Online Supplement for

Multi Objective Optimization of Computationally Expensive Multi-Modal Functions with RBF Surrogates and Multi-Rule Selection

Journal of Global Optimization

Taimoor Akhtar[1] and Christine A. Shoemaker[2]

[1] *School of Civil and Environmental Engineering, Cornell University, Ithaca, NY 14853, USA.*

e-mail: ta223@cornell.edu

[2] *School of Civil and Environmental Engineering, Cornell University, Ithaca, NY 14853, USA.*

e-mail: cas12@cornell.edu

A. Introduction

This document may be considered as an online supplement to the paper titled "Multi Objective Optimization of Computationally Expensive Multi-Modal Functions with RBF Surrogates and Multi-Rule Selection" (the paper will be referred as the main paper in subsequent discussions). The primary purpose of the supplementary material within this document is to provide an elaborate discussion on the use of 1) Alternative algorithms for embedded evolutionary optimization in Steps 2.2 and 2.3 of the GOMORS Algorithm Framework described in Section 3.1 of the main paper, and 2) Multiple Selection rules employed within the GOMORS algorithm for selection of points for expensive evaluations. Furthermore, this supplement also provides a description of the Test Problems used for comparison of GOMORS with ParEGO and NSGA-II, and a description of the algorithm parameters employed for GOMORS, ParEGO and NSGA-II.

Section C of this document provides a summary of computer experiments performed with different evolutionary algorithms for solving the embedded global optimization problems in Steps 2.2 and 2.3 of the Algorithm (defined in Section 3.1 of the main paper). Section D of this document provides a summary of the results obtained from computer experiments performed for analyzing the comparative advantage of using multiple selection rules for selection of points for expensive evaluation (especially, if they are used to select multiple points in a single algorithm iteration for evaluation in parallel). Section B of this document provides a description of the Synthetic Test Problems used for comparing algorithms, as well as an elaborate description of the Hypothetical Groundwater Remediation Problem used in the computer experiments. Section E provides the parameters used for all algorithms compared in this study, which include NSGA-II, ParEGO and GOMORS. Section F provides additional figures for comparative analysis of GOMORS and ParEGO.

B. Test Problems

B.1 Synthetic Test Suite

The performance of GOMORS was tested via computational experiments performed on eleven continuous test functions (collectively referred as the Synthetic Test Suite). Five of the test functions employed in the experiments are part of the ZDT test suite [16], while the remaining six are derived from the work of Li and Zhang [7]. The problems incorporated in the synthetic test suite are not expensive to evaluate. However, they incorporate various optimization challenges (for instance multi modality, non-convexity of the Pareto Front, complicated Pareto Sets etc) inherent in real world multi objective optimization problems.

Table B-1: Description of the ZDT Test Functions

Name	Mathematical Description	Problem Challenge
ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$ where $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$ $x \in [0,1]^n$ n may vary between 2 and 30	
ZDT2	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ where $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$ $x \in [0,1]^n$ n may vary between 2 and 30	The Pareto Front is non-convex.
ZDT3	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)} - \frac{f_1(x)}{g(x)} \sin(10\pi x_1)]$ where $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$ $x \in [0,1]^n$ n may vary between 2 and 30	The Pareto front is disconnected and has several noncontiguous convex parts.
ZDT4	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$ where $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}[x_i^2 - 10\cos(4\pi x_i)]$ $x_1 \in [0,1], x_i \in [-5,5], \forall i \in \{2,\dots,n\}$ n may vary between 2 and 30	21^9 locally optimal fronts exist.
ZDT6	$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ where $g(x) = 1 + 9[(\sum_{i=2}^{n} x_i)/(n-1)]^{0.25}$ $x \in [0,1]^n$ n may vary between 2 and 30	The search space is non-uniform, and distribution of solutions on the Pareto front is non-uniform.

Table B-2: Description of the LZF Test Functions

Name	Mathematical Description	Problem Challenge
LZF1	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}[x_j - \sin(6\pi x_1 + \frac{j\pi}{n})]^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}[x_j - \sin(6\pi x_1 + \frac{j\pi}{n})]^2$, where $J_1 = \{j \mid j$ is odd and $2 \le j \le n\}$, $J_2 = \{j \mid j$ is even and $2 \le j \le n\}$, and $x \in [0,1] \text{ x } [-1,1]^{n-1}$	The Pareto Set is a non-linear curve. Problem is also referred as F2 in [7].
LZF2	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} y_j^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} z_j^2$, where $J_1 = \{j \mid j$ is odd and $2 \le j \le n\}$, $J_2 = \{j \mid j$ is even and $2 \le j \le n\}$, $y_j = x_j - [0.3x_1^2\cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]\cos(6\pi x_1 + \frac{j\pi}{n})$, $z_j = x_j - [0.3x_1^2\cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]\sin 6\pi x_1 + \frac{j\pi}{n})$, and $x \in [0,1] \text{ x } [-1,1]^{n-1}$	The Pareto Set is a non-linear curve. Problem is also referred as F5 in [7].
LZF3	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}[4\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_1}\cos(\frac{20y_j\pi}{\sqrt{j}}) + 2]$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{\lvert J_1\rvert}[4\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_1}\cos(\frac{20y_j\pi}{\sqrt{j}}) + 2]$ where J_1 and J_2 are the same as in LZF1, $y_j = x_j - x_1^{0.5(1+\frac{3(j-2)}{n-2})}$, and $x \in [0,1]^n$	Many locally optimal fronts exist. Problem is also referred as F8 in [7].
LZF4 .	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} h(y_j)$ $f_2(x) = 1 - x_1^2 + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} h(y_j)$, where $J_1 = \{j \mid j$ is odd and $2 \le j \le n\}$, $J_2 = \{j \mid j$ is even and $2 \le j \le n\}$, $y_j = \sin(6\pi x_1 + \frac{j\pi}{n})$, $h(t) =\mid t \mid /(1 + 2e^{2\lvert t\rvert})$, and $x \in [0,1] \text{ x } [-2,2]^{n-1}$	Pareto Front is non-convex and Pareto Set is non-linear. The Problem is referred as UF4 in .
LZF5	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}[x_j - 0.8x_1\cos(6\pi x_1 + \frac{j\pi}{n})]^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}[x_j - 0.8x_1\sin(6\pi x_1 + \frac{j\pi}{n})]^2$ $J_1 = \{j \mid j$ is odd and $2 \le j \le n\}$ $J_2 = \{j \mid j$ is even and $2 \le j \le n\}$ $x \in [0,1] \text{ x } [-1,1]^{n-1}$	The Pareto Set is a non-linear curve. Problem is also referred as F3 in [7].
LZF6	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}[x_j - 0.8x_1\cos(\frac{6\pi x_1 + \frac{j\pi}{n}}{3})]^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}[x_j - 0.8x_1\sin(6\pi x_1 + \frac{j\pi}{n})]^2$ $J_1 = \{j \mid j$ is odd and $2 \le j \le n\}$ $J_2 = \{j \mid j$ is even and $2 \le j \le n\}$ $x \in [0,1] \text{ x } [-1,1]^{n-1}$	The Pareto Set is a non-linear curve. Problem is also referred as F4 in [7].

The ZDT test problems proposed by Zitzler et. al [16] include five continuous multi-objective optimization test problems all of which propose different challenges in converging to the Pareto optimal front. These challenges, along with mathematical descriptions of the ZDT optimization problems, are provided in Table B-1.The Pareto fronts for the five test problems are known. Meaningful comparisons of relative performance of different algorithms on these test problems can be based on how the quality of optimization solutions increases as the number of function evaluations increase, rather than on the actual running time of the algorithms (since purpose of comparison is evaluation of algorithm performance for computationally expensive problems).

Prior literature on the design of suitable test problems indicate that complicated shapes of Pareto Fronts add to the challenges of solving multi objective optimization problems via heuristics [7]. The ZDT test functions incorporate some of these challenges, including non-convex Pareto Fronts, disconnected Pareto fronts etc. However, the shapes of Pareto Sets for the ZDT functions are very simple. The Pareto Set of a problem refers to the set of Pareto optimal solutions of the problem in the decision space that corresponds to the Pareto Front. Pareto Sets for the ZDT problems are line segments. It is highly unlikely for real world problems to have such simple Pareto Sets.

Li and Zhang [7] introduced a general class of continuous multi objective optimization test problems where Pareto Sets could be defined arbitrarily. Furthermore, Li and Zhang [7] proposed nine specific test problems from the general class of problems introduced. We have incorporated five of these test in our synthetic test suite. We will refer these test problems as the LZF test functions for future discussion reference. The LZF test suite also includes a test problem which was proposed in the CEC 2009 MOEA competition [13]. This test function (LZF4) is also from the general class of optimization problems introduced by Li and Zhang. The LZF test problems propose different optimization challenges. These challenges, along with mathematical descriptions of the LZF optimization problems, are provided in Table B-2.

The ZDT and LZF functions will collectively be referred as the Synthetic Test Suite (eleven test problems) in future discussions. While Synthetic Test Suite identified for performance analysis addresses most of the optimization challenges discussed, some additional optimization challenges have been left unexplored, for instance, constraint handling, and testing on problems with more than two objectives.

B.2 Groundwater Remediation Design Problem

The test problems employed in our computer experiments also include a complex model describing the detoxification of contaminated groundwater using aerobic bioremediation [8]. Aerobic bioremediation is a decontamination process that involves the injection of electron acceptors (e.g. oxygen) or nutrients (e.g. nitrogen and phosphorus) into the groundwater to promote the growth of microorganisms that can transform the contaminant into a harmless substance. Cleaning up contaminated groundwater is financially expensive and the cost could run into tens of millions of dollars. Hence, the use of a complex model describing the decontamination process, coupled with an efficient global optimization algorithm could assist in devising cleanup strategies that may yield huge savings for environmental cleanup.

The groundwater remediation model used in our experiments considers a hypothetical contaminated aquifer with characteristics that are symmetric about a horizontal axis. We assume that there are 6 injection wells in the contaminated aquifer, the locations of which have been fixed. 84 monitoring wells are also assumed which measure the concentration of the contaminant at specified time periods. The aquifer is discretized using a two-dimensional finite element mesh. It is assumed that the injection wells and monitoring wells are located at the nodes of the mesh and these wells are also symmetric about the horizontal axis. A two-dimensional finite element simulation model called Bio2D [10], is used to describe groundwater flow and the changes in the concentrations of the contaminant, oxygen, and biomass. The model equations are nonlinear because the growth equations for the microorganisms represent Monod kinetics.

The optimization process considers the problem of deciding the pumping rates for injection wells, which are used to supply oxygen into the groundwater. Due to the symmetry of the contaminated area, we only need to make pumping decisions for the 3 injection wells on one side of the axis of symmetry. The planning time period is divided into multiple management periods, and pumping rates remain constant in each management period. We use 2, 4 and 8 management periods in our computer experiments which correspond to 6, 12 and 24 decision variables, respectively. Because the pumping of oxygenated water is expensive, our goal is to determine the pumping rates for each injection well at the beginning of each management period that minimize the total pumping cost and the contaminant concentration at the monitoring wells. Yoon and Shoemaker [12] incorporated the minimization of contaminant concentration, into the total pumping cost objective by means of a penalty term. This resulted in the formulation of the groundwater remediation design problem as a single objective global optimization problem.

We visualize the groundwater remediation design problem as a multi objective optimization problem where the conflicting objectives of the problem are 1) minimization of remediation cost and 2) minimization of contaminant concentration. The decision variables of the problem are the injection rates at 3 different well locations during each of m management periods. The number of decision variables vary between 6 and 24 (as discussed above), depending upon the number of management periods incorporated in the numerical computation model. The groundwater problem can be formulated as two separate multi-objective optimization problems. In the first formulation, the first objective is minimization of the cost of remediation, and the second objective is minimization of the maximum contaminant concentration recorded at the 84 monitoring wells at the end of the planning time. This formulation is referred to as the GWDM problem in the main paper and in this document. In the second formulation, the first objective is minimization of the cost of remediation and the second objective is minimization of the average contaminant concentration recorded at the 84 monitoring wells at the end of the planning time. This formulation is referred to as the GWDA problem in the main paper and in this document. Mathematical formulations of the two multi objective variants of the groundwater remediation design problem are provided in Table 2 of the main paper.

The simulation times for groundwater remediation problems can vary between minutes and hours depending on the complexity of the model and the size of the modeled region. This example uses a relatively coarse grid and requires only about 0.5 sec. to run each simulation on a 1.3 GHz Intel Core i5 machine. However, it is representative of the type of multi objective optimization formulations used in more complex and computationally expensive groundwater bioremediation problems.

C. Alternative Methods for Embedded Evolutionary Optimization

Section 3.1 of the main paper defines the general framework of the GOMORS algorithm where surrogate optimization problems defined in Steps 2.2 and 2.3b of the algorithm framework are solved within each algorithm iteration. The primary purpose of the surrogate optimizations is to find near optimal fronts given that the expensive functions, $F(x)$, are replaced by the corresponding response surface model approximations. Evaluation points are subsequently chosen from the solutions obtained via the surrogate optimizations (also referred as candidate solutions or candidate populations) during each algorithm iteration.

Step 2.2 performs a global surrogate assisted search and hence, solves a surrogate multi objective optimization problem over the whole decision space, while Step 2.3b performs a local surrogate assisted search and hence, solves a surrogate optimization problem within a localized neighborhood of the least crowded solution on the expensively evaluated non-dominated front (see Step 2.3a of the algorithm framework defined in Step 3.1 of the main paper). While solving them may be relatively inexpensive, the surrogate optimization problems are not trivial (since surrogates can be multi modal).

The optimization literature indicates that Multi Objective Evolutionary Algorithms (MOEAs) are a revolutionary heuristic approach for solving non trivial and multi modal multi objective optimization problems. Deb at al. [5] and Coello et al. [3] suggest that use of MOEAs can be highly beneficial, since the population based structure of an evolutionary algorithm can be exploited to simultaneously converge towards the Pareto front, and maintain a diverse set of trade-offs. Since the surrogate optimization problems defined in Steps 2.2 and 2.3 of the algorithm framework could potentially have non-linear and multi-modal objectives, and our aim is to find good approximate solutions of the surrogate problems (both in terms of convergence and diversity), we make use of evolutionary optimization for solving the surrogate problems within each iteration of GOMORS. In this regard, we performed computer experiments with either NSGA-II [4], MOEA/D [14], or AMALGAM [11] as evolutionary algorithms for solving the surrogate optimization problems within each iteration of the GOMORS algorithm framework. The purpose of these experiments was to identify one of these algorithm as most suited for surrogate optimization within GOMORS. Please note that surrogate optimization with reference to GOMORS is also referred as embedded evolutionary optimization in subsequent discussions. The evolutionary algorithms mentioned above are described briefly in sub sections C.1 - C.3. The experimental setup employed for comparison of performance of these algorithms within the GOMORS framework, and pertaining results are discussed in sub sections C.4 and

C.5.

C.1 Non Dominated Sorting Genetic Algorithm - II (NSGA-II)

NSGA-II, proposed by Deb et al. [4] is a revolutionary and extremely popular multi-objective (MOEA), and has been applied to various applied engineering problems across numerous disciplines [3] [2]. NSGA-II handles the evolutionary search optimization process by ranking and archiving parent and child populations according to a non-domination sorting (a measure of convergence), and crowding distance, which is a measure of diversity of a solution, on a particular front. The NSGA-II aims at using Pareto-dominance to move the the non-dominated front towards convergence, during the search process, and can handle highly non-linear and multi modal objectives.

C.2 Multi Objective Evolutionary Algorithm - Decomposition (MOEA/D)

Aggregate functions can also be used in the multi-objective optimization process in order to convert a multi-objective problem into many single- objective problems. The aggregate methodology falls under a separate class of MOEAs within Coello et al.'s [3] classification, where the aim is to solve many single objective optimization problems, with different aggregation priorities, to converge to the Pareto front, and maintain divergence. The MOEA/D [14] uses aggregate functions, and simultaneously solves many single-objective Tchebycheff decompositions (Tchebycheff decomposition is a preference based aggregation method for converting a vector of objectives into a single objective) of multi-objective problems in a single run.The MOEA/D is an established benchmark algorithm and has won the 2009 IEEE Congress on Evolutionary Computation (CEC 2009) competition.

C.3 AMALGAM - A Multi Algorithm Genetically Adaptive Multi-Objective Optimization Method

AMALGAM [11] is a multi-method evolutionary algorithm, which incorporates search mechanics of various algorithms. The key feature of AMALGAM is simultaneous multi-method search and the search mechanism selection is self-adaptive. AMALGAM incorporates four candidate MOEAs, NSGA-II, PSO, DE, and AMS. Recent literature has seen various applications of AMALGAM including Zhang et al.'s work [15], which includes application of AMALGAM for multi-site calibration, with comparison against SPEA2 and NSGA-II.

C.4 Experimental Setup

The relative performance of NSGA-II, MOEAD and AMALGAM as embedded MOEAs within the GOMORS algorithm framework was assessed on the synthetic test suite. Number of decision variables for all

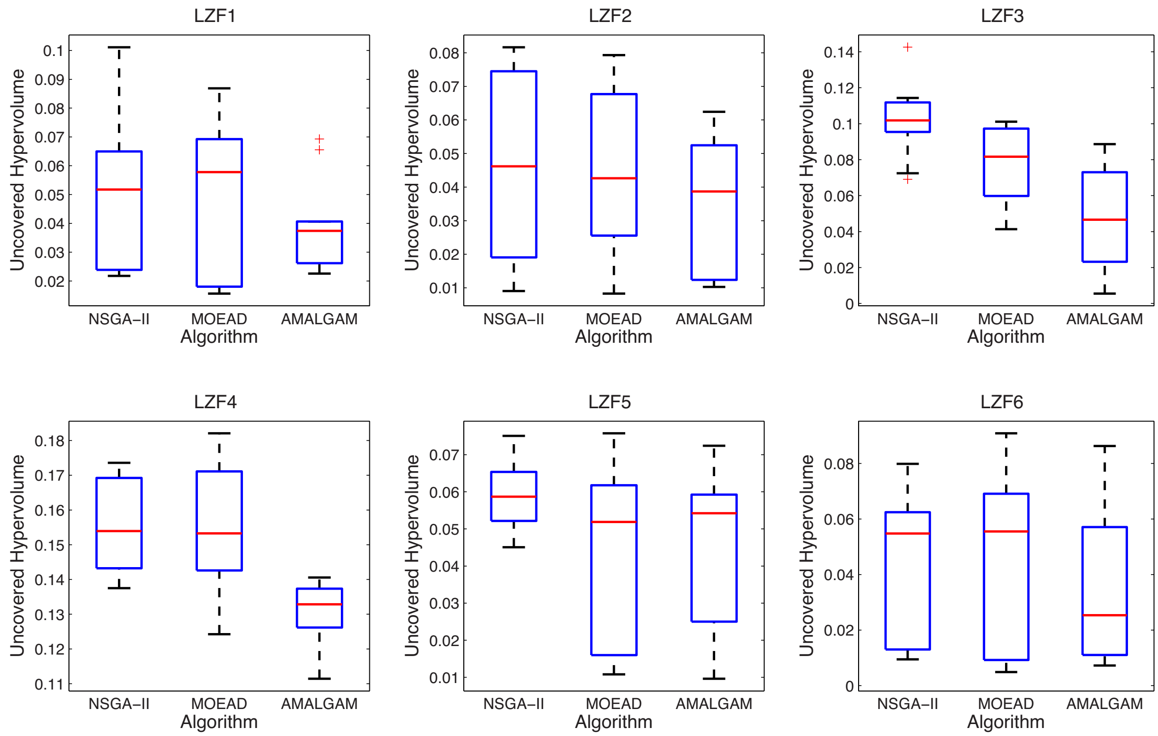

Figure C-1: Comparative analysis of evolutionary algorithms embedded into GOMORS - Box plots of uncovered hypervolume metric values obtained after 200 function evaluations with application of GOMORS using either of NSGA-II, MOEAD or AMALGAM as embedded evolutionary optimization algorithms . Results for the six LZF functions with 8 decision variables are shown in each subplot. In all cases lower values of uncovered hypervolume are best.

test problems were fixed at eight. All parameter settings of GOMORS were kept constant. The parameter settings of GOMORS used in the comparative analysis of the embedded MOEAs are defined in Table E-6 and Section E.3. The only change in GOMORS was the use of either of the three embedded MOEAs for solving the embedded optimization problems defined in Steps 2.2 and 2.3b of the algorithm framework (see Section 3.1 of the main paper).Ten optimization experiments were performed for each of the three modified versions of GOMORS, on each test problem and function evaluations were limited to 200. The same Latin hypercube samples were used for corresponding trials of each version of GOMORS, on each test problem.

The uncovered hypervolume metric [1] was used to compare the performance of the three versions of GOMORS with different MOEAs. Uncovered hypervolume is the difference between the total feasible objective space (illustrated in Figure 1(a) of the main paper) and the objective space dominated by estimate of the Pareto front obtained by an algorithm. A lower value of the index indicates a better solution and the ideal value is zero. The next section discusses the results obtained from the experiments with different MOEAs.

C.5 Results and Analysis

The uncovered hypervolume metric values obtained after performing optimization experiments on the LZF problems with application of GOMORS with the three alternative embedded MOEAs, is depicted via box plots in Figure C-1. Each of the six sub plots of the figure display the results pertaining to each of the six LZF problems. Lower values of the metric signify superiority of performance and a lower spread within a box plot depicts definiteness of performance.

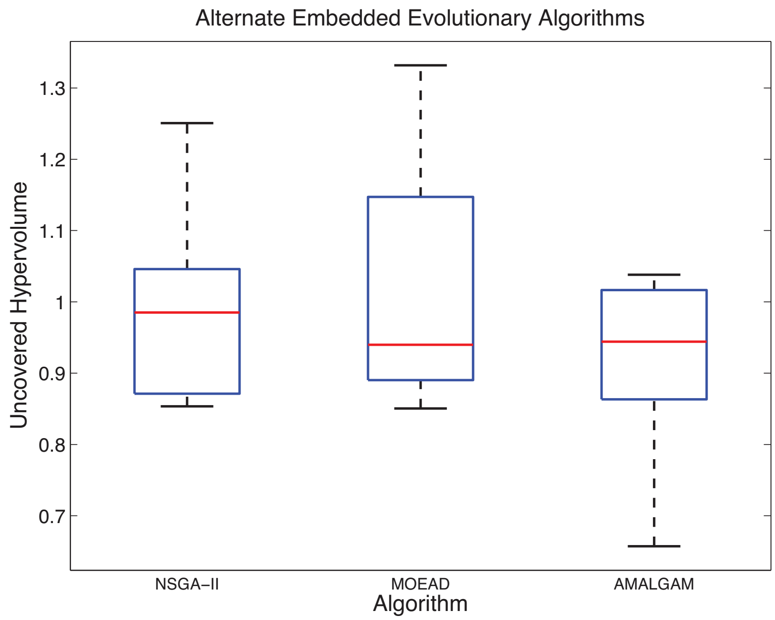

Figure C-2: Comparative analysis of evolutionary algorithms embedded into GOMORS - Box plots of uncovered hypervolume metric values summed over the eleven synthetic test functions, and obtained after 200 function evaluations with application of GOMORS using either of NSGA-II, MOEAD or AMALGAM as embedded evolutionary optimization algorithms. In all cases lower values of uncovered hypervolume are best.

The first conclusion that can be drawn from the results depicted in Figure C-1 is that it is hard to identify the difference between performance of all three embedded MOEAs. This is true especially in the case of LZF2, LZF5 and LZF6 (see Table B-2 and Figure C-1). However, in the case of LZF1, LZF3, and LZF4, it can be seen that the overall performance of AMALGAM (as embedded MOEA of GOMORS) is better than NSGA-II and MOEAD, both in terms of attainment of lower median metric values and lower spread in the box plots. After an overall analysis of the box plots of Figure C-1, it can concluded that performance of AMALGAM is slightly better than NSGA-II and MOEAD as an embedded MOEA for GOMORS.

Results of all synthetic test problems were also compiled for analysis by summing the metric values of each of the ten optimization experiments performed on each test problem. The uncovered hypervolume metric value obtained from every trial of each individual test problem and algorithm combination could be considered as a random sample from an unknown distribution. Assuming that the unknown distributions of each individual test problem and algorithm combination are independent across a single algorithm, a sum of metric values of each trial across a single algorithm is another random sample from an unknown distribution which is a convolution of the independent unknown distributions. This convolution based metric summarizes overall performance of an algorithm on all synthetic test problems and is depicted in Figure C-2 for comparison of performance of embedded MOEAs across all synthetic test functions.

Figure C-2 also highlights that performance of all embedded MOEAs is comparable with reference to their application to the eleven synthetic test functions (and considering that results are shown as a convolution across all test problems). However, Figure C-2 also indicates that performance of AMALGAM is slightly better than NSGA-II and MOEAD. The worst attained convoluted metric value from AMALGAM is significantly lower than the worst attained convoluted metric values of NSGA-II and AMALGAM. Based on the results depicted in Figures C-1 and C-2, we used AMALGAM as the embedded MOEA in GOMORS. However, it is worth mentioning that performances of the tested embedded MOEAs were close.

D. Experimental Assessment of Selection Rules

Section 3.4 of the main paper provides a discussion on the various selection rules employed within the GOMORS algorithm for selection of points for expensive evaluations within each algorithm iteration. Points for expensive evaluations are selected from the candidate points obtained after embedded optimizations of Steps 2.2 and 2.3 of the algorithm framework (defined in Section 3.1 of the main paper). The selected points are subsequently evaluated by the expensive objective functions, $F(x)$, within an algorithm iteration.

Selection of points for expensive evaluations is a critical step in the algorithm because evaluation of selected evaluations points is usually, by far, the most computationally expensive step of the algorithm, when applied to a computationally expensive problem. Furthermore, (as is discussed in Section 3.4 of the main paper) the strategy for selection of evaluation points is important for retraining of the surrogate functions in subsequent algorithm iterations, and can have a significant impact on the search dynamics of the algorithm. Major decisions pertaining to deduction of an appropriate selection strategy include the rule(s) for selection of evaluation point(s) and the number of points to be selected for expensive evaluation in each algorithm iteration.

Section 3.4 of the main paper proposes five selection rules which are employed to choose points for expensive evaluations in each iteration of GOMORS. In Section 3.4 of the main paper it is stated that GOMORS chooses one point for expensive evaluation in each algorithm iteration from four of the five rules stated (*Rules 1 to 4*), while one point for expensive evaluation is chosen from *Rule 0* with a probability of 0.1. Hence either 4 or 5 points are selected for expensive evaluations in each iteration of GOMORS. In order to deduce the above-mentioned strategy for selection of evaluation points, we performed various computer experiments with alternate evaluation point selection strategies. A total of six alternate selection strategies were compared in our analysis. The six selection strategies are discussed in Sections D.1 - D.6. The experimental setting for comparison of the selection strategies is discussed in Section D.7. Results obtained from the computer experiments are discussed in Section D.8.

D.1 Hypervolume Improvement

The first evaluation point selection strategy makes use of *Rule 1* (described on Section 3.4 of the main paper) to select one point for expensive evaluation in each iteration of GOMORS from the candidate solutions obtained from the surrogate optimization depicted in Step 2.2 of the algorithm framework (see Section 3.1 of the main paper for algorithm framework). *Rule 1* tends to choose a point from the candidate points obtained via global surrogate-assisted evolutionary optimization in Step 2.2, such that approximate improvement in hypervolume coverage is maximized. (see Figure 1 of the main paper for illustration of hypervolume improvement). This selection strategy, where one point is chosen for expensive evaluation via *Rule 1* in each iteration of GOMORS, is referred as the Hypervolume Improvement strategy in subsequent discussions. Please note that a second point for expensive evaluation may be selected in this selection strategy via random sampling (*Rule 0*), with a probability of 0.1.

D.2 Decision Space Euclidean Distance

The second point selection strategy analyzed in our computer experiments makes use of *Rule 2* (described in Section 3.4 of the main paper) to select one point for expensive evaluation in each iteration of GOMORS from the candidate solutions obtained from the surrogate optimization depicted in Step 2.2 of the algorithm framework (see Section 3.1 of the main paper for algorithm framework). *Rule 2* selects a point from the candidate points obtained via global surrogate-assisted evolutionary optimization in Step 2.2, such that the minimum euclidean distance (with respect to the decision space) from already evaluated points is maximized. This selection strategy, where one point is chosen for expensive evaluation via *Rule 2*, is referred to as the Decision Space Euclidean Distance strategy in subsequent discussions. Please note that a second point for expensive evaluation may be selected in this selection strategy via random sampling (*Rule 0*), with a probability of 0.1.

D.3 Objective Space Euclidean Distance

The third evaluation point selection strategy analyzed in our computer experiments makes use of *Rule 3* (described in Section 3.4 of the main paper) to select one point for expensive evaluation in each iteration of GOMORS from the candidate solutions obtained from the surrogate optimization depicted in Step 2.2 of the algorithm framework (see Section 3.1 of the main paper for an overview of the algorithm framework). *Rule 3* selects a point from the candidate points obtained via global surrogate-assisted evolutionary optimization in Step 2.2, such that the minimum euclidean distance (with respect to the objective function space) from already evaluated points is maximized. This selection strategy, where one point is chosen for expensive evaluation via *Rule 3*, is referred as the Objective Space Euclidean Distance strategy in subsequent discussions. Note that a second point for expensive evaluation may be selected in this selection strategy via random sampling (*Rule 0*), with a probability of 0.1.

D.4 Local Surrogate Assisted Search

The fourth evaluation point selection strategy analyzed in our computer experiments for comparing different selection strategies makes use of *Rule 4* (described in Section 3.4 of the main paper) to select one point for expensive evaluation in each iteration of GOMORS from the candidate solutions obtained from the surrogate optimization depicted in Step 2.3b of the algorithm framework (see Section 3.1 of the main paper for an overview of the algorithm framework). Evaluation point selection based on *Rule 4* is similar to *Rule 1*. However an evaluation point is selected from the candidate points obtained via local surrogate-assisted (or embedded) evolutionary optimization depicted in Step 2.3b. This selection strategy, where one point is chosen for expensive evaluation via *Rule 4* in each iteration of GOMORS, is referred to as the Local Surrogate Assisted Search strategy in subsequent discussions. Note that a second point for expensive evaluation may be selected in this selection strategy via random sampling (*Rule 0*), with a probability of 0.1.

D.5 Rule Cycling

The fifth evaluation point selection strategy selects one point for expensive evaluation in each iteration of GOMORS by cycling between the four point selection rules mentioned in the previous sections, i.e, hypervolume improvement, maximizing minimum euclidean distance in the decision space from previously evaluated points, maximizing minimum objective space difference from previously evaluated points, and surrogate-assisted local search. This selection strategy, where only a single point is chosen for expensive evaluation via cycling between selection rules, is referred as the rule cycling selection strategy in subsequent discussions. Please note that a second point for expensive evaluation may be selected in this selection strategy via random sampling (*Rule 0*), with a probability of 0.1.

D.6 Parallel Selection

The sixth and final evaluation point selection strategy analyzed in our computer experiments selects one point each from Rules 1 to 4 (described on Section 3.4 of the main paper) for simultaneous expensive evaluations during each iteration of GOMORS. The primary purpose of investigating this strategy is to target an improvement in efficiency of GOMORS through parallel evaluation of expensive objectives in each algorithm iteration. This selection strategy, where four points are selected for expensive evaluation via Rules 1 to 4, is referred as the parallel selection strategy in subsequent discussions. Please note that a fifth point for expensive evaluation may be selected in this selection strategy during each algorithm iteration via random sampling (*Rule 0*), with a probability of 0.1.

D.7 Experimental Setup

The relative performance of the alternate evaluation point selection strategies of GOMORS was assessed on the synthetic test suite. Number of decision variables for all test problems were fixed at eight. All other parameter settings of the algorithm, for instance the choice of embedded MOEA, and related parameters, were the same in all experiments. The parameter settings are provided in Table E-6 and Section E.3. The only change in the algorithm was the use of the alternate evaluation point selection strategies.

Ten optimization experiments were performed for each strategy, on each test problem and the same Latin hypercube samples were used for corresponding trials of each strategy (in an effort to ensure that the analysis is as fair as possible). The function evaluations were limited to 200 for all strategies. However, for the parallel selection strategy, we also performed experiments with a limit of 400 evaluations, in order to analyze the advantage, if any, of simultaneous selection and evaluation of expensive objectives in each algorithm iteration, in terms of speed of convergence of the algorithm.

Table D-3: The alphanumeric codes assigned to each selection strategy for Figures 3 and 4

Strategy Name	**Code**
Approximate Hypervolume Improvement	HYP
Decision Space Euclidean Distance	D-DS
Objective Space Euclidean Distance	D-OS
Surrogate-assisted Local Search	LS
Rule Cycling	CYC
Parallel Selection with 200 expensive evaluations	P200
Parallel Selection with 400 expensive evaluations	P400

The uncovered hypervolume metric [1] was used to compare the performance of the alternate selection strategies. As discussed earlier, Uncovered hypervolume is the difference between the total feasible objective space (illustrated in Figure 1(a) of the main paper) and the objective space dominated by estimate of the Pareto front obtained by an algorithm. A lower value of the index indicates a better solution and the ideal value is zero. Figures D-3 and D-4 summarize the results obtained from the experiments. These are discussed further in the next section. Please note that alphanumeric codes are used to identify the different selection strategies in Figures D-3 and D-4. Table D-3 provides the selection strategy names corresponding to the alphanumeric codes used in Figures D-3 and D-4. We refer the alphanumeric during our discussion or the results in the next section.

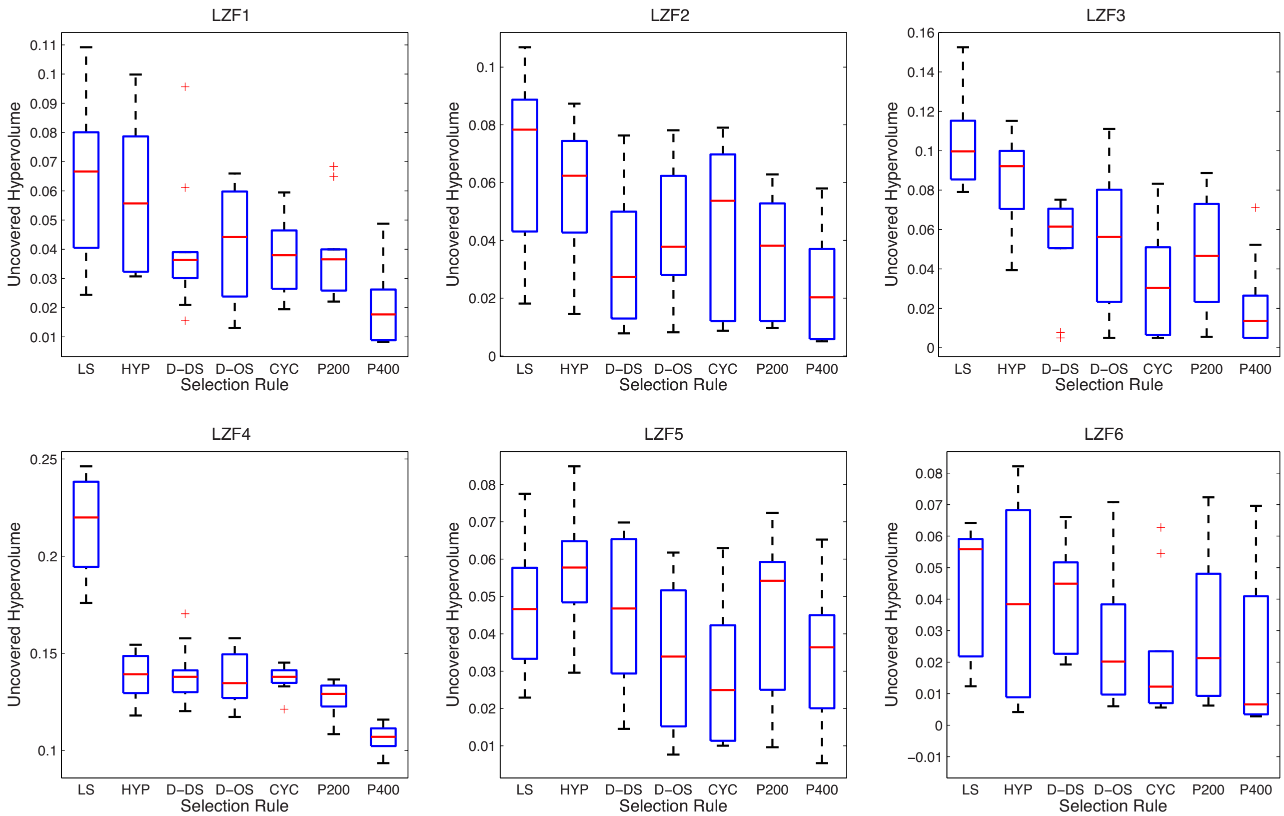

Figure D-3: Comparative analysis of selection strategies for LZF functions - Box plots of uncovered hypervolume metric values obtained after 200 function evaluations (400 in case of P400) through application of GOMORS with alternate evaluation point selection strategies. Results for the six LZF functions with 8 decision variables are shown in each subplot. In all cases lower values of uncovered hypervolume are best.

D.8 Results and Discussion

The purpose of this section is to evaluate the performance of GOMORS with the different evaluation point selection strategies discussed in Sections D.1 to D.6. This is done to analyze the difference in performance of GOMORS when only one of four selection rules is employed for evaluation point selection, when cycling between the rules is employed for evaluation point selection and when all rules are employed simultaneously in each algorithm iteration to select multiple points for expensive evaluations. The purpose of the analysis is to understand the advantage (if any) of using multiple rules to select multiple points for simultaneous evaluations in each algorithm iteration.

The uncovered hypervolume metric values obtained after performing optimization experiments on the LZF problems with application of GOMORS with the alternate selection strategies, are depicted via box plots in Figure D-3. The six sub plots in the figure display the results pertaining to each of the six LZF problems. Lower values of the metric signify superiority of performance and a lower spread within a box plot depicts definiteness of performance.

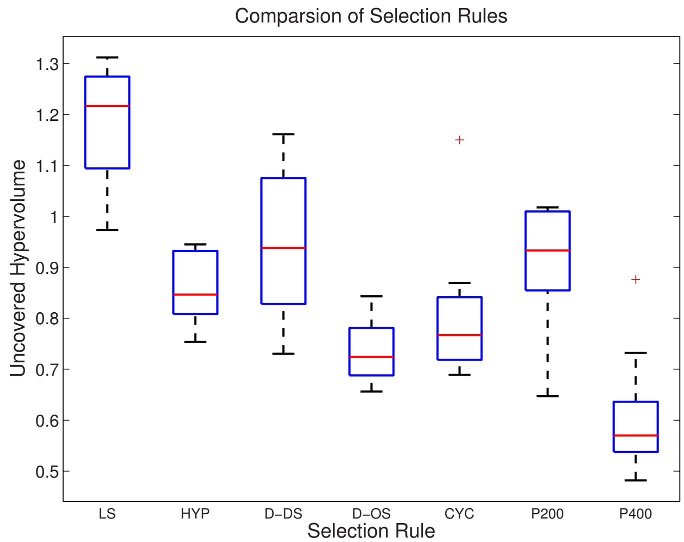

Figure D-4: Comparative analysis of election strategies - Box plots of uncovered hypervolume metric values summed over the eleven synthetic test functions, and obtained after 200 function evaluations (400 in case of P400) through application of GOMORS with alternate evaluation point selection strategies. In all cases lower values of uncovered hypervolume are best.

We first analyze the performance of the first four selection strategies, where a selection rules is exclusively used to select one point for expensive evaluation in each algorithm iteration. An interesting observation in this regard is that none of these four selection strategies (i.e, LS, HYP, D-DS and D-OS) clearly outperforms the others for all LZF test functions. However, it can be observed that performance of LS and HYP is relatively poor when considering the results on all LZF functions. The performance of LS is relatively poor for LZF1, LZF2, LZF3 and LZF4, and performance of HYP is relatively poor for LZF1, LZF2 , LZF3 and LZF5. The relatively poor performance of LS, where only a surrogate-assisted local search is performed to select an evaluation point in each algorithm iteration, is understandable. The relatively poor performance of HYP may be a consequence of errors in RBF approximation due to the existence of complicated Pareto sets in the LZF functions. Performance of D-DS and D-OS is comparable, and relatively better than the LS and HYP selection strategies. Since D-DS and D-OS select evaluation points based on maximization of minimum euclidean distance from already evaluated points (in the decision space and objective space respectively), they tend to maintain a balance between exploration and exploitation during the surrogate-assisted search. This might be the reason behind their relatively superior performance on the LZF functions where the Pareto sets are formed by complicated curves in the decision space.

The rule cycling (CYC) selection strategy also performs adequately on the LZF functions. The most interesting finding though is the performance of the Parallel selection strategy (P200 and P400) where multiple points (one point each from Rules 1 to 4) are simultaneously selected for expensive evaluations in each algorithm iteration. Hence, four points are selected for simultaneous evaluations (a fifth point may be selected via random sampling with probability of 0.1). The primary motivation behind this strategy is to enable the GOMORS to evaluate expensive function in parallel during each algorithm iteration in order to improve performance in terms of speed of convergence.

Results for the Parallel selection strategy with a limit of 400 evaluations (P400) are particularly interesting in this regard. Results of the P400 selection strategy depicted in Figure D-3 indicate that performance of the P400 strategy is either comparable or better than the other selection strategies. If the value of parallelization is considered, the P400 selection strategy produces comparable results to the other selection strategies with approximately 100 effective expensive function evaluations since four points may be evaluated in parallel during each algorithm iteration (Our approximation of 100 effective evaluations is made with the assumption that the communication time overhead is inconsequential in comparison to the computational cost of an expensive function evaluation).

Results of all synthetic test problems (including the ZDT functions) were also compiled for analysis by summing the metric values of each of the ten optimization experiments performed on each test problem. The resulting box-plots are depicted in Figure D-4, and are consistent with the observations deduced from Figure D-3. Figure D-4 indicates that results from the parallel selection strategy after 400 function evaluations (P400) are considerably better than the other selection strategies after 200 function evaluations. If the value of parallelization is considered, Figure D-4 implies that the parallel selection strategy with 100 effective function evaluations (since 4 points may be evaluated in parallel during each algorithm iteration) outperforms the other selection strategies with 200 function evaluations.

Hence, in the main paper (Section 3.4 of the main paper) we recommend the use of a Multi-Rule selection strategy where selection rules 1 to 4 are employed to select four points for simultaneous evaluations in each iteration of GOMORS (A fifth point may be selected via Rule 0 with a probability of 0.1). If a serial implementation of GOMORS is employed, we recommend the use of either CYC or D-OS as the selection strategy, based on the results obtained from experiments performed on the synthetic test suite (see Figure D-4).

E. Parameter Settings

In order to evaluate and compare the performance of GOMORS, ParEGO and NSGA-II on the synthetic test suite and the groundwater remediation problems, ten optimization experiments were performed for each algorithm, on each test problem. As mentioned in Section 5.1 of the main paper, multiple experiments were performed on each test problem, since all algorithms are stochastic. Furthermore, the evaluation budget was limited to four hundred, since the purpose of the experiments was to evaluate algorithm performance within a limited evaluation budget. Parameters for the algorithms were set such that the comparison be as fair as possible.

Parameter settings for each algorithm are discussed in sections E.1 - E.3. The initial Latin hypercube sample sizes for GOMORS and ParEGO were fixed at $2d + 2$, where d is the number of decision variables. Since both GOMORS and ParEGO use Latin hypercube sampling for initiation, the fairest way to compare them is to have trial j for both algorithms have the same Latin hypercube for $j = 1, \ldots, N$ (when a particular problem is tested). Hence, the same Latin hypercube samples were used for corresponding trials of GOMORS and ParEGO on each test problem.

E.1 NSGA-II

We used a MATLAB implementation of the NSGA-II, coded by Aravind Seshadri. This implementation is available on the MATLAB Central File Exchange and is based on the algorithm description provided by Deb. et. al. [4]. Default settings of the algorithm parameters were used with the exception of the population size and the number of generations. We experimented with population sizes of 10, 20, 50 and 100, on the test functions, with an evaluation budget limit of four hundred. Results from the experiments indicated that a population size of 20 was most feasible in terms of algorithm performance. Hence, a population size of 20 was used and the number of generations deduced were 19. Table E-4 summarized the parameter settings used for NSGA-II.

Table E-4: Parameter Settings for NSGA-II

Parameter Name	Setting
Population Size	20
Maximum Generations	19
Crossover Probability	0.9
Real-value SBX parameter	20
Polynomial mutation parameter	20

E.2 ParEGO

The implementation code for ParEGO was obtained from the research web page of Joshua Knowles (first author of [6]). Default settings of the algorithm parameters, as prescribed in [6] were used with the exception of the number of initial Latin hypercube samples. Knowles [6] recommend $11d - 1$ as the number of initial Latin hypercube samples. However since our computer experiments incorporate problems with up to 24 decision variables with an evaluation budget of 400, an initial sample size of $11d - 1$ would result in a large initial sample relative to the evaluation budget. Consequently, we used an initial Latin hypercube sample size of $2(d + 2)$, as per the recommendations provided by Regis and Shoemaker [9]. Table E-5 summarized the parameter settings used for ParEGO (Please note that k is the number of objective functions).

Table E-5: Parameter Settings for ParEGO

Parameter Name	**Setting**
No. of initial Latin hypercube samples	$2(d+2)$
No. of maximum evaluations	400
No. of scalarizing vectors	$11(k=2)$
Scalarizing function type	Augmented Tchebycheff
Internal GA population size	20
Internal GA generations	10000
Crossover Probability	0.9
Real-value SBX parameter	10
Real-value mutation probability	$1/d$
Polynomial mutation parameter	50

E.3 GOMORS

GOMORS was implemented in MATLAB version 2012b. For the embedded optimization, MATLAB implementation codes for NSGA-II, MOEA/D and AMALGAM were obtained externally. The implementation for NSGA-II, coded by Aravind Seshadri, is available on the MATLAB Central File Exchange and is based on the algorithm description provided by Deb. et. al. [4]. The implementation for MOEA/D was obtained from Qingfu Zhang's research web page (Qingfu Zhang is the first author of MOEA/D [14]). The implementation code for AMALGAM was obtained from Jasper Vrugt through an email request (Jasper Vrugt is the first author of AMALGAM [11]). Default settings of the embedded algorithms were used in the analysis discussed in Section C. Subsequent to the analysis AMALGAM was highlighted as an appropriate embedded evolutionary algorithm for use within GOMORS. Default settings of AMALGAM were used within the embedded optimizations of Steps 2.2 and 2.3b of GOMORS (please refer to Section 3.1 of the main paper for reviewing the algorithm framework). Table E-6 summarized the parameter settings used for GOMORS.

Table E-6: Parameter Settings for GOMORS

Parameter Name	Setting
No. of initial Latin hypercube samples	$2(d+2)$
No. of maximum evaluations	400
Local Search / Gap radius	0.1
Random selection probability	0.1
Embedded global MOEA population size	100
Embedded global MOEA generations	25
Embedded local MOEA population size	100
Embedded local MOEA population size	25

Additionally, it should be noted that one point each is selected for expensive evaluation from rules no. 1 to 4, during each algorithm iteration (all selection rules are described in Section 3.4 of the main paper), in the final implementation of GOMORS. Furthermore, a point is selected via rule no. 0, during each algorithm iteration, with a probability of 0.1 (Please note that rule 0 is selection through random sampling). Hence, in this application of GOMORS only 4 or 5 points are selected for parallel evaluation in each algorithm iteration. However, more points could be selected from each rule during an algorithm iteration and more rules could be incorporated in the algorithm in future, resulting in an increase in the number of simultaneous evaluations in each algorithm iteration. The parallel properties of GOMORS are discussed briefly in Section D of this online supplement. The MATLAB implementation of GOMORS will be available on the research webpage of Christine Shoemaker (if paper is accepted for publication).

F. Additional Results

Additional results pertaining to Non-Dominated Front visualizations for the two groundwater problems are provided in this section. There are two figures (F-5 and F-6) corresponding to the worst solutions (as per uncovered hypervolume) obtained from multiple experiments performed via GOMORS and ParEGO on the GWDA and GWDM groundwater problems. The red line within each sub-plot of the figures is an estimate of the Pareto front. This estimate was obtained (for both GWDM and GWDA) through a single trial of NSGA-II with 50000 function evaluations. The green dots within each sub-plot correspond to the non-dominated solutions obtained via application of the algorithm referenced in the sub-plot. There are six sub-figures within each figure and two sub-plots within each sub-figure, depicting the difference in performance of GOMORS and ParEGO. Each sub-figure corresponds to the worst non-dominated fronts obtained from each algorithm for a different number of decision variables (6 and 24) and after a fixed number of function evaluations (100, 200 and 400). A bottom to top traversal of each figure provides a visualization of the change in performance of algorithms as function evaluations increase, and a left to right traversal assists in visualizing performance differences when decision variables increase.

Figures F-5 and F-6 are consistent with the conclusions drawn in Section 5 of the main paper, and depict that performance of GOMORS is superior to ParEGO even in the worst case scenario. GOMORS is able to produce diverse trade-offs in the worst case scenario for the 24 variable problems, while ParEGO fails to obtain diverse solutions. This notion is true for both GWDA and GWDM.

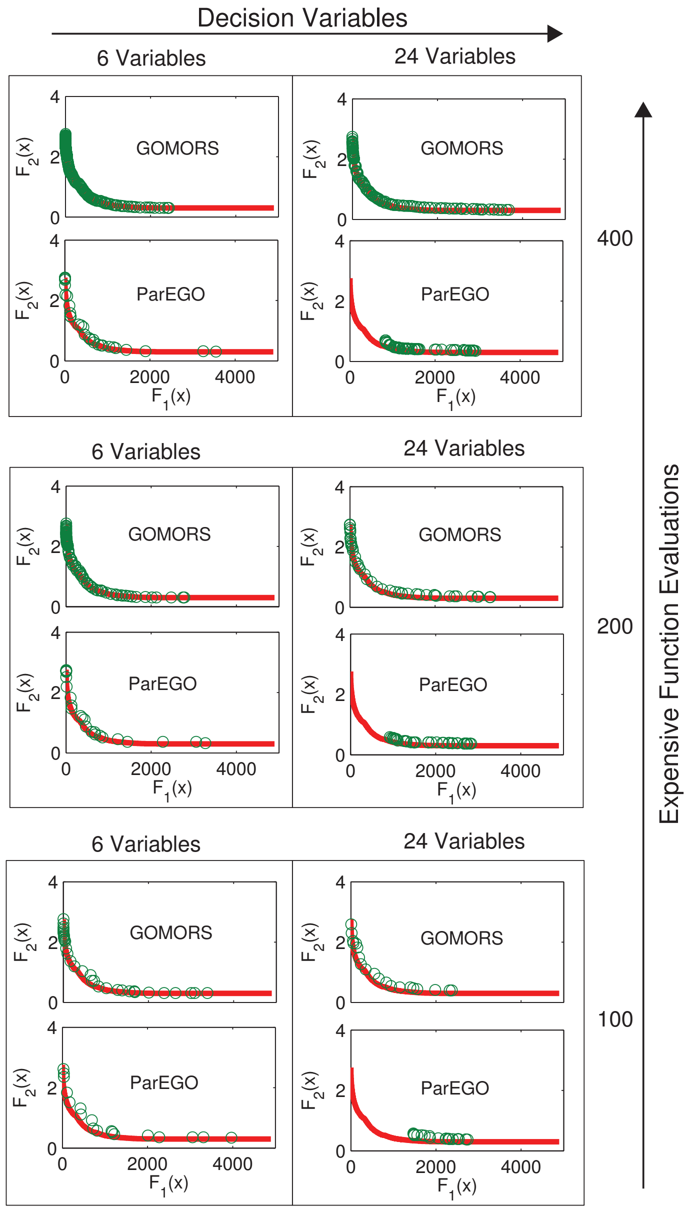

Figure F-5: Estimated Non-dominated Front for Groundwater Problem GWDM: Visual comparison of worst non-dominated fronts obtained from GOMORS and ParEGO for GWDM with 6 and 24 decisions, and after 100, 200 and 400 expensive function evaluations. Red line is result of 50,000 evaluations with NSGA-II. Green circles are non-dominated solutions from GOMORS or ParEGO.

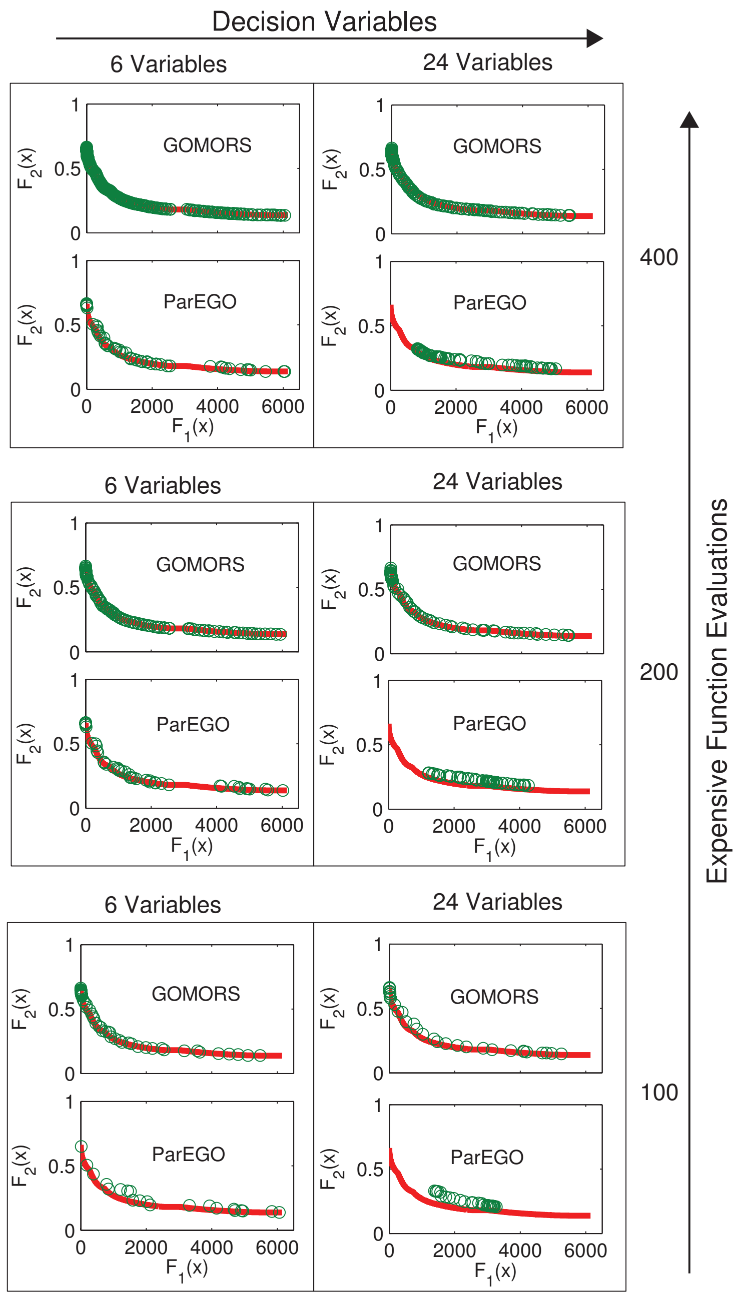

Figure F-6: Estimated Non-dominated Front for Groundwater Problem GWDA: Visual comparison of worst non-dominated fronts obtained from GOMORS and ParEGO for GWDM with 6 and 24 decisions, and after 100, 200 and 400 expensive function evaluations. Red line is result of 50,000 evaluations with NSGA-II. Green circles are non-dominated solutions from GOMORS or ParEGO.

References

[1] Johannes M. Bader. *Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods*. CreateSpace, Paramount, CA, 2010.

[2] E Bekele and J Nicklow. Multi-objective automatic calibration of SWAT using NSGA-II. *Journal of Hydrology*, 341(3-4):165–176, August 2007.

[3] C.A. Coello Coello, G.L. Lamont, and D.A. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer, Berlin, Heidelberg, 2nd edition, 2007.

[4] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, PPSN VI, pages 849–858, London, UK, UK, 2000. Springer-Verlag.

[5] Kalyanmoy Deb and Deb Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 1 edition, June 2001.

[6] J. Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 8(5):1341–66, February 2006.

[7] Hui Li and Qingfu Zhang. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, April 2009.

[8] Barbara S. Minsker and Christine A. Shoemaker. Differentiating a finite element biodegradation simulation model for optimal control. *Water Resources Research*, 32(1):187–192, 1996.

[9] R Regis, C Shoemaker, and A. stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. of Computing*, 19:497–509, 2007.

[10] S. W. Taylor. Modeling enhanced in-situ biodegradation in groundwater: Model response to biological parameter uncertainty. In *1993 Groundwater Modeling Conference*, 1993.

[11] Jasper a Vrugt and Bruce a Robinson. Improved evolutionary optimization from genetically adaptive multimethod search. *Proceedings of the National Academy of Sciences of the United States of America*, 104(3):708–11, January 2007.

[12] J.H. Yoon, , and Christine A. Shoemaker. Comparison of optimization methods for ground-water bioremediation. *Journal of Water Resources Planning and Management*, 125:54–63, 1999.

[13] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari. Multiobjective optimization test instances for the cec 2009 special session and competition. Technical Report CES-487, University of Essex and Nanyang Technological University, 2008.

[14] Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[15] Xuesong Zhang, Raghavan Srinivasan, and Michael Van Liew. On the use of multi-algorithm, genetically adaptive multi-objective method for multi-site calibration of the SWAT model. *Hydrological Processes*, 24(8):955–969, April 2010.

[16] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8:173–195, 2000.