Deb et al. [60] for a more detailed description of ε-dominance.

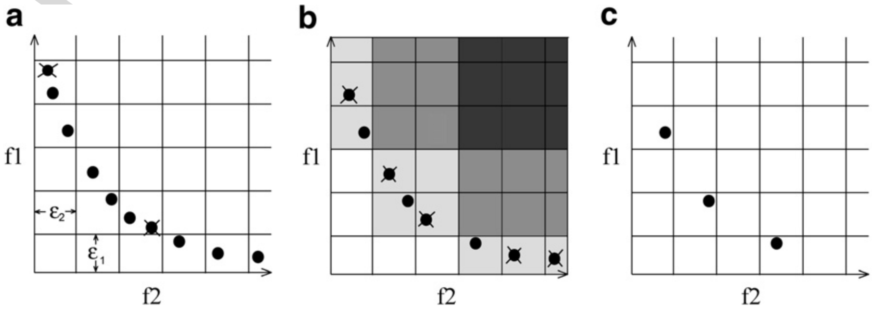

Fig. 1. Illustration of the ε-dominance concept. Step 1 (a), step 2 (b), and the final result (c) following the application of ε-dominance. Figure is adapted from [16].

The ε-NSGAII uses a series of "connected runs" where small populations are initially exploited to pre-condition search and automatically adapt population size commensurate with problem difficulty. As the search progresses, the population size is automatically adapted based on the number of ε-non-dominated solutions that the algorithm has found. Epsilon-non-dominated solutions found after each generation are stored in an archive and subsequently used to direct the search. Although the adaptation of population size will differ depending on the random seed chosen, exploiting small populations to precondition search will on average greatly reduce computation times. Theoretically, this approach allows the population size to increase or decrease, and in the limit when the ε-dominance archive size stabilizes, the ε-NSGAII's "connected runs" are equivalent to a diversity-based EA search enhancement recommended by Goldberg [58] termed "time continuation". In this study, search was terminated across all runs (i.e., across all populations used) after a user-specified maximum run time has been reached.

The ε-NSGAII uses time continuation to enhance population diversity and extend the time of active exploration. Time continuation results from combining random solutions with ε-dominance archive members each time the population size is adapted. Each new population is equal to four times the current ε-dominance archive's size so that ε-non-dominated archive members compose 25% of the new population and the remaining 75% is composed of randomly generated solutions. This aspect of the ε-NSGAII's search is particularly important in the context of computational demands and parallelization. Time continuation allows the algorithm to maintain effective search for as long as is necessary or computationally tractable. For many water resources applications, problem difficulty and/or the computational expense of solution evaluations heavily impact search time constraints and solution quality. This paper specifically demonstrates parallelization schemes for the ε-NSGAII that effectively exploit the algorithm's ε-dominance archiving, adaptive population sizing, solution injection, and time continuation to dramatically improve solution quality, search efficiency, and algorithmic reliability.

2.3. Parallelization strategies for the ε-NSGAII

Parallel computing is well suited for overcoming the computational bottleneck posed by computationally expensive objective functions while also improving the effectiveness and reliability of EMO algorithms [6,65–67]. There are three main parallel paradigms for EMO algorithms [6,65]: the master–slave model, the multi-population model, and the diffusion model. These models can be hybridized and their implementations vary significantly in the computer science literature. In this study, the MS and MP models are adopted since they are better suited for smaller scale clusters (<50 processors). In this study, we used a Linux computing cluster with 133 computer nodes composed of dual or quad AMD Opteron processors and 64 GB of RAM. The cluster runs the GNU/Lunix operating system and is installed with the Argonne National Laboratory's Infinicon Message Passing Interface (MPI). The Infinicon MPI runs natively over a very high-bandwidth, ultra low-latency network interconnect. This application utilized up to 16 processors to represent the typical size of computer clusters.

2.3.1. The master–slave ε-NSGAII

The MS parallelization scheme (see Fig. 2) is the simplest potential parallelization strategy for EAs. In this scheme, the master processor has a fully functional version of the ε-NSGAII that uses slave processors to evaluate solutions and return objective values. The master processor uses the objective function values to perform all of the required evolutionary search operations (ranking, selection, mating, and mutation). In our master–slave version of the ε-NSGAII, the master processor also evaluates potential solutions while managing issues associated with slave synchronization and load balance (i.e., all slaves have approximately equal workload). When the master reaches the point of fitness function evaluation, it tells all of the processors to be ready to receive the individuals for evaluation and "passes a message" containing the variables of potential solutions to the slaves. After evaluating its own group of individuals, the slave sends the objective values back to the master and the master uses all of the evaluated objectives' values to continue the evolutionary process until a stop criterion is met. The MS paradigm has the advantage of being extremely easy to implement. Prior criticisms of the MS model in the EA literature (for reviews see [68,69]) include:

- Communication costs increase significantly with an increasing number of processors.
- The parallelization scheme does not change the serial algorithm's expected performance.
- In multiobjective applications, the slowest slave processor will control the overall efficiency and timing of the remaining processors because non-domination sorting is delayed until the slowest slave processor returns its evaluation.

Water resources researchers should carefully consider if these criticisms generalize to their own applications. For

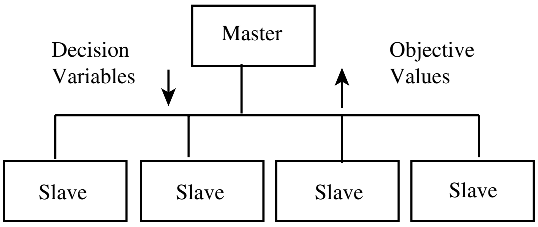

Fig. 2. Master–slave model paradigm.

example, while it is generally true that the master–slave parallelization scheme will result in the same search dynamics as the serial algorithm, it should be noted that the time savings associated with parallelization can allow for significantly more search. If an application requires 4 h to complete 100,000 evaluations, then four processors could potentially evaluate 400,000 solutions in the same 4 h period. This is particularly important for the ε-NSGAII where time-continuation can serve to maintain sufficient diversity to support more extensive search for as long as is necessary or feasible when solving water resources applications. Synchronization problems (i.e., search is delayed until the slowest slave processor send its evaluation) occur for MS applications when the average solution evaluation time T_f, has a high variance. Communication costs are of primary concern when judging the efficacy of a parallel application using the concept of speedup, S_p, defined in Eq. (1)

$$S_\mathrm{p} = \frac{T_\mathrm{s}}{T_\mathrm{p}} \tag{1}$$

Speedup is used to judge parallel performance by comparing the clock time required to solve an application in serial (i.e., on one processor), T_s, with the clock time required using multiple processors, T_p. As highlighted by Cantu-Paz [68], it is very important when judging speedup that solution quality should also be monitored. Monitoring solution quality will ensure that prematurely converged results with small clock times and poor solution quality do not bias speedup assessments. Ideally, the goal of parallelization is to attain "linear speedups" which means that when P processors are used to solve an application, the parallel computing time T_p, will equal $\frac{1}{P} \times T_\mathrm{s}$ (i.e., speedup is equal to the number of processors used). In each generation of MS search, Eq. (2) from Cantu-Paz [68] shows that the parallel processing time T_p, and speedups are a function of the population size n, the average solution evaluation time T_f, the number of processors P, and the communication time T_c associated with passing messages

$$T_\mathrm{p} = PT_\mathrm{c} + \frac{nT_\mathrm{f}}{P} \tag{2}$$

Eqs. (1) and (2) show that as the number of processors increase, there exists an asymptotic limit to the speed-ups that can be attained with the MS parallelization scheme. It should be noted that many water resources applications have solution evaluation times that are significantly larger than communication times (i.e., $T_\mathrm{c} \ll T_\mathrm{f}$) yielding very good speed-ups for master–slave EMO applications.

2.3.2. The multi-population ε-NSGAII

The MP parallelization scheme is often called the island model, which is a reference to the biological theory of punctuated equilibrium [70,71] formulated to model rapid evolutionary changes that result from isolated populations. The approach utilizes multiple populations distributed on different processors, each of which has a fully functional version of the ε-NSGAII. Recent literature reviews [68,6] highlight that multiple population schemes have emerged as one of the most popular parallelization strategies for evolutionary algorithms due to the emerging prevalence of distributed computing clusters and because early studies showed the potential for "superlinear" speed-ups (e.g., an algorithm is five times faster using only four processors). Some basic terms used in the multi-population EA literature are provided below:

- *Deme* refers to the population assigned to each processor.
- *Epoch* refers to the number of generations a deme evolves without communicating with other demes.
- *Topology* describes "processor connectivity" or the rules for communication between demes.
- *Migration rate* specifies how many solutions a deme will share with other demes.
- *Migration frequency* specifies how often migration will occur.

The terms defined above highlight that standard MP implementations of both single and multiobjective EAs significantly increase the design and parameterization issues that users must address to attain rapid and reliable search. The auto-adaptive population sizing, ε-dominance archiving, and solution injection methods used in the serial ε-NSGAII were generalized to a P-processor environment, simplifying the design and parameterization of the MP version of the algorithm. As illustrated in Fig. 3, the major difference between the MP version and the serial version of the ε-NSGAII is that the parallel algorithm implements adaptive population sizing, dynamic archiving, and solution injection across multiple processors. Worker processors and the coordinator evolve populations based on different random initializations. Each processor searches the entire parameter space for the Pareto front (i.e., every processor has its own fully operational version of the ε-NSGAII). Please note that the MP version of the ε-NSGAII automatically and dynamically adapts deme sizes, epochs, communication topologies, migration rates, and migration frequencies without additional user-inputs.

The MP version of ε-NSGAII uses asynchronous and dynamic messaging to automatically migrate solutions

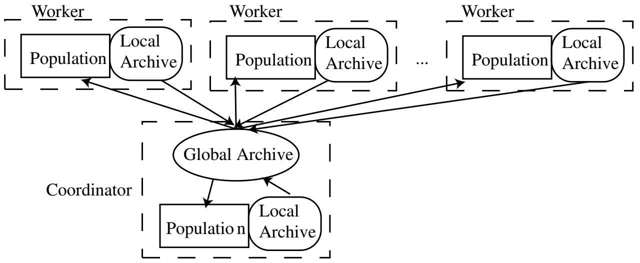

Fig. 3. Population–archive relationship for the MP parallelization scheme.

between processors and adjust deme sizes (see Fig. 3). Each processor starts with an arbitrarily small deme size (i.e., 12 members) and initiates search for 250 generations (i.e., the maximum epoch). After the first full epoch, the worker processors can then send requests to the coordinator for global archive solutions (initially representing a fully connected topology and migration strategy). Next, the coordinator collects the solutions from all of the local workers' archives to update the global archive using ε-non-domination sorting. The coordinator then sends the global archive solutions back to the processors who requested population changes. The new population sizes for each worker are four times the updated global archive size. One fourth of the individuals are injected from the global archive into each new population, and the remaining individuals are generated randomly. After the first 250 generation epoch, the worker processors can make asynchronous and dynamic population sizing requests if within 10 generations they fail to improve more than 10% of their local ε-dominance archived solutions. When search progress is insufficient, each worker will send a request to the coordinator and will receive a new population of which 25% of the solutions are global archive members and the remaining 75% are randomly generated. Migration rate and frequency are determined dynamically based on each deme's search progress as measured by how quickly it updates its local archive. Overall, search termination is the responsibility of the coordinator processor once the termination criterion is satisfied (i.e., maximum clock time). The coordinator sends out termination messages to all of the workers if the criterion is met. The communications of global archive solutions or termination signals are conducted by using token loops [72] to avoid processor deadlocks.

Attaining efficient speedups for a multiple population EA is challenging and related directly to computing hardware, decisions on parameter settings, and the difficulty of the application being solved. Eq. (3) provides a simplified but informative [68] description of the factors impacting parallel computation time T_p, for MP parallelization schemes

$$T_\mathrm{p} = gn_\mathrm{d}T_\mathrm{f} + (r-1)T_\mathrm{c} \tag{3}$$

The total parallel computation time for a multiple population EA is the sum of the time required for each deme to evolve ($gn_\mathrm{d}T_\mathrm{f}$) and the time spent communicating between processors ($(r-1)T_\mathrm{c}$). The component of parallel computation time spent evolving demes is given by the product of the deme size n_d, the number of generations g, and the average time required per evaluation T_f. The second component of Eq. (3) approximates the total time spent sharing or communicating solutions among $r-1$ demes. T_c is the average clock time required per communication. Eq. (3) highlights key design and parameterization issues that impact a multiple-population EA's ability to attain efficient speedups. Minimizing the deme size (n_d) will help to reduce parallel computation times and enhance speedups. Unfortunately, a small deme or population size will often cause EAs to have a low reliability [73]. Substitution of Eq. (3) into Eq. (1) also shows that as communication between processors increases, there will be a degradation in parallel speedup. This degradation causes parallel performance to again be "asymptotic" as the number of processors increase. In other words, increasing communication costs by using more processors will result in an upper bound limit where adding more processors will not improve T_p. The speedup asymptote is largely controlled by the ratio of function evaluation time and communication time. It is easier to attain efficient speedups when this ratio is large (i.e., $T_\mathrm{f} \gg T_\mathrm{c}$).

2.4. Case studies

The test cases used in this study were selected to encompass a broad range of problem properties. The DTLZ6 test function [74] is a benchmark three-objective problem in the EMO literature with a known Pareto front. DTLZ6 is one of the most difficult problems in the DTLZ suite due to its deceptive and multimodal search space (i.e., false fronts that cause premature convergence). Prior literature [60,74] has shown that modern EMO algorithms are unreliable in exactly quantifying DTLZ6's global Pareto surface. The second case study is a benchmark hydrologic model calibration problem based on long-term data for the Leaf River in Collins, Mississippi. The Leaf River test case has been used extensively in the water resources literature to develop and test optimization algorithms [3,9–11,13,14]. More recently, Tang et al. [3] have shown that the Leaf River case study is difficult for modern serial EMO algorithms, including the ε-NSGAII, to solve reliably. The Leaf River case study represents an unconstrained, continuous problem that is both deceptive and multimodal (see [3]). The final test case, a long-term groundwater monitoring (LTM) application represents a challenging discrete and constrained problem that has been used extensively in the development and testing of the serial version of the ε-NSGAII [2,46]. Nearly half of this problem's search space is infeasible and it has a very large four-objective Pareto optimal surface that has been identified through enumeration. Each of these case studies and their problem formulations are presented in detail in Sections 2.4.1, 2.4.2, 2.4.3.

2.4.1. Case 1: Test problem DTLZ6

The three-objective formulation of DTLZ6 is shown in Eq. (4) where the x_i and θ_i are decision variables and f_i is the objective function

$$\begin{aligned}
\text{Min} \quad & f_1(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(\theta_1/2)\cos(\theta_2/2) \\
& \qquad \cdots\cos(\theta_{M-2}/2)\cos(\theta_{M-1}/2) \\
\text{Min} \quad & f_2(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(\theta_1/2)\cos(\theta_2/2) \\
& \qquad \cdots\cos(\theta_{M-2}/2)\sin(\theta_{M-1}/2) \\
\text{Min} \quad & f_3(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(\theta_1/2)\cos(\theta_2/2) \\
& \qquad \cdots\sin(\theta_{M-2}/2)
\end{aligned}$$

$$\text{where } \theta_i = \frac{\Pi}{4(1+g(r))}(1+2g(r)x_i), \quad \text{for } i = 2,3,\cdots,(M-1)$$
$$g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x_M}} x_i^{0.1}$$
$$0 \leqslant x_i \leqslant 1, \quad \text{for } i = 1,2,\ldots,n \tag{4}$$

There are 12 decision variables as recommended by Deb [8]. Although this is a three-objective problem, the true Pareto front of this test function is a curved line surrounded by larger false fronts. As illustrated in Fig. 4, EMO algorithms tend to pre-converge to the large local non-dominated areas intersecting the Pareto front. For example, if only solutions A and B on the true Pareto front are found, then a very large non-dominated region exists (indicated by the shaded area). The true Pareto optimal solutions comprise a very small fraction of the overall number of non-dominated solutions that exist between A and B yielding a high probability of failure for search algorithms.

2.4.2. Case 2: Model calibration in the 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 [3,9–14]. Readers interested in the full details of the Leaf River case study's dataset should reference earlier works (e.g., [75]). The Leaf River case study used in this paper has been developed based on the original studies used to develop and demonstrate MOSCEM-UA [13,14]. 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 [76], for more details on the model). The algorithm searched the same 13 SAC-SMA parameters (three parameters are commonly fixed a priori) and parameter ranges as recommended by Vrugt et al. [14]. The algorithm is tested based on its ability to quantify a two-objective tradeoff using a root-mean square error (RMSE) problem formulation. The first objective was formulated using a Box-Cox transformation of the hydrograph as recommended by Misirli et al. [77] 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 objective functions are shown in Eq. (5)

$$\begin{aligned} \text{Min} \quad f_1(\theta) &= \left[\frac{1}{N}\sum_{i=1}^{N}[Q_{\text{obs},i} - Q_{\text{sim},i}(\theta)]^2\right]^{1/2} \\ \text{Min} \quad f_2(\theta) &= \left[\frac{1}{N}\sum_{i=1}^{N}[T(Q_{\text{obs},i}) - T(Q_{\text{sim},i}(\theta))]^2\right]^{1/2} \end{aligned} \tag{5}$$

where $Q_{\text{obs},i}$ is the observed discharge at time i, $Q_{\text{sim},i}(\theta)$ is the simulated discharge, and N is the total number of time steps in the calibration period. T is the Box-Cox transformation function ($T(f) = [(f+1)^{\lambda} - 1]/\lambda$ where $\lambda = 0.3$).

A 65-day warm-up period was used based on the methodological recommendations of Vrugt et al. [14]. A ten-year calibration period was used from 1 October 1952 to 30 September 1962. In the application, 50 trials were used for the serial run (one processor) and multi-processor runs of the MS and MP versions of the ε-NSGAII (2, 4, 8, 16-processors settings were tested), yielding a total of 450 EMO algorithm trial runs. Each EMO algorithm trial run utilized a maximum of $P \times 100{,}000$ (P is the number of processors) SAC-SMA model evaluations, yielding a total of maximum 305,000,000 SAC-SMA model evaluations used in our Leaf River case study analysis. A best known approximation set was generated for this problem by conducting non-dominated sorting on the final results collected from all of the trial runs. The best known solution set for the Leaf River case study is shown in Fig. 5.

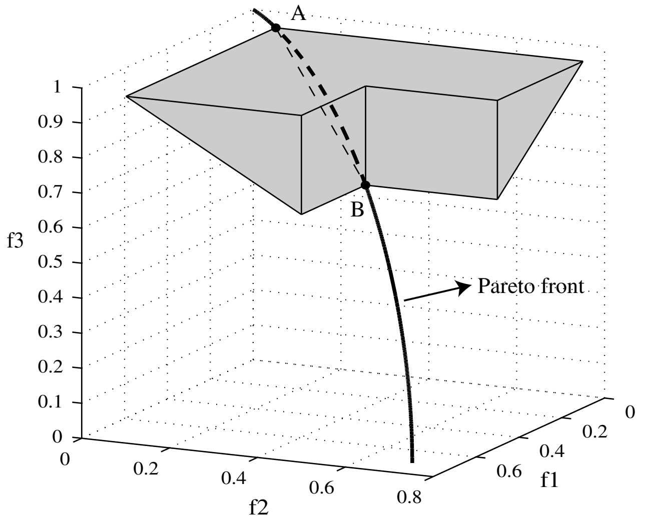

Fig. 4. Pareto front of DTLZ6 and the non-dominated region of point A and B on the front. The shaded region is the non-dominated region of A and B. Adapted from [8].

2.4.3. Case 3: Long-term groundwater monitoring design

The LTM test case used in this study is based on a 50-million node flow and transport simulation originally developed by Maxwell et al. [78]. This test case represents the migration of a hypothetical perchloroethylene (PCE) plume originating from an underground storage tank. The hydrogeology of the site has been extensively characterized and is based on a highly heterogeneous alluvial aquifer located at the Lawrence Livermore National Laboratory in Livermore, California. Concentration data are provided at 58 hypothetical sampling locations in a 29 well monitoring network for a snapshot in time 8 years following the initial release of contaminant. Each well can be sampled from one to three times along its vertical axis and the sampling domain extends 650 m in the x-direction, 168 m in the y-direction, and 38.4 m in the z-direction with

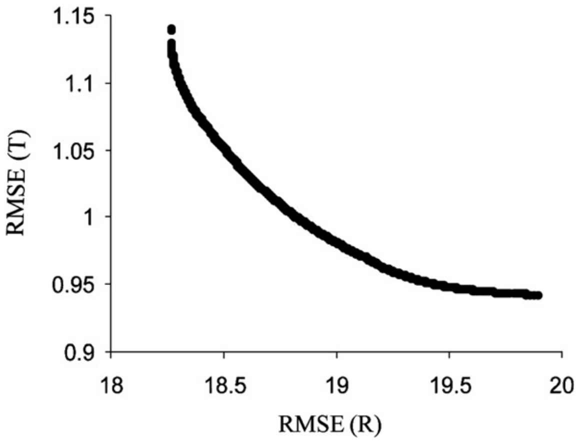

Fig. 5. Reference set generated for the Leaf River test case where RMSE(T) are the errors for the Box-Cox transform of the hydrograph and RMSE(R) are the errors for the raw hydrograph.

a minimum horizontal spacing of 10 m between wells. Additional details on this test case can be found in Reed et al. [15]. Four design objectives were chosen for this study, each of which was minimized: (i) sampling cost, (ii) relative error of local contaminant concentration estimates, (iii) local contaminant concentration estimation uncertainty, and (iv) contaminant mass estimation error. Objectives (ii)–(iv) were obtained using the Quantile Kriging method. Eq. (6) represents the objective formulation where $\mathbf{F}(\mathbf{x}_\kappa)$ is a vector valued performance function in which the four objectives: cost (f_{cost}), concentration estimation error (f_{conc}), local uncertainty (f_{uncert}), and mass estimation error (f_{mass}) are minimized. Eq. (7) subjects $\mathbf{F}(\mathbf{x}_\kappa)$ to the constraint that no points in the interpolation domain remain unestimated (which may occur if a particular sampling plan contains too few sampling points to successfully Krige the entire domain)

$$\text{Minimize}\quad \mathbf{F}(\mathbf{x}_\kappa) = (f_{\text{cost}}(\mathbf{x}_\kappa), f_{\text{conc}}(\mathbf{x}_\kappa), f_{\text{uncert}}(\mathbf{x}_\kappa), f_{\text{mass}}(\mathbf{x}_\kappa)),\quad \forall\kappa \in \Omega \tag{6}$$

$$\text{Subject to}\quad U(\mathbf{x}_\kappa) = 0 \tag{7}$$

The objectives are all a function of the vector $\mathbf{x}_\kappa$ representing the κth sampling plan in the decision space Ω. Each component i of a sampling plan κ is determined from Eq. (8) resulting in a string of binary digits indicating whether or not a well is sampled

$$x_{\kappa,i} = \begin{cases} 1, & \text{if the } i\text{th well is sampled} \\ 0, & \text{otherwise} \end{cases} \quad \forall\kappa, i \tag{8}$$

The sampling cost objective quantifies the monitoring cost of a particular sampling scheme using Eq. (9). The coefficient, C_S defines the cost per sample (normalized to one in this study). Additionally, if a well is sampled, it is assumed that all locations along its vertical axis are sampled resulting in a cost coefficient ranging from 1 to 3. The cost objective is ultimately quantified by summing the cost coefficients of each of the wells sampled in a particular scheme

$$f_{\text{cost}}(\mathbf{x}_\kappa) = \sum_{i=1}^{nwell} C_S(i) x_{\kappa,i} \tag{9}$$

The relative error of local contaminant concentration estimates objective measures how the Kriged estimate of the plume using the κth sampling plan differs from that obtained by sampling from all well locations. Eq. (10) quantifies the concentration error objective by summing the squared differences between the concentration estimate at a grid location $\mathbf{u}_j$ using all wells, $c_{\text{all}}(\mathbf{u}_j)$, and the concentration estimate at the same grid location using the κth sampling plan, $c_\kappa(\mathbf{u}_j)$

$$f_{\text{conc}}(\mathbf{x}_\kappa) = \sum_{j=1}^{n_{\text{est}}} \left(c_{\text{all}}(\mathbf{u}_j) - c_\kappa(\mathbf{u}_j)\right)^2 \tag{10}$$

Local contaminant concentration estimation uncertainty is quantified by summing the estimation standard deviations obtained from Kriging at each grid location $\mathbf{u}_j$ using Eq. (11). The standard error weight coefficient, A_j, can be used to assigned importance to uncertainty estimates at different locations in the interpolation domain. For this study, A_j was assumed constant across the interpolation domain and was assigned a value of $2\sqrt{3}$ based on the standard deviation of a uniform distribution

$$f_{\text{uncert}}(\mathbf{x}_\kappa) = \sum_{j=1}^{n_{\text{est}}} A_j \sigma(\mathbf{u}_j) \tag{11}$$

The contaminant mass estimation error objective quantifies the relative error between the total mass of dissolved contaminant estimated using all well locations, Mass_{all}, and the contaminant mass estimated from the κth sampling plan, Mass_κ. Eq. (12) expresses the relative mass estimation error in terms of a percentage

$$f_{\text{mass}}(\mathbf{x}_\kappa) = \left|\frac{\text{Mass}_{\text{all}} - \text{Mass}_\kappa}{\text{Mass}_{\text{all}}}\right| \cdot 100\% \tag{12}$$

If a well sampling scheme results in unestimated points in the interpolation domain (violating the constraint described by Eq. (7)), the objectives are penalized to ensure that infeasible sampling schemes are eliminated from consideration. Eq. (13) is applied to each objective function if a feasibility violation occurs, resulting in solutions with lower fitness (i.e., higher objective values in a minimization problem)

$$\mathbf{F}_{\text{penalty}}(\mathbf{x}_\kappa) = \begin{cases} f_{\text{cost}}^{\text{penalty}} = f_{\text{cost}} + f_{\text{cost}}^{\max} \\ f_{\text{conc}}^{\text{penalty}} = f_{\text{conc}} + n_{\text{est}} + U(\mathbf{x}_\kappa) + f_{\text{cost}}^{\max} \\ f_{\text{uncert}}^{\text{penalty}} = f_{\text{uncert}} + n_{\text{est}} + U(\mathbf{x}_\kappa) + f_{\text{cost}}^{\max} \\ f_{\text{mass}}^{\text{penalty}} = f_{\text{mass}} + n_{\text{est}} + U(\mathbf{x}_\kappa) + f_{\text{cost}}^{\max} \end{cases} \tag{13}$$

Eq. (13) penalizes the objective functions based on the maximum cost of a sampling scheme, the total number of estimation points in the grid, and the total number of unestimated points, $U(\mathbf{x}_\kappa)$, in the infeasible sampling plan. Penalizing solutions rather than eliminating them ensures that sampling schemes which are "almost" feasible are given the opportunity to be further evolved by the MOEA into feasible designs (for more details on this problem formulation, see Reed and Minsker [28]).

Spatial interpolation of the contamination plume was conducted using Quantile Kriging (QK) based on the recommendations of Reed et al. [15]. Kriging provides a minimum error variance estimate value at an unsampled location provided the data at the sampled locations [79]. QK extends Ordinary Kriging (OK) by transforming the sample values to quantile space according to their rank. The quantile values represent the empirical cumulative distribution function (CDF) of the sample values, resulting in normalized data. Samples are Kriged in quantile space and then transformed back to concentration space using the generated CDF [80,81]. Since OK assumes stationarity of the concentration mean, moving local neighborhoods are used to estimate the expected value at each location [79]. Reed et al. [15] found that QK showed the least bias with respect to variability of PCE concentrations and preferential sampling, and was most robust in representing the plume when compared to five other interpolation methods.

2.5. Performance metrics

When judging the performances of the parallel versions of the ε-NSGAII, it is important to monitor both speedup and solution quality. For EMO applications, solution quality metrics must consider both convergence and diversity. Convergence metrics quantify how distant an approximation set is from the reference set of optimal solutions and diversity metrics measure how well the approximation set's solutions capture the full extent of the Pareto front. In this study, runtime convergence and diversity metrics [82] were used to measure these aspects of algorithm performance separately. The runtime ε-performance [1] and ε-indicator [83,84] metrics were also used in this study to quantify these aspects of performance simultaneously.

The runtime convergence metric proposed by Deb and Jain [82] is used to quantify the average convergence of the algorithm to some reference set. This metric measures convergence using the average normalized Euclidean distance between the algorithms solutions and the optimal reference set of solutions. This metric was normalized to have a minimum value of zero for perfect convergence and a maximum value of one indicating very poor convergence. The diversity metric also proposed by Deb and Jain [82] measures how well the approximation set captures the full extent of the tradeoffs between a case study's objectives. Deb and Jain's diversity metric ranges from a maximum of one to a minimum of zero. A value of one represents a perfect solution diversity in terms of the metric formulation and user specified parameters. The metric is calculated by projecting the non-dominated solutions obtained during a single run of an EMO algorithm as well as the reference solutions onto a hyper-plane that is partitioned into grids. The metric is then computed by counting the number of non-dominated solutions which fall into the same grids as the solutions in the reference set. A detailed description of this metric can be found in [82].

The ε-performance metric proposed by Kollat and Reed [1] accounts for both convergence and diversity simultaneously. This metric ranges between zero and one where a metric value of one would indicate 100% convergence to within user specified ε values of the reference set. Epsilon-performance counts the number of approximate set solutions that fall into ε hyperboxes defined around the reference set and is calculated as the percentage of solutions which have successfully converged to the Pareto front within the user specified precision. For more details on this metric, see [2]. The ε-indicator metric [83,84] represents the smallest distance that an approximation set must be translated to dominate the reference set, implying that smaller indicator values are preferred. Theoretically, the ε-indicator accounts for both convergence and the distribution of the solutions. For a more detailed description of this metric, see [83,84].

2.6. Description of computational experiment

The ε-NSGAII used simulated binary crossover and polynomial mutation operators for all three test cases. Based on the most commonly recommended settings for these operators [8,31,59,82,85], the crossover probability was set equal to 1.0 and the probability of mutation was set equal to $1/L$ where L is the number of decision variables. The distribution indices for crossover and mutation were set equal to 15 and 20, respectively. An initial population size of 12 was used for the MS and the MP versions of the ε-NSGAII in all applications. Table 1 provides a detailed list of the parameters used for both parallel versions of the ε-NSGAII.

Epsilon resolution settings were chosen based on application specific resolution goals. For the DTLZ6 test case, uniform ε values equal to 0.0045 were specified for all objectives so that the Pareto optimal set could be represented discretely by 100 solutions. For the Leaf River test case, the ε values were set to 10^{-6} representing the minimum meaningful RMSE difference that should be resolved. In the LTM test case, ε values were specified so that the

Table 1
Summary of parameter settings for both the MS and MP versions of the algorithm and metric calculations

Parameters	DTLZ6	Leaf	LTM
Crossover probability	1.0	1.0	1.0
Mutation probability	0.0833	0.077	0.04
Crossover dist. index	15	15	15
Mutation dist. index	20	20	20
ε for algorithm	0.0045[a]	10^{-6}[a]	$[1.0\ 10^{-5}\ 10^{-2}\ 10^{-6}]$
Maximum NFE	$P \times 10^{6}$	$P \times 10^{5}$	$P \times 4 \times 10^{5}$
ε for metric calc.	0.0045[a]	10^{-3}[a]	$[1.0\ 10^{-5}\ 10^{-2}\ 10^{-6}]$

NFE = number of function evaluations; P = number of processors.

[a] The same ε value is used for all the objectives.

ε-NSGAII could quantify the Pareto optimal set at the same resolution as was used in our enumeration. Each of the four objectives had the following ε values ($\varepsilon_{\text{cost}} = 10^0$, $\varepsilon_{\text{conc}} = 10^{-5}$, $\varepsilon_{\text{uncert}} = 10^{-2}$, and $\varepsilon_{\text{mass}} = 10^{-6}$) respectively for each of the four objectives as were used in [2].

Termination was based on the maximum number of function evaluations that the MS and MP versions of the ε-NSGAII could use in a trial run. The termination criteria have been carefully designed based on previous studies [2,3,8] and problem difficulty while also considering limits on computing resources. In all applications, evaluations of speedup and performance metrics were based on 50 random trial runs for both the MS and MP versions of the ε-NSGAII. The performance of both parallelization strategies were analyzed for 1, 2, 4, 8, and 16 processor test cases to represent small-scale computing clusters and to maintain feasible queue times for the trial runs. The termination criteria are summarized in Table 1. For the DTLZ6 case study and the LTM case study, the ε values of the ε-performance metric calculation were set to be equal to those of the algorithms ε-dominance values as described above. For the Leaf River test case, lower resolutions were used for the ε-performance metric calculation since many of the runs could not find reference solutions at 10^{-6} accuracy (see Table 1).

3. Results

Section 3.1–3.3 present optimization results for each of the three case studies used to test the MS and MP versions of the ε-NSGAII. Each of the sections provides detailed tables of the average metric performance, run-time plots of search dynamics, and plots of success rates (defined in terms metric value goals). The Mann–Whitney test [86] was used to validate if differences in the distributions of metric values attained at the same cutoff times for the MS and MP versions of the ε-NSGAII were statistically significant. The null hypothesis for the tests assumed that any two metric distributions were the same. In the results below, we report when there was at least a 95% confidence that the null hypothesis was rejected (i.e., the metric distributions are significantly different). The Mann–Whitney tests were used in two contexts: (1) inter-comparisons of the MS and MP versions of the ε-NSGAII at the same number of processors and (2) intra-comparisons that analyzed how increasing processor numbers improved the individual parallelization strategies relative to themselves.

3.1. Optimization results for case study 1: DTLZ6

Recall from Table 1 that the total number of function evaluations used to solve DTLZ6 was allowed to vary with increasing numbers of processors (i.e., $\text{NFE} = P \times 10^6$). The DTLZ6 problem is one of the most difficult test functions available in the EMO literature [74] and provides an excellent test of how problem difficulty impacts the efficiency and reliability of the MS and MP versions of the ε-NSGAII. Overall, Table 2 shows that the MP version of the ε-NSGAII had the best overall average metric values and that performance improved markedly with the increased search afforded by increased processors. All versions of the ε-NSGAII struggled to reliably solve DTZL6, particularly the MS strategy. Mann–Whitney inter-comparisons of the MS and MP strategies showed that more than two processors are required for the methods to yield statistically meaningful performance differences for DTLZ6. In general, the MP strategy was superior to the MS approach for all performance metrics when four or more processors were used in search. The Mann–Whitney inter-comparison confirmed the MP strategy's superiority at greater than a 97% confidence level.

The Mann–Whitney intra-comparisons indicated that an increase in the number of processors and the concomitant increase in NFE led to improved metrics' distributions for either the MS or the MP versions of the ε-NSGAII. As indicated in Table 2 and confirmed with the Mann–Whitney intra-comparisons, increasing the number of processors used by the MS strategy did not yield statistically

Table 2
DTLZ6 case study averages (AVG) and standard deviations (STD) in terms of the convergence (Conv.), diversity (Div.), ε-indicator (Eind.), and ε-performance (Eperf.) metrics

#P	Strategy	Conv. ($\times 10^{-2}$) Ideally = 0		Div. ($\times 10^{-2}$) Ideally = 1		Eind. ($\times 10^{-2}$) Ideally = 0		Eperf. ($\times 10^{-2}$) Ideally = 1	
		AVG	STD	AVG	STD	AVG	STD	AVG	STD
1P	MS/MP	4.92	2.37	27.4	7.15	3.73	1.64	5.67	22.7
2P	MS	4.92	2.37	27.6	7.01	3.73	1.64	5.67	22.7
	MP	4.51	2.25	29.4	7.16	3.48	1.58	7.56	25.9
4P	MS	4.91	2.37	27.6	6.97	3.73	1.64	5.67	22.7
	MP	3.29	2.27	34.1	8.71	2.62	1.58	20.8	39.5
8P	MS	4.91	2.37	27.5	6.98	3.73	1.64	5.67	22.7
	MP	3.05	2.26	34.7	9.31	2.45	1.55	26.5	42.9
16P	MS	4.92	2.37	27.5	7.05	3.73	1.64	5.67	22.7
	MP	**2.21**	1.54	**36.2**	8.46	**1.84**	1.03	**26.5**	43.0

All metrics were computed using 50 random trials. MS and MP designate master–slave and multiple-population versions of the ε-NSGAII. The results are shown for 1, 2, 4, 8, and 16 processors. The best overall metrics are shown in underlined bold.

significant performance improvements. The implication of this result is that the serial version of the ε-NSGAII will not improve its ability to solve DTLZ6 with increases in search duration (i.e., using time continuation). This demonstrates that for extremely challenging problems where the serial version of the ε-NSGAII fails, the MS parallelization strategy will not improve search results.

Alternatively, the Mann–Whitney intra-comparisons for the MP version of the ε-NSGAII showed statistically significant improvements in all metrics with increasing processor count. Fig. 6 illustrates how processor count impacted the success rates for both the MS and MP versions of the ε-NSGAII. The ε-performance metric was used in Fig. 6 because it best captured algorithmic reliability and showed similar dynamic trends compared to the other performance metrics. Fig. 6 clearly shows that the MS parallelization strategy did not improve the ε-NSGAII's success rate regardless of increasing processor count and search evaluations (e.g., 16 processors used 16 million function evaluations). Fig. 6 shows that the MP parallelization strategy yielded more than a fourfold increase in success rate relative to the serial version of the ε-NSGAII. Mann–Whitney intra-comparisons for the MP strategy showed that increasing the processor count beyond eight did not yield statistically significant improvements in performance. Please note that speedup results were not presented for DTLZ6 since function evaluation time (i.e., T_f in Eqs. (2) and (3)) was negligible compared to those used in the Leaf River and LTM case studies.

3.2. Optimization results for case study 2: Leaf river calibration application

Tang et al. [3] used 100,000 model evaluations per ε-NSGAII trial run when calibrating the SAC-SMA hydrologic model of the Leaf River and showed that the algorithm had a relatively poor success rate. Recall that the Leaf River application is an unconstrained, continuous space problem with a potentially infinite solution space. Two key questions must be considered with the ε-NSGAII's modest success rate: Is the Leaf River problem so difficult that the ε-NSGAII will always fail to reliably approximate the best known Pareto set, or, does the problem simply require much longer periods of search facilitated by the ε-NSGAII's use of time-continuation? If true, the first question implies that the MP version of the ε-NSGAII should show superior performance as was observed for DTLZ6. Otherwise if the second question is true, than the MS version of the ε-NSGAII should be competitive if not superior to the MP version. Table 3 shows that in fact, the MS version of the ε-NSGAII was able to attain superior average scores for all of the performance metrics except convergence. When analyzing Table 3, recall that the MS and MP versions of the ε-NSGAII used a total NFE equal to $P \times 10^5$ for the Leaf River case (where P is processor count).

Mann–Whitney inter-comparisons between the MS and MP schemes for the Leaf River case study showed that performance differences were significant at greater than a 95% confidence level. Table 3 and the Mann–Whitney intra-comparisons show that increasing the processor count always improved the average performance metrics for the MS strategy. Distributional differences were validated at greater than a 98% confidence level for all of the performance metrics. The MP scheme required at least four processors to attain results that were statistically different from those attained with the serial version of the ε-NSGAII. Increasing the processor count beyond four processors, generally improved the MP scheme's metric's values at the 99% confidence level.

Fig. 7 presents run-time ε-indicator dynamic results for the full distribution of random trials used at every processor count. The run-time dynamics provide a more detailed description of the MS and MP schemes' dynamics and reliabilities. The results of all 50 random trial runs for each parallelization scheme and processor count are shown in the figure. The ε-indicator metric plots are representative of the performance dynamics observed for all of the other performance metrics except convergence. Readers should exercise caution when interpreting or using the convergence

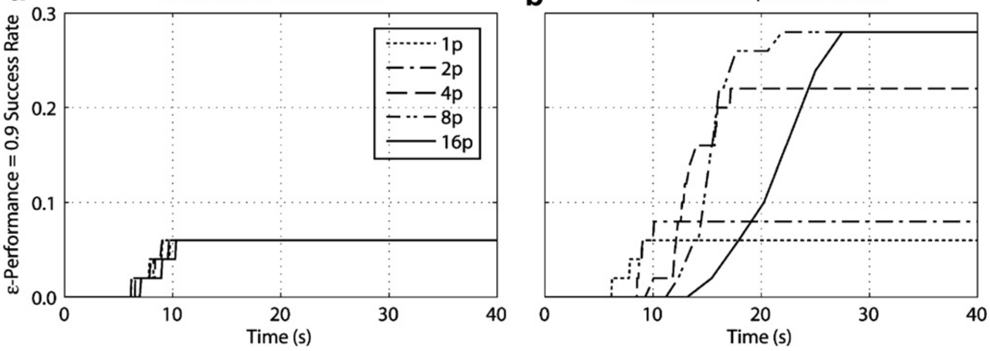

Fig. 6. Dynamic success rate plots for (a) the MS and (b) the MP configurations of the ε-NSGAII. Success rates are computed as the percentage of 50 trials that were able to attain an ε-performance value of 0.9. The success rates are shown as a function of computing clock time and processor count. Please note that 50 trial runs were used at each processor count.

Table 3
Leaf River case study averages (AVG) and standard deviations (STD) in terms of the convergence (Conv.), diversity (Div.), ε-indicator (Eind.), and ε-performance (Eperf.) metrics

#P	Strategy	Conv. ($\times 10^{-2}$) Ideally = 0		Div. ($\times 10^{-2}$) Ideally = 1		Eind. ($\times 10^{-2}$) Ideally = 1		Eperf. ($\times 10^{-2}$) Ideally = 0	
		AVG	STD	AVG	STD	AVG	STD	AVG	STD
1P	MS/MP	4.47	6.19	71.1	7.54	18.6	14.5	0.00	0.00
2P	MS	3.55	5.84	77.5	8.74	13.3	13.3	0.09	0.33
	MP	5.74	7.21	72.2	9.95	19.7	16.5	0.00	0.00
4P	MS	2.56	5.36	81.5	8.85	9.60	12.2	0.67	1.01
	MP	2.04	3.64	80.1	5.52	9.75	7.40	0.00	0.00
8P	MS	1.88	4.81	86.1	8.96	6.66	11.7	4.53	3.91
	MP	1.53	2.58	83.7	4.32	7.42	6.19	0.19	0.61
16P	MS	1.69	4.96	**91.3**	9.08	**4.75**	11.6	**20.6**	9.82
	MP	**0.96**	0.17	86.9	4.40	5.51	2.80	0.26	0.78

All metrics were computed using 50 random trials. MS and MP designate master–slave and multiple-population versions of the ε-NSGAII. The results are shown for 1, 2, 4, 8, and 16 processors. The best overall metrics are shown in underlined bold.

metric because as its name implies scores only require a small number of points to be close to the reference set independent of how well they capture the full extent of its tradeoffs.

Generally both parallelization schemes have similar ranges in performance until greater than 8 processors were used. The individual trace lines for each trial run plotted in Fig. 7 provide a more detailed understanding of why the MS strategy has better final average metrics. Although a small number of trials fail, the overall distribution of MS trials had dramatically better final metric values. Fig. 7 highlights that the two parallelization strategies lead to very different search dynamics. Fig. 7a and b are best interpreted by looking at the serial single processor search results and analyzing the influence of the increasing processor count. In particular, an increase from 2 to 16 processors results in a clear shift in the search traces towards earlier time. These shifts towards earlier time represent the impacts of search speedup and indicate that improved ε-indicator values are being found earlier in time. Both Fig. 7a and b show that perceived "failures" of the serial version of the ε-NSGAII are largely a function of time of search. As speedups increase, the percent of time dedicated to new search increases and the number of failures decreases.

Fig. 8a and b clearly show that the MS scheme is able to attain high success rates more rapidly than the MP scheme with increasing processor counts. For example, for a processor count of 16 (i.e., NFE = 16 × 10^5) the MS strategy attains an 80% success rate in half of the time required by the MP scheme. In both Fig. 8a and b the general trend of increasing success rates in shorter time periods reflects the importance of speedup in allowing either parallelization scheme to actively search new regions of the Leaf River case study's search space. Another key result shown in Fig. 8a and b is that for water resources users constrained to using two-processor workstations, the MS strategy clearly outperforms MP search.

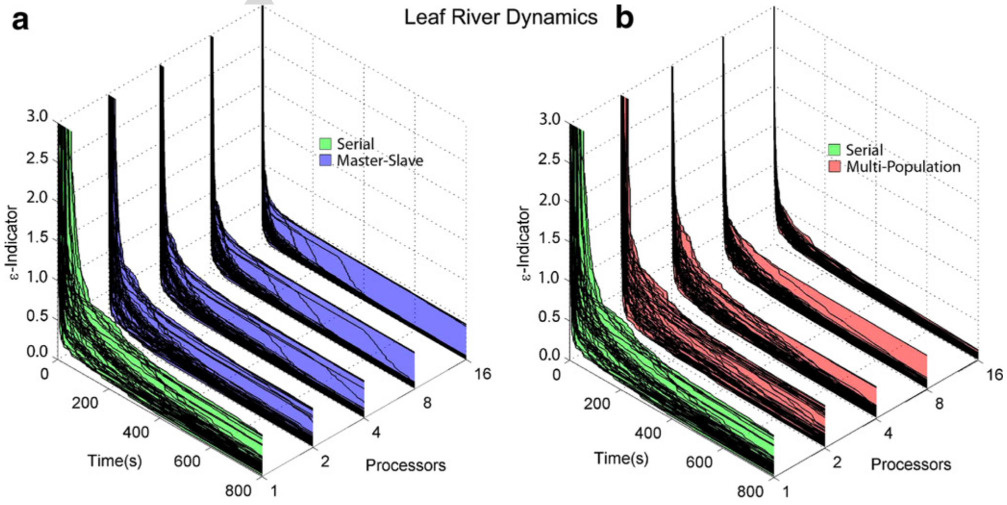

Fig. 7. Dynamic performance plots of ε-indicator for (a) the MS and (b) the MP versions of the ε-NSGAII. Each random trial is indicated with a solid line and the shaded regions show the ranges in performance. The ε-indicator values are shown as a function of computing clock time and processor count. Please note that 50 trial runs were used at each processor count.

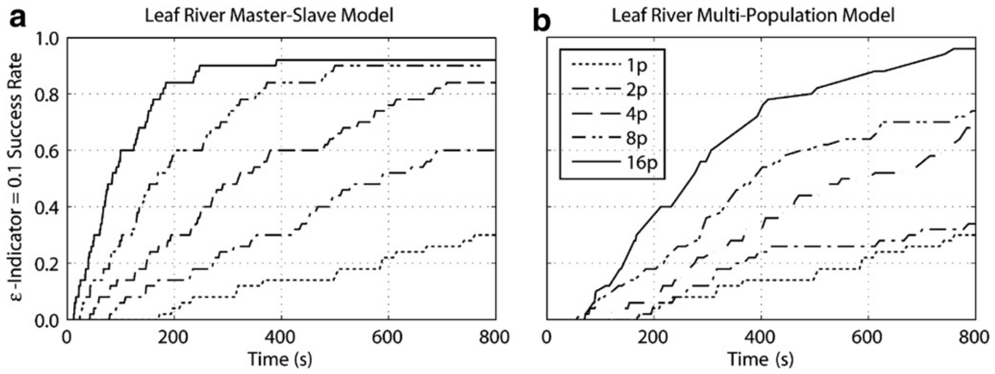

Fig. 8. Dynamic success rate plots for (a) the MS and (b) the MP configurations of the ε-NSGAII. Success rates are computed as the percentage of 50 trials that were able to attain an ε-indicator value of 0.1. The success rates are shown as a function of computing clock time and processor count. Please note that 50 trial runs were used at each processor count.

Judging speedup in parallel EMO applications is particularly challenging since solution quality must be considered explicitly. Solution quality judgments depend on the metrics being used. Fig. 9 illustrates this issue for both parallelization strategies. In Fig. 9, speedups were computed as the ratio of the average serial solution time T_S, versus the average parallel solution time T_P, required to attain each level of ε-indicator (see Eq. (1)). When interpreting the results shown in Fig. 9 it is important to remember what aspect of EMO performance is being captured by each of the performance metrics. The ε-indicator performance metric requires close proximity and a diverse distribution of solutions to attain near zero results. Fig. 9a indicates that the MS scheme's final metric values were attained with linear or near linear speedups. In Fig. 9a the 8-processor and 16-processor speedups generally increase with more stringent metric values (i.e., as the metrics approach zero). Sub-linear speedups result from increased communication costs and the fact that the serial version of the ε-NSGAII quickly reduces the metric from 8 to 1 in the time allocated. Fig. 9b demonstrates that the MP version of the ε-NSGAII results in very different search dynamics and speedups. The most interesting result in the figure is that the speedup of the 16-processor case reaches 18 for ε-indicator metric. This result indicates that multiple populations increase search diversity and in some cases can in fact attain superlinear speedups. Note Fig. 9b also shows that the MP version of the ε-NSGAII exhibits poor speedups early and late in the runs. This result supports our observations in Fig. 7, which show that reliably solving the Leaf River calibration problem requires sustained, high quality speedups to allow the ε-NSGAII to search new areas of the solution space. The poor speedups for the MP scheme can be attributed largely to the redundancy of search across processors and communication costs as has been observed in other studies [66].

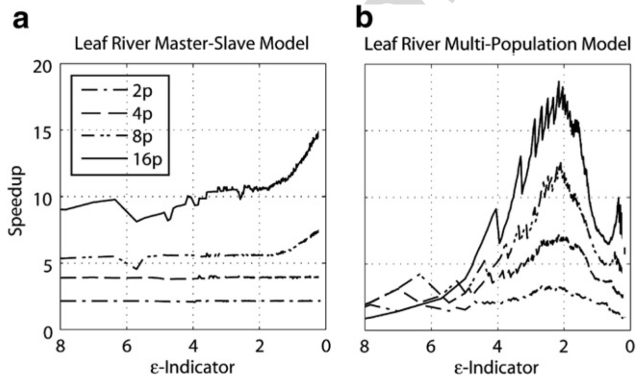

Fig. 9. Joint plots of average speedup versus average solution quality for (a) the MS and (b) the MP configurations of the ε-NSGAII with respect to the ε-indicator metric. The averages are computed for 50 trial runs at each processor count. Speedups were computed as the ratio of the average serial time (T_S) versus the average parallel time (T_P) required to attain each level of ε-indicator.

3.3. Optimization results for case study 3: long-term monitoring application

The LTM application represents an interesting contrast to the continuous, deceptive, and multimodal problem properties for the Leaf River calibration case study. The LTM application represents a discrete, non-deceptive constrained space that must be searched to identify a four-objective Pareto optimal set. Enumerative analysis has shown that nearly half of the decision space is infeasible and that the Pareto optimal set presents scaling challenges where objectives' values range over several orders of magnitude [2,16]. The average LTM solution evaluation time is approximately twice as long as solution evaluations for the Leaf River. Recall from Table 1 that both the MS and MP versions of the ε-NSGAII used a total NFE equal to $P \times 400{,}000$ (where P is processor count). Table 4 provides the average values of the performance metrics attained by both parallelization schemes for an increasing processor count. The table shows that the MS and MP schemes attained very similar average final metrics. Mann–Whitney inter-comparisons between the parallelization schemes showed that only the final ε-performance values' statistical distributions were significantly different. The

Table 4
LTM case study averages (AVG) and standard deviations (STD) in terms of the convergence (Conv.), diversity (Div.), ε-indicator (Eind.), and ε-performance (Eperf.) metrics

#P	Strategy	Conv. ($\times 10^{-4}$) Ideally = 0		Div. ($\times 10^{-2}$) Ideally = 1		Eind. ($\times 10^{-2}$) Ideally = 0		Eperf. ($\times 10^{-2}$) Ideally = 1	
		AVG	STD	AVG	STD	AVG	STD	AVG	STD
1P	MS/MP	6.74	1.27	79.8	1.87	99.8	1.10	73.8	1.21
2P	MS	4.65	1.10	82.6	1.36	98.5	4.21	80.3	0.75
	MP	4.42	0.90	83.3	1.66	97.7	4.44	80.5	0.82
4P	MS	2.98	0.89	85.1	1.02	88.5	9.45	85.4	0.45
	MP	3.18	0.83	85.2	1.37	88.0	11.6	84.7	0.54
8P	MS	2.39	0.67	86.8	1.11	81.2	13.7	88.5	0.36
	MP	2.72	0.78	86.6	0.92	77.0	14.8	87.2	0.56
16P	MS	**2.16**	0.60	**88.4**	1.07	**66.8**	21.6	**90.8**	0.42
	MP	2.42	0.83	88.0	1.15	70.8	18.6	89.2	0.45

All metrics were computed using 50 random trials. MS and MP designate master–slave and multiple-population versions of the ε-NSGAII. The results are shown for 1, 2, 4, 8, and 16 processors. The best overall metrics are shown in underlined bold.

ε-performance metric is the most stringent of the metrics in terms of both convergence and diversity because the algorithms must find solutions that fall within very small ε-hyperboxes of the enumerated Pareto set for this test case. The Mann–Whitney intra-comparisons showed that increasing the processor count (and implicitly the NFE used) improved all metrics for both parallelization schemes with performance differences validated at greater than a 99% confidence level. The LTM results in Table 4 clearly show that the ε-NSGAII's performance is enhanced by the increased search afforded by parallelization, but they do not clearly differentiate the performances of the MS and MP strategies.

The run-time ε-performance plots presented in Fig. 10 provide a more detailed description of the MS and MP strategies' dynamics. Epsilon-performance was selected because this metric shows statistically significant performance differences between the MS and MP strategies. Although the final metrics for each of the parallelization strategies are very similar, Fig. 10 shows that they produced very different ranges of performance with increasing processor counts. The MS strategy clearly scales well with increasing numbers of processors as evidenced by its rapid and reliable run-time dynamics for the 8 and 16 processor cases. As was observed for the Leaf River case study, increased processor counts allow the MS version of the ε-NSGAII to better exploit time-continuation, resulting in the exploration of new regions of the decision space much earlier in the allocated 6000 s of run time.

Fig. 11 further differentiates the MS and MP strategies' performances using success rate plots. The success rate plots in Fig. 11 show the cumulative distributions of the run times required to approximate 80% of the LTM case study's Pareto optimal set. The steepness of the distributions gives a visual measure of the variance of run times (i.e., a perfectly vertical distribution would represent 50 trial runs with identical run times). Note that the serial results for the ε-NSGAII have been omitted from Fig. 11 since none of the single processor trials satisfied our success criterion in 6000 s. Figs. 10 and 11 show that two and four processor performance for the MS and MP strategies are very similar in terms of their reliability and speed. The largest performance differences between the MS and MP schemes resulted for the 8 and 16 processor trials. The MS strategy's runtime distributions for processor counts of 8 and 16 are nearly vertical, which implies that their runtimes were nearly independent of random seed effects. Additionally, MS trials for 8 and 16 processors were completed 2–3 times faster than the MP trials for the same processor counts. The results shown in Fig. 11 imply that the MS strategy is exploiting superior speedups to rapidly quantify the LTM case study's Pareto surface.

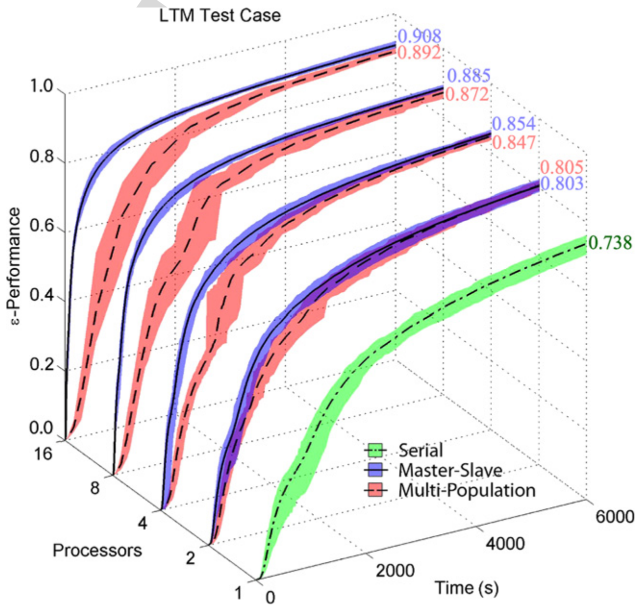

Fig. 10. Dynamic performance plots of ε-performance for the MS and MP versions of the ε-NSGAII. Mean performances are indicated as lines and the shaded regions show the ranges in performance. The ε-performance values are shown as a function of computing clock time and processor count. Please note that 50 trial runs were used at each processor count.

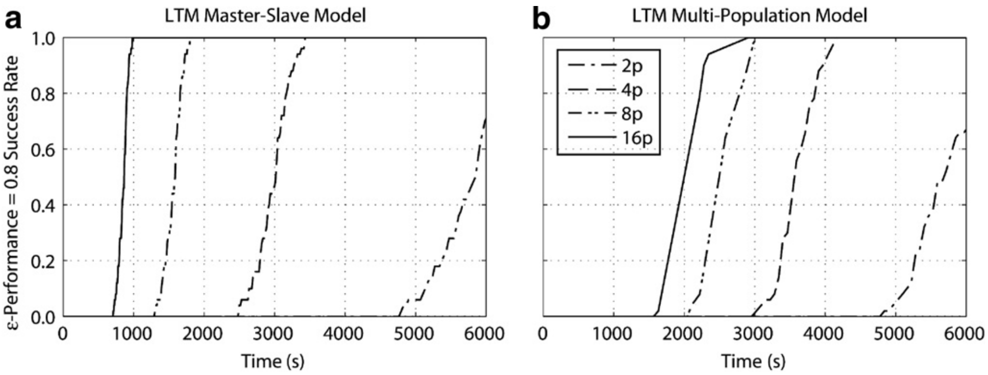

Fig. 11. Dynamic success rates for (a) the MS and (b) the MP configurations of the ε-NSGAII. Success rates are computed as the percentage of 50 trials that were able to attain an ε-performance value of 0.8. The success rates are shown as a function of computing clock time and processor count.

Fig. 12 confirms that the MS version of the ε-NSGAII was able to better sustain speedups over the full duration of runs. Fig. 12a shows that the MS scheme is able to attain nearly linear speedups for both performance metrics. The MS results exhibit the expected asymptotic limit on speedup (see Eqs. (1) and (2)) with the increased communication costs associated with increasing processor counts. Fig. 12a and b show that the two and four processor speedups for both the MS and MP strategies are comparable as can also be seen in Fig. 10. As shown in both Fig. 12a and b, although the MP version of the ε-NSGAII was able to enhance initial diversity which allowed it to identify feasible LTM solutions much more rapidly than the serial version of the ε-NSGAII, the MP strategy failed to sustain its search advantage when identifying solutions that are close in proximity and distributed uniformly over the LTM case study's four-objective Pareto surface. Overall, the MS version of the ε-NSGAII exhibits superior performance on the LTM case study, especially when considering its simplicity and ease-of-implementation relative to the MP scheme.

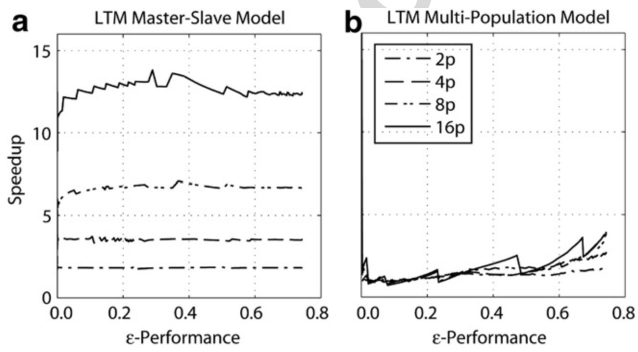

Fig. 12. Joint plots of average speedup versus average solution quality for (a) the MS and (b) the MP configurations of the ε-NSGAII with respect to the ε-performance metric. The averages are computed for 50 trial runs at each processor count. Speedups were computed as the ratio of the average serial time (T_S) versus the average parallel time (T_P) required to attain each level of ε-performance.

4. Discussion

Generally, there are three primary factors that should be considered when evaluating parallelization strategies for multiobjective water resources applications: (1) problem difficulty, (2) the parallelization schemes' ease-of-implementation, and (3) the parallel algorithm's ease-of-use. The DTLZ6 test problem is potentially one of the hardest test functions available and has been shown to cause very high failure rates in all of the currently available benchmark EMO algorithms [8]. This is evident from our own analysis shown in Fig. 4. The DTLZ6 problem serves two purposes in this study: (1) it provides a baseline upper bound for analyzing how problem difficulty impacts the relative performances of the MS and MP parallelization schemes and (2) it validates that the MP version of the ε-NSGAII is more effective than the MS scheme when solving extremely difficult problems. The success rates shown in Fig. 6 show that the MP parallelization scheme is far superior when solving DTLZ6. As has been highlighted in prior studies [6,68], the MP scheme significantly changes solution diversity and search dynamics, whereas the MS parallelization strategy only changes the duration of search. Although our dynamic deme-sizing and migration strategies simplify the parameterization requirements of the MP version of the ε-NSGAII, the parallelization scheme still requires a much higher degree of user sophistication than does the MS strategy. Multiple-population EMO algorithms take considerably more effort to implement and yield dramatically more complex speedup dynamics relative to MS schemes. Although the MP version of the ε-NSGAII was required to solve the DTLZ6 problem, the interesting issue that water resources scientists and engineers need to consider is: How many water resources applications are as difficult as DTLZ6?

This question is particularly relevant considering that the MP version of the ε-NSGAII was inferior to the MS scheme for the Leaf River and LTM applications. The Leaf River and the LTM test cases encompass two very different problem types from the water resources literature that have been shown to be challenging for modern EMO algorithms (e.g., see [2,3,14]). Figs. 8 and 11 show that the serial version of the ε-NSGAII had very low success rates when solving the Leaf River and the LTM test cases (defined in terms of attaining a goal performance metric value within a user defined runtime). In water resources applications where high EMO algorithm failure rates are observed, two key questions arise: Are these problems so difficult that the EMO algorithm will always fail to reliably approximate the best known Pareto set, or, do the problems simply require much longer periods of search facilitated by parallelization and time-continuation?

Sections 3.2 and 3.3 show that enhanced search durations and time-continuation allowed the MS version of the ε-NSGAII to rapidly and reliably improve search success rates. In the serial and both of the parallel versions of the ε-NSGAII, time continuation results from combining random solutions with ε-dominance archive members each time the population size is adapted. Each new population is equal to four times the current ε-dominance archive's size so that ε-non-dominated archive members compose 25% of the new population and the remaining 75% is composed of randomly generated solutions. Time continuation allows the algorithm to maintain effective search for as long as is necessary or is computationally tractable. The ε-NSGAII's failure rates for the Leaf River and LTM test cases shown in Figs. 8 and 11 result from time constraints and not from algorithmic limitations, which is often the case for water resources applications (e.g., see [87,88]).

In both the Leaf River and LTM test cases, linear to near linear speedups allowed the MS version of the ε-NSGAII to better exploit time-continuation, resulting in the exploration of new regions of their decision spaces much earlier in the allocated run times. The MP version of the ε-NSGAII has significantly more complex speedup dynamics relative to the MS version. For extremely difficult problems, the MP scheme's redundancy in search and its enhanced diversity from multiple independently searching populations dramatically improves its search capabilities relative to the single-population versions of the ε-NSGAII. Ironically, the strengths of the MP scheme that emerge for very difficult problems cause it to be inferior to the single population MS version of the ε-NSGAII for less difficult problems where extended search times will reliably yield high quality results. In other words, it takes much longer to converge multiple populations to Pareto optimal fronts and the MP scheme's complex speedup dynamics do not guarantee extended search periods.

A contribution of this research is to demonstrate that the ε-NSGAII's auto-adaptive population sizing, ε-dominance archiving, and time continuation when combined with a simple MS strategy can yield superior search relative to MP strategies. Additionally, as the solution evaluation times for water resources applications increases, the scalability of the MS version of the ε-NSGAII will improve (i.e., the asymptotic limits on MS speedups that result from communication costs decrease). Readers should also note the importance of monitoring solution quality using multiple EMO performance metrics, especially when performing speedup calculations. Overall, the MS version of the ε-NSGAII exhibits superior performance on both of the water resources applications, especially when considering its simplicity and ease-of-implementation relative to the MP scheme.

5. Conclusions

This study uses a formal metrics-based framework to demonstrate the MS and MP parallelization schemes for the ε-NSGAII. The MS and MP versions of the ε-NSGAII generalize the algorithm's auto-adaptive population sizing, ε-dominance archiving, and time continuation to a distributed processor environment. A key finding of this work is that time-continuation and parallel speedups can dramatically improve the efficiency and reliability of EMO algorithms in water resources applications. Time continuation is an evolutionary algorithm search enhancement that promotes solution diversity and allows the ε-NSGAII to maintain effective search for as long as is necessary or is computationally tractable. This study uses three test cases to compare the MS and MP versions of the ε-NSGAII: (1) an extremely difficult benchmark test function termed DTLZ6 drawn from the computer science literature, (2) an unconstrained continuous hydrologic model calibration test case for the Leaf River near Collins, Mississippi, and (3) a discrete, constrained, four-objective long-term groundwater monitoring (LTM) application. These test cases were carefully selected to encompass a broad range of problem properties and clearly demonstrate the importance of problem difficulty when selecting parallelization strategies for multiobjective applications.

The MP version of the ε-NSGAII is more effective than the MS scheme when solving DTLZ6. This study highlights that artificially constructed and extremely difficult test problems such as DTLZ6 may wrongly bias water resources applications towards using more complicated algorithms (i.e., the MP parallelization scheme), when a simpler MS strategy may work as well or better. Overall, the MS version of the ε-NSGAII exhibits superior performance on both of the water resources applications, especially when considering its simplicity and ease-of-implementation relative to the MP scheme. This study also clearly demonstrates the importance of monitoring solution quality using multiple EMO performance metrics, especially when performing speedup calculations. Overall, the three test cases used in this study demonstrate that the MS and MP versions of the ε-NSGAII can broaden the size and difficulty of multiobjective water resources applications that can be solved efficiently and reliably. Future extensions of this work should explore how heterogeneous computer networks impact the relative benefits of the MS and MP parallelization strategies.

Acknowledgements

This work was partially supported by the National Science Foundation under grant EAR-0310122. 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 and United State Department of Interior under grant USDI-01HQGR0099.

References

[1] Kollat JB, Reed PM. The value of online adaptive search: a performance comparison of NSGA-II, ϵ-NSGAII, and ϵMOEA. In: Coello CC, Aguirre AH, Zitzler E, editors. The third international conference on evolutionary multi-criterion optimization (EMO 2005). Lecture Notes in Computer Science, 3410. Guanajuato, Mexico: Springer Verlag; 2005. p. 386–98.

[2] Kollat JB, Reed PM. Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design. Adv Water Resour 2006;29(6):792–807.

[3] Tang Y, Reed P, Wagener T. How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration. Hydrol Earth Syst Sci 2006;10:289–307.

[4] Gropp W. Using MPI: portable parallel programming with message-passing interface. 2nd ed. Cambridge, MA: MIT Press; 1999.

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

[6] Coello Coello C, Van Veldhuizen DA, Lamont GB. Evolutionary algorithms for solving multi-objective problems. New York, NY: Kluwer Academic Publishers; 2002.

[7] Goldberg DE. The design of innovation: lessons from and for competent genetic algorithms. Norwell, MA: Kluwer Academic Publishers; 2002.

[8] Deb K. Scalable test problems for evolutionary multi-objective optimization. Technical report, KanGAL Report No. 2001001, Indian Institute of Technology, 2001.

[9] Duan Q, Gupta VK, Sorooshian S. Effective and efficient global optimization for conceptual rainfall-runoff models. Water Resour Res 1992;28(4):1015–31.

[10] Yapo PO, Gupta HV, Sorooshian S. Multi-objective global optimization for hydrologic models. J Hydrol 1998;204:83–97.

[11] Boyle D, Gupta H, Sorooshian S. Toward improved calibration of hydrologic models: combining the strengths of manual and automatic methods. Water Resour Res 2000;36:3663–74.

[12] Wagener T, Boyle D, Lees MJ, Wheater HS, Gupta HV, Sorooshian A. A framework for development and application of hydrological models. Hydrol Earth Syst Sci 2001;5(1):13–26.

[13] Vrugt J, Gupta HV, Bouten W, Sorooshian S. A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. Water Resour Res 2003;39(8):1201. doi:10.1029/2002WR001642.

[14] Vrugt J, Gupta HV, Bastidas LA, Bouten W, Sorooshian S. Effective and efficient algorithm for multiobjective optimization of hydrologic models. Water Resour Res 2003;39(8):1214. doi:10.1029/2002WR001746.

[15] Reed P, Ellsworth T, Minsker BS. Spatial interpolation methods for nonstationary plume data. Ground Water 2004;42(2):190–202.

[16] Kollat J, Reed P. A computational scaling analysis of multiobjective evolutionary algorithms in long-term groundwater monitoring applications. Adv Water Resour, in press, doi:10.1016/j.advwatres.2006.05.009.

[17] Chankong V, Haimes Y. Multiobjective decision making: theory and methodology. North-Holland series in system science and engineering. New York, NY: North-Holland; 1983.

[18] Haimes Y. Risk modeling, assessment, and management. Wiley series in systems engineering. New York, NY: John Wiley & Sons Inc.; 1998.

[19] Keeney RL, Raiffa H. Decisions with multiple objectives: preferences and value trade-offs. Cambridge, UK: Cambridge University Press; 1993.

[20] Farmani R, Savic DA, Walters GA. Evolutionary multi-objective optimization in water distribution network design. Eng Optim 2005;37(2):167–83.

[21] Ritzel BJ, Eheart JW, Ranjithan SR. Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. Water Resour Res 1994;30(5):1589–603.

[22] Horn J, Nafpliotis F. Multiobjective optimization using the Niched Pareto Genetic Algorithm. Technical report IlliGAL Report No. 93005, University of Illinois, 1993.

[23] Halhal D, Walters GA, Ouazar D, Savic DA. Water network rehabilitation with structured messy genetic algorithm. J Water Resour Plann Manage 1997;123(3):137–46.

[24] Loughlin DH, Ranjithan SR, Baugh Jr JW, Brill Jr ED. Application of genetic algorithms for the design of ozone control strategies. J Air Waste Manage Assoc 2000;50:1050–63.

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

[26] Erickson MA, Mayer A, Horn J. Multi-objective optimal design of groundwater remediation systems: application of the Niched Pareto Genetic Algorithm (NPGA). Adv Water Resour 2002;25(1):51–6.

[27] Reed P, Minsker BS, Goldberg DE. A multiobjective approach to cost effective long-term groundwater monitoring using an elitist nondominated sorted genetic algorithm with historical data. J Hydroinform 2001;3(2):71–90.

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

[29] Fleming PJ, Purshouse RC, Lygoe RJ. Many-objective optimization: an engineering design perspective. In: Coello Coello C, Hernandez A, Zitzler E, editors. Evolutionary multi-criterion optimization. Lecture notes in computer science, vol. 3410. Guanajuato, Mexico: Springer; 2005. p. 14–32.

[30] Farina M, Amato P. On the optimal solution definition for many-criteria optimization problems. In: Keller J, Nasraoui O, editors. Proceedings of the 2002 NAFIPS-FLINT international conference. New Jersey: IEEE Service Center, Piscataway; 2002. p. 233–8.

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

[32] Kumar SV, Ranjithan SR. Evaluation of the Constraint Method-based Multiobjective Evolutionary Algorithm (CMEA) for a three objective optimization problem. In: Langdon WB et al., editors. Proceedings of the genetic and evolutionary computation conference (GECCO 2002). New York, NY: Morgan Kaufmann; 2002. p. 431–8.

[33] Muleta M, Nicklow J. Decision support for watershed management using evolutionary algorithms. J Water Resour Plann Manage 2005;131(1):35–44.

[34] Farmani R, Walters GA, Savic DA. Trade-off between total cost and reliability for anytown water distribution network. J Water Resour Plann Manage 2005;131(3):161–71.

[35] Ren X, Minsker BS. Which groundwater remediation objective is better: a realistic one or a simple one? J Water Resour Plann Manage 2005;131(5):351–61.

[36] McPhee J, Yeh WG. Multiobjective optimization for sustainable groundwater management in semiarid regions. J Water Resour Plann Manage 2004;130(6):490–7.

[37] Labadie J. Optimal operation of multireservoir systems: state-of-the-art review. J Water Resour Plann Manage 2004;130(2):93–111.

[38] Gupta H, Sorooshian S, Yapo P. Toward improved calibration of hydrologic models: multiple and noncommensurable measures of information. Water Resour Res 1998;34:751–63.

[39] Madsen H. Automatic calibration of a conceptual rainfall-runoff model using multiple objectives. J Hydrol 2000;235:276–88.

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

[41] Madsen H. Parameter estimation in distributed hydrological catchment modeling using automatic calibration with multiple objectives. Adv Water Resour 2003;26:205–16.

[42] Seibert J. Multi-criteria calibration of a conceptual runoff model using a genetic algorithm. Hydrol Earth Syst Sci 2000;4(2):215–24.

[43] Ajami N, Gupta H, Wagener T, Sorooshian S. Calibration of a semi-distributed hydrologic model for streamflow estimation along a river system. J Hydrol 2004;298:112–35.

[44] Muleta M, Nicklow JW. Sensitivity and uncertainty analysis coupled with automatic calibration for a distributed watershed model. J Hydrol 2004:1–19.

[45] Vrugt J, Schoups G, Hopmans J, Young C, Wallender W, Harter T. Inverse modeling of large-scale spatially distributed vadose zone properties using global optimization. Water Resour Res 2004;40(6):W06503. doi:10.1029/2003WR002706.

[46] Reed P, Kollat JB, Devireddy V. Using interactive archives in evolutionary multiobjective optimization: case studies for long-term groundwater monitoring design. Environ Modell Software, in press.

[47] Knopman DS, Voss CI. Multiobjective sampling design for parameter estimation and model discrimination in groundwater solute transport. Water Resour Res 1989;25(10):2245–58.

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

[49] James BR, Gorelick SM. When enough is enough: the worth of monitoring data in aquifer remediation design. Water Resour Res 1994;30(12):3499–513.

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

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

[52] Storck P, Eheart JW, Valocchi AJ. A method for the optimal location of monitoring wells for detection of groundwater contamination in three-dimensional aquifers. Water Resour Res 1997;33(9):2081–8.

[53] Task Committee on Long-Term Groundwater Monitoring Design, Long-term groundwater monitoring: the state of the art, American Society of Civil Engineer, Reston, VA, 2003.

[54] Knopman DS, Voss CI, Garabedian SP. Sampling design for groundwater solute transport: tests methods and analysis of Cape Cod tracer test data. Water Resour Res 1991;27(5):925–49.

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

[56] Back T, Fogel D, Michalewicz Z. Handbook of evolutionary computation. Bristol, UK: IOP Publishing Ltd. and Oxford University Press; 2000.

[57] Salomon R. Evolutionary algorithms and gradient search: similarities and differences. IEEE Trans Evol Comput 1998;2(2):45–55.

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

[59] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6(2):182–97.

[60] Deb K, Mohan M, Mishra S. Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of pareto optimal solutions. Evol Comput 2005;13(4):501–26.

[61] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength pareto evolutionary algorithm, Technical report TIK-103, Department of Electrical Engineering, Swiss Federal Institute of Technology 2001.

[62] Deb K, Agrawal R. Simulated binary crossover for continuous search space. Complex Syst 1995;9:115–48.

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

[64] Reed P, Minsker BS, Goldberg DE. Simplifying multiobjective optimization: an automated design methodology for the Nondominated Sorted Genetic Algorithm-II. Water Resour Res 2003;39(7):1196. doi:10.1029/2002WR001483.

[65] Van Veldhuizen DA, Zydallis JB, Lambont GB. Considerations in engineering parallel multiobjective evolutionary algorithms. IEEE Trans Eolut Comput 2003;7(2):144–73.

[66] Deb K, Zope P, Jain A. Distributed computing of pareto-optimal solutions using multi-objective evolutionary algorithms, Technical report KanGAL Report No. 2002008, Indian Institute of Technology, 2002.

[67] Reed P, Yamaguchi S. Making it easier to use the multiple population parallelization scheme for evolutionary algorithms. In: Sehlke J et al., editors. 2004 EWRI world water and environmental resources congress. Salt Lake City, Utah: ASCE; 2004.

[68] Cantu-Paz E. Efficient and accurate parallel genetic algorithms. Norwell, MA: Kluwer Academic Publishers; 2000.

[69] Coello Coello C, Pulido GT. Multiobjective optimization using a micro-genetic algorithm. In: Spector L et al., editors. Proceedings of the genetic and evolutionary computation conference (GECCO 2001). San Francisco, CA: Morgan Kaufmann Publishers; 2001. p. 274–82.

[70] Eldredge N, Gould SJ. Punctuated equilibria: an alternative to phyletic gradualism. In: Schopf T, editor. Models of paleobiology. San Francisco, CA: Freeman, Cooper & Co.; 1972. p. 82–115.

[71] Eldredge N. Macro-evolutionary dynamics: species, niches, and adaptive peaks. New York, NY: McGraw-Hill; 1989.

[72] Dijkstra EW, Scholten CS. Termination detection for diffusing computations. Inf Process Lett 1980;11(1):1–4.

[73] Reed P, Yamaguchi S. Simplifying the parameterization of real-coded evolutionary algorithms. In: Sehlke G, editor. 2004 EWRI world water and environmental resources congress. Salt Lake City: ASCE; 2004.

[74] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable test problems for evolutionary multi-objective optimization. In: Proceedings of the congress on evolutionary computation (CEC-2002), 2002.

[75] Sorooshian S, Duan Q, Gupta HV. Calibration of rainfall-runoff models: application of global optimization to the sacramento soil moisture accounting model. Water Resour Res 1993;29(4):1185–94.

[76] Burnash RJC. The NWS river forecast system-catchment model. In: Singh VP, editor. Computer models of watershed hydrology. Highlands Ranch, CO: Water Resources Publications; 1995.

[77] Misirli F, Gupta HV, Sorooshian S, Thiemann M. Bayesian recursive estimation of parameter and output uncertainty for watershed models. In: Duan Q et al., editors. Calibration of watershed models, vol. 6. Washington, DC: American Geophysical Union; 2003. p. 113–24.

[78] Maxwell R, Carle FS, Tompson FB. Contamination, risk, and heterogeneity: on the effectiveness of aquifer remediation, Technical repot UCRL-JC-139664, Lawrence Livermore National Laboratory 2000.

[79] Deutsch CV, Journel AG. GSLIB: geostatistical software library and user's guide. New York, NY: Oxford University Press; 1998.

[80] Cooper RM, Istok JD. Geostatistics applied to groundwater contamination. I: Methodology. J Environ Eng 1988;114(2):270–86.

[81] Cooper RM, Istok JD. Geostatistics applied to groundwater contamination. II: Application. J Environ Eng 1988;114(2):287–99.

[82] Deb K, Jain S. Running performance metrics for evolutionary multi-objective optimization, Technical report 2002004, Indian Institute of Technology Kanpur, 2002.

[83] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evolut Comput 1999;3:257–71.

[84] Zitzler E, Thiele L, Laumanns M, Fonseca CM, Grunert da Fonseca V. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput 2003;7(3): 117–32.

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

[86] Conover W. Practical nonparametric statistics. 3rd ed. New York: -John Wiley & Sons; 1999.

[87] Pickup G. Testing the efficiency of algorithms and strategies for automatic calibration of rainfall-runoff models. Hydrol Sci Bull 1977;12(2):257–74.

[88] Isabel P, Villeneuve JV. Importance of convergence criterion in the automatic calibration of hydrological models. Water Resour Res 1986;22(10):1367–70.