

# Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection

Taimoor Akhtar · Christine A. Shoemaker

Received: 31 October 2013 / Accepted: 15 January 2015 / Published online: 8 February 2015  
© The Author(s) 2015. This article is published with open access at Springerlink.com

**Abstract** GOMORS is a parallel response surface-assisted evolutionary algorithm approach to multi-objective optimization that is designed to obtain good non-dominated solutions to black box problems with relatively few objective function evaluations. GOMORS uses Radial Basic Functions to iteratively compute surrogate response surfaces as an approximation of the computationally expensive objective function. A multi objective search utilizing evolution, local search, multi method search and non-dominated sorting is done on the surrogate radial basis function surface because it is inexpensive to compute. A balance between exploration, exploitation and diversification is obtained through a novel procedure that simultaneously selects evaluation points within an algorithm iteration through different metrics including Approximate Hypervolume Improvement, Maximizing minimum domain distance, Maximizing minimum objective space distance, and surrogate-assisted local search, which can be computed in parallel. The results are compared to ParEGO (a kriging surrogate method solving many weighted single objective optimizations) and the widely used NSGA-II. The results indicate that GOMORS outperforms ParEGO and NSGA-II on problems tested. For example, on a groundwater PDE problem, GOMORS outperforms ParEGO with 100, 200 and 400 evaluations for a 6 dimensional problem, a 12 dimensional problem and a 24 dimensional problem. For a fixed number of evaluations, the differences in performance between GOMORS and ParEGO become larger as the number of dimensions increase. As the number of evaluations increase, the differences between GOMORS and ParEGO become smaller. Both surrogate-based methods are much better than NSGA-II for all cases considered.

---

**Electronic supplementary material** The online version of this article (doi:[10.1007/s10898-015-0270-y](https://doi.org/10.1007/s10898-015-0270-y)) contains supplementary material, which is available to authorized users.

---

T. Akhtar (✉) · C. A. Shoemaker  
School of Civil and Environmental Engineering and School of Operations Research and Information Engineering, Cornell University,  
Holister Hall, Ithaca, NY, USA  
e-mail: ta223@cornell.edu

C. A. Shoemaker  
e-mail: cas12@cornell.edu

**Keywords** Function approximation · Multi objective optimization · Radial basis functions · Expensive optimization · Global optimization · Evolutionary optimization · Parallel · Metamodel

## 1 Introduction

Multi-objective optimization (MO) approaches involve a large number of function evaluations, which make it difficult to use MO in simulation–optimization problems where the optimization is multi-objective and the nonlinear simulation is computationally expensive and has multiple local minima (multi modal). Many applied engineering optimization problems involve multiple objectives and the computational cost of evaluating each objective is high (e.g minutes to days per objective function evaluation by simulation) [12, 32]. We focus on special algorithms that aim to produce reasonably good results within a limited number of expensive objective function evaluations.

Many authors (for instance, Deb et al. [5] and Zhang et al. [52]) have successfully employed evolutionary strategies for solving multi-objective optimization problems. Even with the improvement over traditional methods, these algorithms require, typically, many objective function evaluations which can be infeasible for computationally expensive problems. Added challenges to multi-objective optimization of expensive functions arise with increase in dimensionality of the decision variables and objectives.

The use of iterative response surface modeling or function approximation techniques inside an optimization algorithm can be highly beneficial in reducing time for computing objectives for multi-objective optimization of such problems. Since the aim of efficient multi-objective optimization is to identify good solutions within a limited number of expensive function evaluations, approximating techniques can be incorporated into the optimization process to reduce computational costs. Gutmann [15] introduced the idea of using radial basis functions (RBF) [3] for addressing single objective optimization of computationally expensive problems. Jin et al. [18] appears to be the first journal paper to combine a non quadratic response surface with a single objective evolutionary algorithm by using neural net approximations. Regis and Shoemaker [36] were the first to use Radial Basis Functions (not a neural net) to improve the efficiency of an evolutionary algorithm with limited numbers of evaluations. Later they introduced a non-evolutionary algorithm Stochastic-RBF [37], which is a very effective radial basis function-based method for single objective optimization of expensive global optimization problems. These methods have been extended to include parallelism [40]; high-dimensional problems [42]; constraints [38]; local optimization [49, 50]; integer problems [30, 31] and other extensions [39, 41]. Kriging-based methods have also been explored for addressing single objective optimization problems [9, 16, 19]. Jones et al. [19] introduced Efficient Global Optimization (EGO), which is an algorithm for single objective global optimization within a limited budget of evaluations.

Response surface methods have also become popular for multi-objective optimization problems, with kriging-surrogate techniques being the most popular. Various authors have used kriging-based methods by extending the EGO framework for multi-objective optimization of computationally expensive problems. For instance, Knowles [21] combined EGO with Tchebycheff weighting to convert an MO problem into numerous single objective problems. Tsang et al. [53] and Emmerich et al. [10] combined EGO with evolutionary search assistance. Ponweiser et al. [33] and Beume et al. [2] explored the idea of maximizing expected improvement in hypervolume. Authors have also explored the use of other function approximation techniques inside an optimization algorithm, including Radial Basis Functions (RBFs)

[13, 20, 25, 45, 51], Support Vector Machines [24, 43] and Artificial Neural Networks [7]. Evolutionary algorithms are the dominant optimization algorithms used in these methods. Some papers also highlight the effectiveness of local search in improving performance of response surface based methods [20, 23, 45].

Various authors have indicated that RBFs could be more effective than other approximation methods in multi-objective optimization of computationally expensive problems with more than 15 decision variables [8, 29, 43]. Various authors have employed RBFs for surrogate-assisted multi-objective optimization of expensive functions with focus on problems with more than 15 decision variables. For instance, Karakasis and Giannakoglou [20] employ RBFs within an MOEA framework and use an inexact pre-evaluation phase (IPE) to select a subset of solutions for expensive evaluations in each generation of an MOEA. They also recommend the use of locally fitted RBFs for surrogate approximation. Georgopoulou and Giannakoglou [13] build upon [20] by employing gradient based refinements of promising solutions highlighted by RBF approximation during each MOEA generation. Santana et al. [45] divide the heuristic search into two phases where the first phase employs a global surrogate-assisted search within an MOEA framework, and rough set theory is used in the second phase for local refinements.

This paper focuses on the use of RBFs and evolutionary algorithms for multi-objective optimization of computationally expensive problems, where the number of function evaluations are limited relative to the problem dimension (e.g. to a few hundred evaluations for the example problems tested here). An added purpose of the investigation is to be able to solve MO problems where the number of decision variables varies between 15 and 25.

To this effect we propose a new algorithm, GOMORS, that combines radial basis function approximation with multi-objective evolutionary optimization, within the general iterative framework of surrogate-assisted heuristic search algorithms. Our approach is different from prevalent RBF based MO algorithms that use evolutionary algorithms [13, 20, 25, 45, 51]. Most RBF based evolutionary algorithms employ surrogates in an inexact pre-evaluation phase (IPE) in order to inexpensively evaluate child populations after every MOEA generation. By contrast our proposed methodology employs evolutionary optimization within each iteration of the algorithm framework to identify numerous potential points for expensive evaluations. Hence, multiple MOEA generations evolve via surrogate-assisted search in each algorithm iteration in GOMORS.

The novelty of the optimization approach is in the amalgamation of various evaluation point selection rules in order to ensure that various value functions are incorporated in selecting some points for expensive evaluations from the potential points identified by surrogate-assisted evolutionary search during each algorithm iteration. The combination of multiple selection rules targets a balanced selection between exploration, exploitation and front diversification. The selection strategy incorporates the power of local search and also ensures that the algorithm can be used in a parallel setting to further improve its efficiency.

## 2 Problem description

Let  $\mathcal{D}$  (domain space) be a unit hypercube and a subset of  $\mathbb{R}^d$  and  $x$  be a variable in the domain space, i.e.,  $x \in [0, 1]^d$ . Let the number of objectives in the multi-objective problem equal  $k$  and let  $f_i(x)$  be the  $i$ th objective.  $f_i$  is a function of  $x$  and  $f_i : \mathcal{D} \mapsto \mathbb{R}$  for  $1 \leq i \leq k$ . The framework of the multi-objective optimization problem we aim to solve is as follows:

$$\begin{aligned} &\text{minimize } F(x) = [f_1(x), \dots, f_k(x)]^T \\ &\text{subject to } x \in [0, 1]^d \end{aligned} \tag{1}$$

Our goal for the multi-objective optimization problem is (within a limited number of objective function evaluations) to find a set of Pareto-optimal solutions  $P^* = \{x_i^* \mid x_i^* \in \mathcal{D}, 1 \leq i \leq n\}$ .  $P^*$  is defined via the following definitions:

**Domination** A solution  $x_1 \in \mathcal{D}$  dominates another solution  $x_2 \in \mathcal{D}$  if and only if  $f_i(x_1) \leq f_i(x_2)$  for all  $1 \leq i \leq k$ , and  $f_i(x_1) < f_i(x_2)$  for at least one  $i \in \{1, \dots, k\}$ .

**Non-domination** Given a set of solutions  $S = \{x_i \mid x_i \in \mathcal{D}, 1 \leq i \leq n\}$ , a solution  $x^* \in S$  is non-dominated in  $S$  if there does not exist a solution  $x_j \in S$  which dominates  $x^*$ .

**Pareto optimality** A candidate solution  $x^* \in S$  which is non-dominated in  $S$  is called a globally Pareto-optimal solution if  $S = \mathcal{D}$ , i.e.,  $S$  is the entire feasible domain space of the defined problem.

### 3 Algorithm description

#### 3.1 General framework

The proposed optimization algorithm is called Gap Optimized Multi-objective Optimization using Response Surfaces (GOMORS). GOMORS employs the generic iterative setting of surrogate-assisted algorithms in which the response surface model used to approximate the costly objective functions is updated after each iteration. Multiple points are selected for expensive function evaluation in each iteration, evaluated in parallel, and subsequently used to update the response surface model. The algorithm framework is defined below:

---

##### Step 0 - Define Algorithm Inputs:

$M$  - Maximum number of expensive function evaluations

$r$  - The Gap radius parameter used in Step 2.3

$t$  - Number of expensive evaluations to be performed after each algorithm iteration

**Step 1 - Initial Evaluation Points Selection:** Select an initial set of points  $\{x_1, \dots, x_m\}$ , where  $x_i \in \mathcal{D}$ , for  $1 \leq i \leq m$ , using an experimental design. Evaluate the objectives  $F = [f_1, \dots, f_k]$  at the selected  $m$  points, via expensive simulations. Let  $S_m = \{x_i \mid x_i \in \mathcal{D}, i = 1, \dots, m\}$  denote the initial set of evaluated points. Let  $P_m = \{x_i \in S_m \mid x_i \text{ is non-dominated in } S_m\}$  be the set of non-dominated points from  $S_m$ .

**Step 2 - Iterative Improvement:** Run algorithm iteratively until termination condition is satisfied:

**While**  $m \leq M$

**Step 2.1 - Fit/Update Response Surface Models:** Fit/update response surface models for each objective based on the set of already evaluated simulation points  $S_m$ . Let  $\widehat{F}_m(x) = [\widehat{f}_{m,1}, \dots, \widehat{f}_{m,k}]$  be the inexpensive approximate objective functions obtained by fitting a response surface model on  $S_m$ .

**Step 2.2 - Surrogate Assisted Global Search:** Use a multi-objective evolutionary algorithm (MOEA) to solve the following optimization problem:

$$\text{minimize: } \widehat{F}_m(x) = [\widehat{f}_{m,1}(x), \dots, \widehat{f}_{m,k}(x)]^T \tag{2}$$

Let  $\widehat{P}_m^A = \{x_1, \dots, x_{n_A}\}$  denote the solutions obtained by solving Equation 2, i.e, utilizing an MOEA for searching on the response surfaces.

**Step 2.3a - Identify Least Crowded Solution:** Using the crowding distance calculation procedure proposed by Deb et al. [5], identify the least crowded element of the expensively evaluated non-dominated set,  $P_m$ . Let  $x^{crowd}$  be the least crowded element of  $P_m$  (see definition in Table 1).

**Step 2.3b - Local Search:** Use a multi-objective evolutionary algorithm in a small neighborhood of  $x^{crowd}$  to solve the following optimization problem:

$$\begin{aligned} \text{minimize: } & \widehat{F}_m(x) = \{\widehat{f}_{m,1}(x), \dots, \widehat{f}_{m,k}(x)\} \\ \text{subject to: } & (x^{crowd} - r) \leq x \leq (x^{crowd} + r) \end{aligned} \tag{3}$$

The problem solved in Equation 3 we will call the “Gap Optimization Problem”.

Let  $\widehat{P}_m^B = \{x_1, \dots, x_{n_B}\}$  denote the solutions obtained by solving Equation (3), i.e, utilizing an MOEA for local search on the response surfaces, within a radius,  $r$ , around the least crowded solution,  $x^{crowd}$ .

**Step 2.4 - Select Points for Expensive Function Evaluations:** Our evaluation points are then selected from the sets  $\widehat{P}_m^A$  and  $\widehat{P}_m^B$  (from Equations 2 and 3, and also called candidate points) based on the rules described in Sect. 3.4. Let  $S_{curr} = x_1, \dots, x_t$  be the set of  $t$  points selected for expensive evaluations in current algorithm iteration. (Discussed in detail in Sect. 3.4)

**Step 2.5 - Do expensive function evaluations and update Non-dominated solution set:** Evaluate costly functions,  $F$ , on the selected evaluation points,  $S_{curr}$ . Update  $S_m$ , i.e., add the new expensively evaluated points to the set of already evaluated points. Consequently,  $m = m + t$ ,  $S_m = \{S_m\} \cup \{S_{curr}\}$ . Update  $P_m$ , i.e., compute  $P_m = \{x_i \in S_m \mid x_i \text{ is non-dominated in } S_m\}$ .

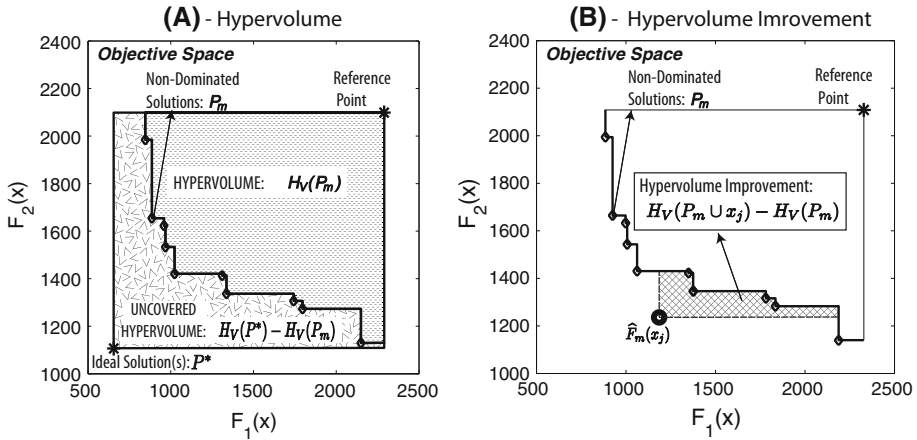
**End Loop**

**Step 3 - Return Best Approximated Front:** Return  $P_M = \{x_i \in S_M \mid x_i \text{ is non-dominated in } S_M\}$  as an approximation to the globally optimal solution set, i.e,  $P^* = \{x_i^* \mid x_i^* \text{ is non-dominated in } \mathcal{D}\}$ .

The algorithm initiates with selection of an initial set of points for costly function evaluations of the objective function set,  $F$ . Latin hypercube sampling, a space-filling experimental design scheme proposed by McKay et al. [26] is used for selection of the initial evaluation set. The iterative framework of GOMORS (Step 2) follows, and consists of three major segments: (1) Building a Response Surface Model for approximating expensive objectives (Step 2.1), (2) Applying an Evolutionary algorithm for solving multi-objective response surface problems (Steps 2.2 and 2.3) and (3) Selecting multiple points for expensive evaluations from the solutions of the multi-objective response surface problems, and evaluating them in parallel (Step 2.4). The algorithm terminates after  $M$  expensive evaluations and the output is a non-dominated set of solutions  $P_M$ , which is an approximation to the true Pareto solution of the problem i.e  $P^*$ . Sections 3.2–3.4 and the online supplement discuss details of the major steps of the algorithm.

**Table 1** Definitions of sets and variables

Item	Description
$F(x) = [f_1(x), \dots, f_k(x)]$	Expensive objectives for the multiple-objective optimization problem
$P^* = \{x_1^*, \dots, x_n^*\}$	Set of Pareto-optimal solutions given the objectives $F$
$S_m = \{x_1, \dots, x_m\}$	Set of points which have been evaluated via costly simulation until iteration $m$ of the optimization algorithm
$\widehat{F}_m(x) = [\widehat{f}_{m,1}(x), \dots, \widehat{f}_{m,k}(x)]$	$\widehat{f}_{m,i}$ is the inexpensive surrogate approximation of the $i^{th}$ objective, $f_i$ , generated using the set of points, $S_m$
$P_m = \{x \in S_m\}$	The <i>non-dominated</i> solutions in $S_m$ based on expensive evaluations
$x^{crowd} = \arg \max_{x \in P_m} [d(x)]$	The least crowded element in $P_m$ , according to the crowding definition, $d(\cdot)$ , where $d(x_j)$ indicates the function by [6] to compute crowding distance for $x_j$ given $(x_j, F(x_j))$ for all $x_j \in P_m$
$\widehat{P}_m^A = \{x_1, \dots, x_{n_A}\}$	Candidate points obtained by <i>Surrogate-Assisted Global Search</i> on the approximate objective function set $\widehat{F}_m$ (see Equation 2)
$\widehat{P}_m^B = \{x_1, \dots, x_{n_B}\}$	Candidate points obtained by <i>Gap Optimization</i> of the approximate objectives, $\widehat{F}_m$ (see Equation 3)
$H_V(P)$	Hypervolume [1] of the objective space dominated by the objective vectors corresponding to the set $P$ . The hyperspace is such that $y \in \mathbb{R}^k$ and $y$ is bounded by a reference vector $b$ (see Fig. 1 for illustration)



**Fig. 1** Visualization of **a** hypervolume and uncovered hypervolume and **b** hypervolume improvement employed in Rule 1 and Rule 4 of Step 2.4 (discussed in Sect. 3.4). Maximization of hypervolume improvement implies selection of a point for function evaluation that predicts maximum improvement in hypervolume coverage of a non-dominated set as per RBF approximation

### 3.2 Response surface modeling

The first major component of the iterative framework is the procedure used for approximating the costly functions,  $F$ , by the inexpensive surrogates,  $\widehat{F}_m(x) = [\widehat{f}_{m,1}, \dots, \widehat{f}_{m,k}]$ . A response surface model based on the evaluated points is generated in Step 2.1 of the algorithm and subsequently updated after each iteration. For example artificial neural networks [35], Support Vector Machines (SVM) [47], kriging [44], and radial basis functions (RBFs) [3,34] could be employed to generate the response surface model.

While kriging has been used in MO [21,33,53], the number of parameters to be estimated for kriging meta-models increases quickly with an increase in the number of decision variables [16,17], and hence, re-evaluating kriging surrogates in each iteration of a kriging based algorithm may itself become a computationally expensive step. Various authors [29,42] including Manriquez et al. [8] have reported that kriging-based surrogate optimization is not effective for high dimensional problems (approximately defined as problems with more than 15 decision variables). In [8] they demonstrate the relative effectiveness of RBF approximation in tackling such high dimensional problems. The GOMORS algorithm proposed in this paper hence makes use of RBFs as the surrogate model for approximating the costly functions, although other surrogate models could be used in the GOMORS strategy.

### 3.3 Evolutionary optimization

The surrogate optimization problems defined in Steps 2.2 (*Surrogate-assisted global search*) and 2.3 (*Gap Optimization*) aim to find near optimal fronts given that the expensive functions,  $F(x)$ , are replaced by the corresponding response surface model approximations, i.e.,  $\widehat{F}_m(x)$  (see Table 1). In Steps 2.2 and 2.3, two different multi-objective optimization problems are solved on the surrogate surface.

Steps 2.2 and 2.3 solve the MO problems depicted in Equations 2 and 3 respectively. Since the objective functions of these problems are derived from the surrogate model, solving the MO problems is a relatively inexpensive step. However, the MO problems of Equations 2 and

3 are not trivial and could potentially have non-linear and multi-modal objectives. Hence, we employ evolutionary algorithms for MO optimization of surrogates in Steps 2.2 and 2.3.

Three algorithms were considered as potential alternatives for optimization of surrogates in Steps 2.2 and 2.3, namely, NSGA-II, [5], MOEA/D [52] and AMALGAM [48]. NSGA-II handles the evolutionary search optimization process by ranking and archiving parent and child populations according to a non-domination sorting. MOEA/D [52] uses aggregate functions, and simultaneously solves many single-objective Tchebycheff decompositions of multi-objective problems in an evolutionary generation. AMALGAM [48] is a multi-method evolutionary algorithm, which incorporates search mechanics of various algorithms. Extensive computer experiments on test problems were performed on GOMORS with either of NSGA-II, MOEA/D and AMALGAM as embedded evolutionary schemes (see Section C of the online supplement for a detailed discussion on the experiments) and AMALGAM was identified as the best performing evolutionary algorithm (although the differences were small) embedded in GOMORS.

### 3.4 Expensive evaluation point selection: Step 2.4

Step 2.4 of the GOMORS algorithm determines which of the candidate points are evaluated by the expensive objective functions,  $F(x)$ , within an algorithm iteration. The candidate points are obtained from Steps 2.2 and 2.3, and are denoted as  $\hat{P}_m^A$  and  $\hat{P}_m^B$  respectively. As mentioned earlier, selection of points for expensive evaluation is a critical step in the algorithm because this is usually by far the most computationally expensive step.

A balance between *exploration* [19], *exploitation* and *diversification* [6] is crucial for selecting points for expensive evaluations from candidate points,  $\hat{P}_m^A$  and  $\hat{P}_m^B$ . *Exploration* of the decision space aims at selecting points in unexplored regions of the decision space. *Exploitation* aims at exploiting the inexpensive response surface approximations of Step 2.1 to assist in choosing appropriate points for expensive evaluations. *Diversification* strives to ensure that the non-dominated evaluated points are spread out in the objective space.

GOMORS employs a multi-rule based strategy for selection of a small subset of candidate points ( $\hat{P}_m^A$  and  $\hat{P}_m^B$ ) for actual function evaluations (computing  $F$  via simulation). The various “rules” employed in the strategy target a balance between *exploration*, *exploitation* and *diversification*. A detailed framework of Step 2.4 of the algorithm is given below which gives an overview of the selection rules (The  $i^{th}$  rule is referred as Rule  $i$ ) (Refer Table 1 for definitions of sets and symbols):

#### Details of Step 2.4 (in Section 3.1):

**Inputs from previous steps:**  $(m, S_m, P_m, \hat{P}_m^A, \hat{P}_m^B)$

**Evaluation Points selection:** Select  $t = \sum_{i=0}^4 t_i$  points for expensive evaluations via rules 0–4. Let  $S_{curr}$  be the selected points. So there are  $t_i$  points generated using Rule  $i$ , where the Rules are listed below.

**Apply Rule 0 - Random Sampling:** Pick  $t_0$  points for expensive function evaluation via random sampling from a uniform distribution. Ensures exploration of decision space.

**Apply Rule 1- Hypervolume Improvement:** Use hypervolume improvement to choose  $t_1$  points from the candidate points  $\hat{P}_m^A$ . Let  $H_V(P_m)$  denote the hypervolume [1] of the objective space dominated by the vectors in the set  $P_m$  (see Figure 1a). A new point for expensive function evaluation is selected as follows (see Table 1 for definition of sets and variables):

$$x^* = \arg \max_{x_j \in \hat{P}_m^A} \left[ H_V(P_m \cup x_j) - H_V(P_m) \right] \tag{4}$$

Equation 4 exploits the RBF approximation and chooses a point for function evaluation which maximizes the improvement in hypervolume in the objective space as per the RBF approximations of the candidate points, as illustrated in Fig. 1b.

**Apply Rule 2- Maximize Minimum Domain Euclidean Distance:** Let  $x_{i,A} \in P_m^A$  and  $x_j \in S_m$ . Choose  $t_2$  points for expensive evaluation such that the maximum of minimum distances of each point from already evaluated points is maximized, i.e.:

$$x^* = \arg \max_{x_{i,A} \in \widehat{P}_m^A} \left[ \min_{x_j \in S_m} \|x_{i,A} - x_j\| \right] \tag{5}$$

**Apply Rule 3- Maximize Minimum Objective Space Euclidean Distance:** Choose  $t_3$  points such that the maximum of minimum distances of each point from already evaluated points, as per the objective function space, is maximized, i.e.:

$$x^* = \arg \max_{x_{i,A} \in \widehat{P}_m^A} \left[ \min_{x_j \in S_m} \|\widehat{F}_m(x_{i,A}) - F(x_j)\| \right] \tag{6}$$

Rule 2 and Rule 3 are a hybrid of exploration and exploitation.

**Apply Rule 4 - Hypervolume Improvement in “Gap Optimization” candidates:** Use “Rule 1” to select  $t_4$  points from the candidate points obtained via “Gap Optimization”, i.e.  $\widehat{P}_m^B$ . Select points for expensive function evaluations as follows (See Table 1 for definition of sets and variables):

$$x^* = \arg \max_{x_j \in \widehat{P}_m^B} \left[ H_V(P_m \cup x_j) - H_V(P_m) \right] \tag{7}$$

The difference between (4) and (7) is that  $\widehat{P}_m^A$  in (4) comes from Step 2.2 and  $\widehat{P}_m^B$  in (7) comes from Step 2.3.

The number of points to be selected for expensive evaluation via each rule i.e.  $t_i$  may vary. However, we performed all our computer experiments with  $t_i = 1$ , for Rules 1–4. A point is selected via Rule 0 in each algorithm iteration with a probability of 0.1. Hence, either four or five points are selected for expensive function evaluations in each iteration of the algorithm. The expensive evaluations of points are performed in parallel to further speed up the algorithm.

In order to assess the individual effectiveness of the rules, we performed computer experiments on eleven test problems with the exclusive use of all rules (i.e, one point is selected from one rule, in all algorithm iterations). Furthermore, we also performed experiments with the idea of cycling between all rules in subsequent iterations of GOMORS framework. These different selection strategies were compared against the multiple rule selection strategy for simultaneous selection of evaluation points (described above). Results of the computer experiments indicated that if the value of parallelization is considered, the multiple rule selection strategy outperforms the other strategies employed on our analysis (See Section D of the online supplement for details.). However, if GOMORS is used in a serial setting, i.e, one point is evaluated in each algorithm iteration, cycling between rules, and use of Rule 3 are most beneficial (see section D of online supplement).

### 4 Test problems

In order to test the performance of GOMORS, computational experiments were performed on various test functions and a groundwater remediation problem. Certain characteristics of various problems can lead to inadequate convergence or poor distribution of points on the Pareto front. These characteristics include high dimensionality, non-convexity and non-uniformity of the Pareto front [6], the existence of multiple locally optimal Pareto



**Table 2** Two MO variants, GWDA and GWDM, of the groundwater optimization problem (Sect. 4.1)

GWDA	GWDM
minimize: $f_1(X) = \sum_{t=1}^T C_t(X_t)$	minimize: $f_1(X) = \sum_{t=1}^T C_t(X_t)$
minimize: $f_2(X) = \frac{\sum_{m=1}^M s_{m,T}(X)}{M}$	minimize: $f_2(X) = \max_{m=1}^M s_{m,T}(X)$
subject to: $X \in [0, 1]^{N \times T}$	subject to: $X \in [0, 1]^{N \times T}$

$M$  is total number of finite element grid nodes,  $N$  is the total number of wells,  $T$  is the total number of management periods,  $X_t$  is the pumping decision in management period  $t$ ,  $s_{m,t}$  is the contaminant concentration at grid  $m$  and at the end of the last time period  $T$ , and  $C_t(X_t)$  is the cost of cleanup for the pumping decision  $X_t$

fronts [4], low density of solutions close to the Pareto front, and existence of complicated Pareto sets (this implies that the Pareto solutions are defined by a complicated curve in the decision space) [22]. We have employed eleven test problems in our experimental analysis which incorporate the optimization challenges mentioned above. Five test problems are part of the ZDT test suite [4], while six were derived from the work done by Li and Zhang [22]. Mathematical formulations and optimization challenges of the test problems are discussed in detail in Section B.1 of the online supplement to this document. These problems are collectively referred as synthetic test problems in subsequent discussions.

#### 4.1 Groundwater remediation design problem

Optimization problems pertaining to groundwater remediation models usually require solving complex and computationally expensive PDE systems to find objective function values for a particular input [27]. The groundwater remediation problem used in our analysis is based on a PDE system which describes the movement and purification of contaminated groundwater given a set of bioremediation and pumping design decisions [28]. Detailed description of the problem is provided in Section B.2 of the online supplement. The decision variables of the problem are the injection rates of remediation agent at 3 different well locations during each of  $m$  management periods. The input dimension size ranges between 6 and 36 variables, depending upon the number of management periods incorporated in the numerical computation model.

The remediation optimization problem can be formulated as two separate multi-objective optimization problems, called GWDA and GWDM (Table 2). In the first formulation, GWDM, the aim is to minimize the maximum contaminant concentration at the end of the pumping time, and to minimize the cost of remediation. The second formulation, GWDA, aims to minimize the average contaminant concentration, along with cost. The mathematical formulation of GWDA and GWDM are depicted in Table 2.

### 5 Results and analysis

#### 5.1 Experimental setup

We tested our algorithm on the test functions and groundwater problems discussed in the previous section. The performance of GOMORS was compared against the Non-Dominated

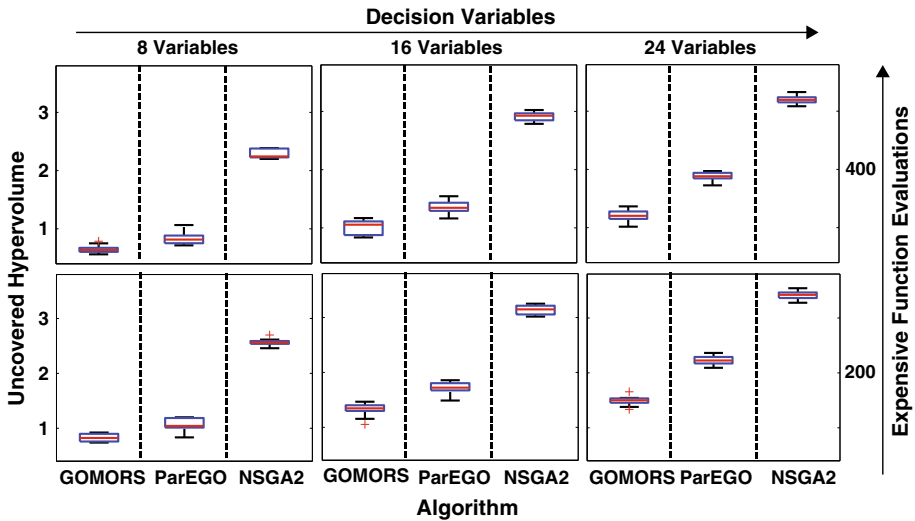
Sorting Algorithm-II (NSGA-II) proposed by Deb et al. [5] (discussed in Sect. 3.3) and the kriging-based ParEGO algorithm proposed by Knowles [21]. All three algorithms are quite different. ParEGO is a multi-objective version of EGO [19] where the multi-objective problem is converted into many single objective optimization problems through Tchebycheff weighting [46]. One single objective problem is chosen at random from the predefined set of decomposed problems and EGO is applied to it to for selection of one point for evaluation per algorithm iteration. ParEGO is not designed for high dimensional problems (more than 15 variables). GOMORS on the other hand embeds RBFs within an evolutionary framework, and selects multiple points (from various rules defined in Sect. 3.4) for simultaneous (parallel) evaluations in each iteration and is designed for low and higher (15–25 decision variables) dimensional problems.

Since the objective of GOMORS is to find good solutions to MO problems within a limited function evaluation budget, our experiments were restricted to 400 function evaluations. Since all algorithms compared are stochastic, ten optimization experiments were performed for each algorithm, on each test problem, and results were compared via visual analysis of fronts and a performance metric based analysis. A detailed description of the experimental setup, including parameter settings for all algorithms and source code references for ParEGO and NSGA-II, is provided in Section E of the online supplement.

The **uncovered hypervolume** metric [1] was used to compare the performance of various algorithms. Uncovered hypervolume is the difference between the total feasible objective space (defined by the reference and ideal points in Fig. 1a) and the objective space dominated by estimate of the Pareto front obtained by an algorithm. A lower value of the metric indicates a better solution and the ideal value is zero.

Results from the synthetic test problems were analyzed in combination through the metric. Experiments were performed for each test problem with 8 decision variables, 16 decision variables and 24 decision variables to highlight performance differences of GOMORS, ParEGO and NSGA-II with varying problem dimensions. Results of all synthetic test problems were compiled for analysis by summing the metric values of each of the ten optimization experiments performed on each test problem. Since the uncovered hypervolume metric values obtained for each individual test problem and algorithm combination could be considered as independent random variables, a sum of them across a single algorithm is another random variable which is a convolution of the independent random variables. This convolution based metric summarizes overall performance of an algorithm on all synthetic test problems and is used as basis of our analysis methodology.

Results from the two variants of the groundwater remediation problem (GWDA and GWDM) were also analyzed through the uncovered hypervolume metric in a similar manner. Experiments were performed with 6 decision variables, 12 decision variables and 24 decision variables for each groundwater problem. Results from individual subsequent experiments performed on each problem were summed to obtain a convoluted metric analysis of the groundwater problems. Performance of GOMORS and ParEGO on the GWDM test problem was further assessed through visual analysis of non-dominated solutions obtained from each algorithm (median solution). The numerical comparisons are based on the number of objective function evaluations and hence do not incorporate parallel speedup. ParEGO does not have a parallel implementation so its wall-clock time is much longer than GOMORS. However, the comparisons here focus on the number of function evaluations and evaluate an estimate of total CPU time. These comparisons do not consider the additional advantage of GOMORS (parallel) in wall-clock time.



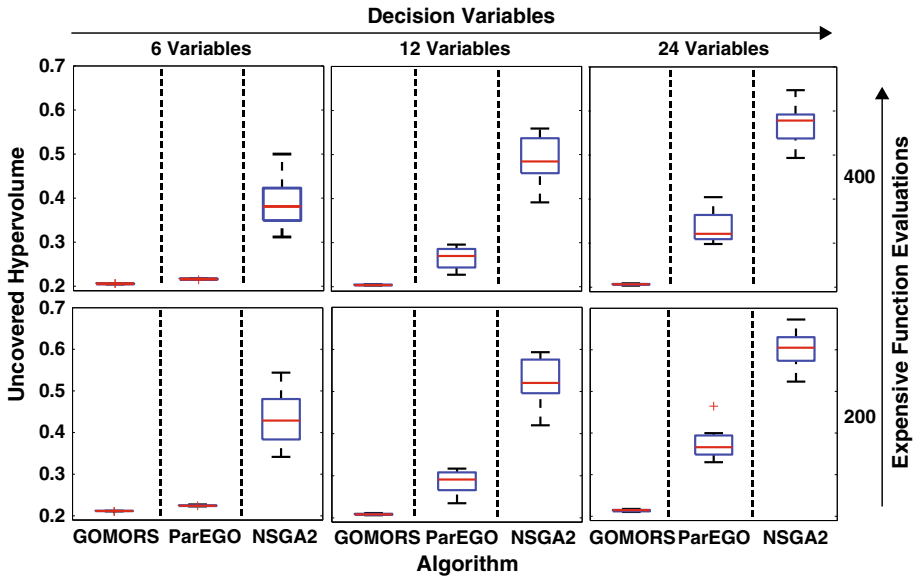
**Fig. 2** Synthetic test problems: *box plots* of uncovered hypervolume metric values summed over the eleven test problems. *Axis scales* are identical across all sub-plots, which describe problems ranging from 8 to 24 decisions and from 200 to 400 function evaluations. *Upper row* is for 400 evaluations and *lower row* is for 200 evaluations. In all cases lower values of uncovered hypervolume are best

### 5.2 Results: synthetic test problems

The metric analysis performed on the synthetic test problems is depicted via box plots in Fig. 2. The vertical axis in Fig. 2 is the convoluted uncovered hypervolume metric described in Sect. 5.1. The lower the uncovered hypervolume, the better is the quality of the Non-dominated Front that the algorithm has found.

Figure 2 aims to (1) visualize speed of convergence of all algorithms (by comparing results of 200 and 400 objective function evaluations) and (2) understand the effect of increasing decision space dimensions on performance of algorithms (by comparing problems with 8, 16 and 24 decision variables). The box plot visualization of each algorithm within each sub-plot of Fig. 2 corresponds to the uncovered hypervolume metric values summed over the eleven test problems (see Sect. 5.1). Lower values of the metric signify superiority of performance and a lower spread within a box plot depicts robustness of performance. Each sub-plot within Fig. 2 compares performance of all three algorithms for a specified number of decision variables (number of decision variables vary between 8 and 24) and a fixed number of function evaluations (either 200 or 400). Traversing from bottom to top, one can visualize the change in performance of algorithms as function evaluations increase (from 200 to 400), and a left to right traversal can help in visualizing performance differences with an increase in decision space dimensions (8, 16 and 24 decision variables).

Figure 2 clearly illustrates that GOMORS outperforms ParEGO for the 24 variable versions of the synthetic problems, both in terms of speed of convergence (at 200 evaluations) and at algorithm termination after 400 function evaluations (see top right sub-plot of Fig. 2). GOMORS’ convergence to a good solution is faster than ParEGO for the 8 and 16 variable versions of the synthetic test problems, but the difference is not as distinguishable (from Fig. 2) as in the case of the 24 variable versions. The Wilcoxon rank-sum test [14] (at 5 percent significance) confirms that performance of GOMORS is better than ParEGO for the 8, 16



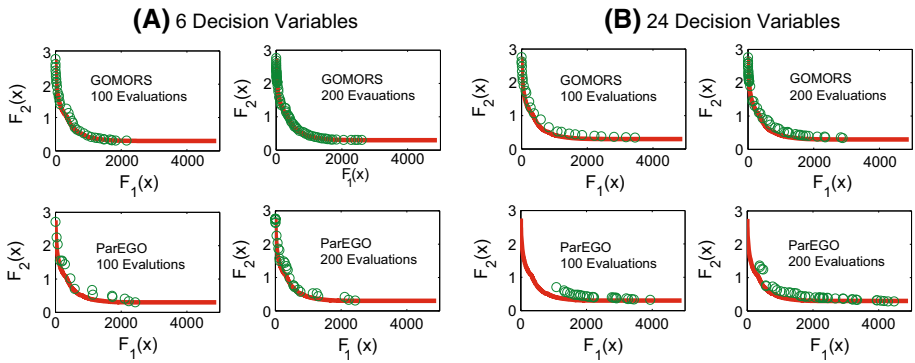
**Fig. 3** Groundwater application problems: *box plots* of uncovered hypervolume metric values summed over the two groundwater remediation design optimization problems, i.e GWDA and GWDM. See Fig. 2 caption for figure explanation

and 24 variable versions of the test problems after 100, 200 and 400 function evaluations. Figure 2 also indicates that both GOMORS and ParEGO significantly outperform NSGA-II with reference to the synthetic problems when evaluations are limited to 400.

### 5.3 Results: groundwater remediation design problem

The box-plot analysis methodology for the groundwater remediation problem is similar to the one employed for the synthetic test problems and the analysis is summarized in Fig. 3. Results summarized in Fig. 3 are consistent with the findings observed with the synthetic test functions in Fig. 2. GOMORS and ParEGO both outperform NSGA-II for the two MO groundwater problems defined in Table 2, GWDA and GWDM, when function evaluations are limited to 400. Performance of GOMORS (as per Fig. 3) is superior to ParEGO with application to the 12 and 24 variable versions of GWDA and GWDM, both in terms of speed of convergence (at 200 evaluations) and upon algorithm termination after 400 function evaluations. The difference between GOMORS and ParEGO is not visually discernible for the 6 variable versions of the problems, but the rank-sum test (at 5 percent significance) confirms that performance of GOMORS is better, which is supported by Fig. 4.

Figure 4 provides a visual comparison of non-dominated trade-offs obtained from GOMORS and ParEGO with application to the GWDM groundwater problem. The red line within each sub-plot is an estimate of the Pareto front of GWDM. Since knowledge of the true front is not known, we obtained our estimate of true Pareto front through a single trial of NSGA-II with 50,000 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 two sub-figures within Fig. 4 and four sub-plots within each sub-figure. Figure 4a corresponds to the median (the fronts ranked fifth in ten experi-



**Fig. 4** Estimated non-dominated front for groundwater problem GWDM: visual comparison of median non-dominated fronts (as per uncovered hypervolume) obtained from GOMORS and ParEGO for GWDM with 6 (Fig. 4a) and 24 (Fig. 4b) decisions, and after 100 and 200 expensive function evaluations. Red line is result of 50,000 evaluations with NSGA-II. Green circles are non-dominated solutions from GOMORS or ParEGO

ments of each algorithm) non-dominated fronts obtained from each algorithm for GWDM with 6 decision variables and Fig. 4b corresponds to the median fronts obtained from each algorithm for GWDM with 24 decision variables. A clock-wise traversal of sub-plots within each sub-figure depict results in the following order: (1) GOMORS after 100 evaluations, (2) GOMORS after 200 evaluations, (3) ParEGO after 200 evaluations, and (4) ParEGO after 100 evaluations.

Figure 4 indicates that both algorithms seem to converge to the estimated Pareto front and also manage to find diverse solutions on the front for the 6 variable version of GWDM. More diverse solutions are obtained from GOMORS however, and with better speed of convergence than with ParEGO (depicted by the 6 variable version results after 100 and 200 evaluations). Performance of GOMORS is significantly better than ParEGO for the 24 variable version of the problem both in terms of speed of convergence (depicted by 24 variable GWDM version results after 100 and 200 evaluations) and diversity of solutions. In case of computationally expensive real-world problems it may not be possible to run multiple MO experiments. Hence, we also visually analyzed trade-offs of worst case solutions (as per uncovered hypervolume) obtained from GOMORS and ParEGO for both GWDM and GWDA. The visualizations are provided in Section F of the online supplement to this paper. After performing a metric based analysis and a visual analysis, it can be concluded that performance of GOMORS is superior to ParEGO on the test problems examined here, especially on the relatively higher dimensional problems.

## 6 Conclusion

GOMORS is a new parallel algorithm for multi-objective optimization of black box functions that improves efficiency for computationally expensive objective functions by obtaining good solutions with a relatively small number of evaluations (<500). GOMORS accomplishes this with the construction in each iteration of a surrogate response surface based on all the values of the objective functions computed in the current and previous iterations. Then evolution and non-dominated sorting are applied to the inexpensive function describing this surrogate surface. It is then possible to evaluate a large number of points on the surrogate

inexpensively so multiple Rules can be used to select a diversity of points for expensive evaluation on parallel processors. The use of these multiple Rules is the innovative aspect of GOMORS and contributes to its effectiveness and robustness with limited numbers of evaluations (<500). GOMORS is very different from ParEGO, which solves multiple single objective problems.

Our numerical results indicate that GOMORS was significantly more effective than ParEGO when the number of evaluations was quite limited relative to the difficulty of the problem (Figs. 2, 3, 4) for both the test functions and the groundwater partial differential equation models. Computational demands for nonlinear optimization will grow rapidly as the dimension increases, so the effectiveness of GOMORS becomes more obvious on higher dimensional problems, as is evident in Figs. 2, 3, 4.

There are many real application models with computational times that are so large that the evaluations will be greatly limited, especially for multi-objective problems. For example, analysis of a carbon sequestration monitoring problem required global optimization with seven decisions of a nonlinear, multiphase flow transient PDF model that took over 2 hours per simulation [11]. So even 100 evaluations for a problem like this is probably more evaluations than are feasible. Hence the GOMORS approach to multi-objective optimization is a contribution in the area of surrogate-assisted multi-objective optimization of objective functions based on computationally expensive, multimodal, black box models.

**Acknowledgments** The research was supported by a Fulbright Fellowship to Akhtar and grants to Shoemaker from NSF (CISE 1116298) and to Mahowald and Shoemaker from DOE (SciDAC).

**Open Access** This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

## References

1. Bader, J.M.: Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods. CreateSpace, Paramount (2010)
2. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.* **181**(3), 1653–1669 (2007)
3. Buhmann, M.D.: Radial Basis Functions. Cambridge University Press, New York (2003)
4. Deb, K.: Multi-objective genetic algorithms: problem difficulties and construction of test problems. *Evol. Comput.* **7**(3), 205–230 (1999)
5. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: 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, pp. 849–858. Springer, London, UK (2000)
6. Deb, K., Kalyanmoy, D.: Multi-Objective Optimization Using Evolutionary Algorithms, 1st edn. Wiley, New York (2001)
7. Deb, K., Nain, P.: An evolutionary multi-objective adaptive meta-modeling procedure using artificial neural networks. In: Yang, S., Ong, Y.S., Jin, Y. (eds.) *Evolutionary Computation in Dynamic and Uncertain Environments*, vol. 51, chap. 13, pp. 297–322. Springer, Berlin (2007)
8. Diaz-Manriquez, A., Toscano-Pulido, G., Gomez-Flores, W.: On the selection of surrogate models in evolutionary optimization algorithms. *IEEE Congr. Evol. Comput.* 2155–2162 (2011)
9. Emmerich, M., Giotis, A., Özdemir, M., Bäck, T., Giannakoglou, K.: Metamodel-assisted evolution strategies. In: *Parallel Problem Solving from Nature VII*. Springer, pp. 361–370 (2002)
10. Emmerich, M.T.M., Giannakoglou, K., Naujoks, B.: Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Trans. Evol. Comput.* **10**(4), 421–439 (2006)
11. Espinet, A., Shoemaker, C.A., Doughty, C.: Estimation of plume distribution for carbon sequestration using parameter estimation with limited monitoring data. *Water Resour. Res.* **49**(7), 4442–4464 (2013)
12. Fang, H., Rais-Rohani, M., Liu, Z., Horstemeyer, M.F.: A comparative study of metamodeling methods for multiobjective crashworthiness optimization. *Comput. Struct.* **83**(25–26), 2121–2136 (2005)

13. Georgopoulou, C.A., Giannakoglou, K.C.: A multi-objective metamodel-assisted memetic algorithm with strength-based local refinement. *Eng. Optim.* **41**(10), 909–923 (2009)
14. Gibbons, J.D., Chakraborti, S.: *Nonparametric Statistical Inference*. Chapman and Hall, Boca Raton (2010)
15. Gutmann, H.M.: A radial basis function method for global optimization. *J. Global Optim.* **19**, 201–227 (2001)
16. Huang, D., Allen, T.T., Notz, W.I., Zeng, N.: Global optimization of stochastic black-box systems via sequential kriging meta-models. *J. Global Optim.* **34**(3), 441–466 (2006)
17. Jin, R., Chen, W., Simpson, T.W.: Comparative studies of metamodeling techniques under multiple modelling criteria. *Struct. Multidiscip. Optim.* **23**(1), 1–13 (2001)
18. Jin, Y., Olhofer, M., Sendhoff, B.: A framework for evolutionary optimization with approximate fitness functions. *IEEE Trans. Evol. Comput.* **6**(5), 481–494 (2002)
19. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. *J. Global Optim.* **13**(4), 455–492 (1998)
20. Karakasis, M.K., Giannakoglou, K.C.: On the use of metamodel-assisted, multi-objective. *Eng. Optim.* **38**(8), 941–957 (2006)
21. Knowles, J.: ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans. Evol. Comput.* **8**(5), 1341–66 (2006)
22. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *IEEE Trans. Evol. Comput.* **13**(2), 284–302 (2009)
23. Lim, D., Jin, Y., Ong, Y.S., Sendhoff, B.: Generalizing surrogate-assisted evolutionary computation. *IEEE Trans. Evol. Comput.* **14**(3), 329–355 (2010)
24. Martínez, S.Z., Coello, C.A.C.: A memetic algorithm with non gradient-based local search assisted by a meta-model. In: *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature: Part I. PPSN'10*. Springer, Berlin, Heidelberg, pp. 576–585 (2010)
25. Martínez, S.Z., Coello, C.A.C.: Combining surrogate models and local search for dealing with expensive multi-objective optimization problems. In: *IEEE Congress on Evolutionary Computation*, pp. 2572–2579. IEEE (2013)
26. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**(2), 239–245 (1979)
27. Minsker, B.S., Shoemaker, C.A.: Differentiating a finite element biodegradation simulation model for optimal control. *Water Resour. Res.* **32**(1), 187–192 (1996)
28. Minsker, B.S., Shoemaker, C.A.: Dynamic optimal control of in-situ bioremediation of ground water. *Water Resour. Plan. Manag.* **124**(3), 149–161 (1998)
29. Montemayor-garcía, G., Toscano-pulido, G.: A Study of Surrogate models for their use in multiobjective evolutionary algorithms. In: *8th International Conference on Electrical Engineering Computing Science and Automatic Control (CCE)* (2011)
30. Mueller, J., Shoemaker, C.A., Piche, R.: SO-I: a surrogate model algorithm for expensive nonlinear integer programming problems including global optimization applications. *J. Global Optim.* **59**(4), 865–889 (2014)
31. Mueller, J., Shoemaker, C.A., Piche, R.: SO-MI: a surrogate model algorithm for computationally expensive nonlinear mixed-integer, black-box global optimization problems. *Comput. Oper. Res.* **40**(5), 1383–1400 (2013)
32. di Pierro, F., Khu, S.T., Savić, D., Berardi, L.: Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms. *Environ. Model. Softw.* **24**(2), 202–213 (2009)
33. Ponweiser, W., Wagner, T., Biermann, D., Vincze, M.: Multiobjective optimization on a limited budget of evaluations using model-assisted s-metric selection. In: *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature: PPSN X*. Springer, Berlin, Heidelberg, pp. 784–794 (2008)
34. Powell, M.J.D.: *The Theory of Radial Basis Function Approximation in 1990*, pp. 105–210. Oxford University Press, USA (1992)
35. Priddy, K.L., Keller, P.E.: *Artificial Neural Networks: An Introduction*. SPIE Press, New York (2005)
36. Regis, R., Shoemaker, C.: Local function approximation in evolutionary algorithms for the optimization of costly functions. *IEEE Trans. Evol. Comput.* **8**(5), 490–505 (2004)
37. Regis, R., Shoemaker, C.: A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. Comput.* **19**, 497–509 (2007)
38. Regis, R.G., Shoemaker, C.A.: Constrained global optimization of expensive black box functions using radial basis functions. *J. Global Optim.* **31**, 153–171 (2005)
39. Regis, R.G., Shoemaker, C.A.: Improved strategies for radial basis function methods for global optimization. *J. Global Optim.* **37**(1), 113–135 (2007)

40. Regis, R.G., Shoemaker, C.A.: Parallel stochastic global optimization using radial basis functions. *INFORMS J. Comput.* **21**(3), 411–426 (2009)
41. Regis, R.G., Shoemaker, C.A.: A quasi-multistart framework for global optimization of expensive functions using response surface models. *J. Global Optim.* **56**(4), 1719–1753 (2013)
42. Regis, R.G., Shoemaker, C.A.: Combining radial basis function surrogates dynamic coordinate search in high dimensional expensive black-box optimization. *Eng. Optim.* **45**(5), 529–555 (2013)
43. Rosales-p. A., Coello, C.A.C., Gonzalez, J.A., Reyes-garcia, C.A., Escalante, H.J.: A hybrid surrogate-based approach for evolutionary multi-objective optimization. In: *IEEE Congress on Evolutionary Computation (CEC)*, pp. 2548–2555 (2013)
44. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. *Stat. Sci.* **4**(4), 409–423 (1989)
45. Santana-Quintero, L., Serrano-Hernandez, V., Coello Coello, C.A., Hernandez-Diaz, A., Molina, J.: Use of radial basis functions and rough sets for evolutionary multi-objective optimization. In: *IEEE Symposium on Computational Intelligence in Multicriteria Decision Making*, pp. 107–114 (2007)
46. Steuer, R., Choo, E.U.: An interactive weighted tchebycheff procedure for multiple objective programming. *Math. Program.* **26**, 326–344 (1983)
47. Vapnik, V.N.: *Statistical Learning Theory*, 1st edn. Wiley, New York (1998)
48. Vrugt, J.A., Robinson, B.A.: Improved evolutionary optimization from genetically adaptive multimethod search. *Proc. Natl. Acad. Sci. USA* **104**(3), 708–711 (2007)
49. Wild, S.M., Regis, R.G., Shoemaker, C.A.: ORBIT: optimization by Radial Basis Function Interpolation in Trust-Regions. *SIAM J. Sci. Comput.* **30**(6), 3197–3219 (2007)
50. Wild, S.M., Shoemaker, C.A.: Global convergence of radial basis function trust-region algorithms for derivative-free optimization. *SIGEST article. SIAM Rev.* **55**(2), 349–371 (2013)
51. Martínez, Zapotecas: S., Coello Coello, C.A.: Moea/d assisted by rbf networks for expensive multi-objective optimization problems. In: *Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference. GECCO '13*, pp. 1405–1412. ACM, New York, NY, USA (2013)
52. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007)
53. Zhang, Q., Liu, W., Tsang, E., Virginas, B.: Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Trans. Evol. Comput.* **14**(3), 456–474 (2010)