

RESEARCH ARTICLE

10.1002/2015WR018230

Key Points:

- Develop a surrogate-based multiobjective optimization method
- Test the method with 13 functions and one land surface model
- Evaluate the effectiveness and efficiency of the proposed method

Correspondence to:

W. Gong,
gongwei2012@bnu.edu.cn

Citation:

Gong, W., Q. Duan, J. Li, C. Wang, Z. Di, A. Ye, C. Miao, and Y. Dai (2016), Multiobjective adaptive surrogate modeling-based optimization for parameter estimation of large, complex geophysical models, *Water Resour. Res.*, 52, 1984–2008, doi:10.1002/2015WR018230.

Received 13 OCT 2015

Accepted 14 DEC 2015

Accepted article online 18 DEC 2015

Published online 16 MAR 2016

The copyright line for this article was changed on 11 APR 2016 after original online publication.

Multiobjective adaptive surrogate modeling-based optimization for parameter estimation of large, complex geophysical models

Wei Gong^{1,2}, Qingyun Duan^{1,2}, Jianduo Li^{1,2}, Chen Wang^{1,2}, Zhenhua Di^{1,2}, Aizhong Ye^{1,2}, Chiyuan Miao^{1,2}, and Yongjiu Dai^{1,2}

¹College of Global Change and Earth System Science (GCESS), Beijing Normal University, Beijing, China, ²Joint Center for Global Change Studies (JCGCS), Beijing Normal University, Beijing, China

Abstract Parameter specification is an important source of uncertainty in large, complex geophysical models. These models generally have multiple model outputs that require multiobjective optimization algorithms. Although such algorithms have long been available, they usually require a large number of model runs and are therefore computationally expensive for large, complex dynamic models. In this paper, a multiobjective adaptive surrogate modeling-based optimization (MO-ASMO) algorithm is introduced that aims to reduce computational cost while maintaining optimization effectiveness. Geophysical dynamic models usually have a prior parameterization scheme derived from the physical processes involved, and our goal is to improve all of the objectives by parameter calibration. In this study, we developed a method for directing the search processes toward the region that can improve all of the objectives simultaneously. We tested the MO-ASMO algorithm against NSGA-II and SUMO with 13 test functions and a land surface model - the Common Land Model (CoLM). The results demonstrated the effectiveness and efficiency of MO-ASMO.

1. Introduction

Specification of model parameters can significantly affect the simulation ability of large, complex geophysical models such as rainfall-runoff models [Duan *et al.*, 1992], land surface hydrological models [Gong *et al.*, 2015a; Li *et al.*, 2013], and numerical weather and climate prediction models [Di *et al.*, 2015; Neelin *et al.*, 2010]. Parameter specification is complicated by the fact that today's geophysical models consider an increasing number of physical processes (e.g., water and energy cycles, biogeochemical processes, etc.), and a multiobjective optimization approach is required to estimate model parameters such that all major physical processes of interest are well simulated. This approach, however, might be too computationally expensive because of two reasons: (1) A typical geophysical model may require several hours or even days to run a multiyear simulation over a large region. (2) The multiobjective optimization problem is much more difficult than single-objective optimization, thus requiring larger number of model runs to find the global Pareto optimal solutions. Although many multiobjective optimization algorithms have long been available, such as MOGA [Fonseca and Fleming, 1993], NPGA [Horn *et al.*, 1994], SPEA [Zitzler and Thiele, 1998], PAES [Knowles and Corne, 1999], NSGA-II [Deb *et al.*, 2002], MOSCEM [Vrugt *et al.*, 2003b], and SMS-EMOA [Emmerich *et al.*, 2005], nearly all of them require a large number of model runs (typically as many as 10^5 – 10^6 or even more) to identify the Pareto optimal sets [Li *et al.*, 2012; Liu *et al.*, 2004, 2005; Vrugt *et al.*, 2003a, 2003b]. This high computational cost makes the parameter estimation for large, complex geophysical models very difficult, if not impossible.

Surrogate modeling-based optimization is an efficient way to reduce the computational burden of parameter optimization. As summarized in the review paper by Razavi *et al.* [2012], surrogate-based optimization, which replaces the original expensive modes with inexpensive statistical surrogates in the optimization process, has been widely used in hydrological modeling and water resource management. In recent years, many multiobjective surrogate-based optimization approaches have emerged. Some of them are based on the classical nondominated sorting of multiple objectives, simply replacing expensive dynamic models with inexpensive surrogates. For example, Nain and Deb [2005] integrated an artificial neural network (ANN) with NSGA-II (Non-dominated Sorting Genetic Algorithm II). This method is called NSGA-II-ANN. For a total of m

© 2015. The Authors.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

generations of NSGA-II-ANN, the first n generations use the original simulation model, whereas the remaining $m-n$ generations use the surrogate model. *Jourdan et al.* [2006] proposed LEMMO, which uses a Learnable Evolution Model [*Michalski*, 2000], the C4.5 machine learning method, as a surrogate to replace the original model in NSGA-II. *Syberfeldt et al.* [2008] proposed the Multi-Objective Parallel Surrogate-Assisted Evolution Algorithm (MOPSA-EA), which uses an ANN as a surrogate and considers surrogate imprecision and parallel computing. *Kourakos and Mantoglou* [2013] developed a multiobjective optimization algorithm named MOSA (Multi-Objective Surrogate Assisted method) based on a modular neural network (MNN) surrogate and applied it to solve the aquifer management problem in Santorini, Greece. They improved the pool of offspring technique proposed by *Syberfeldt et al.* [2008] by using three criteria to select the most promising offspring and evaluated it using the original model. This is actually an adaptive sampling technique. In this constrained multiobjective case study, MOSA (MNN) can yield better optimization results than NSGA-II with fewer model runs (MOSA (MNN) used 3100, whereas NSGA-II used 40000).

Instead of nondominated sorting, some researchers have tried to transform the multiobjective problem into a single-objective problem and use existing single-objective surrogate-based optimization methods. *Knowles* [2006] proposed the ParEGO algorithm, a multiobjective version of EGO (Efficient Global Optimization [*Jones*, 2001; *Jones et al.*, 1998]). EGO uses DoE-inspired initial sampling and a Gaussian processes surrogate model to accelerate single-objective optimization, and ParEGO uses an augmented Tchebycheff function to transform multiobjective optimization problems into multiple single-objective problems. *Emmerich et al.* [2006] proposed the Meta-model Assisted Evolution Strategy (MAES), an integrated framework for comparing prescreening strategies for both single- and multiobjective optimization. Prescreening is a strategy that generates candidate points from a surrogate model to allow for simulation with the original complex model; the surrogate model can thus be adjusted with the newly simulated points. MAES implemented four types of prescreening strategies for single-objective optimization: mean value (MI), probability of improvement (Pol), expected improvement (ExI), lower confidence bound (LCB), and the S hypervolume metric, which was used to transform multiobjective problems into single-objective problems. The S hypervolume metric proposed by *Fleischer* [2003] represents the volume of hyperspace dominated by the Pareto frontier. The unique significance of *Fleischer's* research lies in the formal proof that the maximization of the hypervolume covered by a point set is the necessary and sufficient condition of the multiobjective Pareto optimal. However, although the S metric is theoretically elegant and very intuitive for simple two-objective problems, its application to large problems is very limited because the computational cost grows rapidly if the number of points is large for higher dimensions. Furthermore, *Couckuyt et al.* [2012, 2013] developed the hypervolume-based Pol strategy, which has been implemented in the SUMO toolbox [*Gorissen*, 2010].

Many comparisons of different surrogate-based multiobjective algorithms have been carried out. *Syberfeldt et al.* [2008] compared MOPSA-EA with MAES, SMS-EMOA and NSGA-II-ANN and found that MOPSA-EA is the most effective and efficient method. Over a total of 3000 original model runs, MOPSA-EA fit the Pareto frontier best. *di Pierro et al.* [2009] compared LEMMO and ParEGO (as well as an evolutionary algorithm, PESA-II [*Corne et al.*, 2001]) in addressing two water distribution network design problems. LEMMO was able to find the same solutions using only 10% of the simulations required by PESA-II, whereas ParEGO, which could complete the optimization using only 1% of the simulations, did not fit the Pareto frontier as well as LEMMO and PESA-II did. *Tsoukalas and Makropoulos* [2015] compared the surrogate-based ParEGO, SUMO with hypervolume-based Pol strategy, and SMS-EGO (another modified EGO based on hypervolume, proposed by *Ponweiser et al.* [2008]), as well as the classical nonsurrogate algorithm NSGA-II and SMS-EMOA. The results indicated that the surrogate-based algorithms could archive similar Pareto optimals with much fewer model runs, and SUMO outperformed the ParEGO and SMS-EGO methods. As summarized by *Maier et al.* [2014], despite the report of comparative studies, evaluations of effectiveness and efficiency are still limited to testing several selected algorithms with simple test functions or with respect to one or two special problems; furthermore, there are too many statistically indistinguishable algorithms that can be decomposed into common components. To evaluate algorithm performance comprehensively, therefore, comparisons should be based on an online optimization service featuring various components instead of a large number of different types of algorithms.

In this study, we developed a multiobjective adaptive surrogate modeling-based optimization (MO-ASMO) algorithm to meet the special requirements of large, complex geophysical models. The novel contributions

of this study are (1) the selection of the most suitable initial sampling and surrogate modeling method. As shown by Gong *et al.* [2015b], the Good Lattice Points (GLP) method [Fang, 1980; Korobov, 1959a, 1959b; Wang and Fang, 1981] with ranked Gram-Schmidt (RGS) de-correlation [Owen, 1994] is one of the most effective and efficient sampling methods. In Gong *et al.* [2015a], among five surrogate models that were compared, the Gaussian processes regression (GPR) outperformed others with respect to the Common Land Model [Dai *et al.*, 2003]. However, GPR may not serve as the all-purpose surrogate for any problem, and the best surrogate can be problem dependent (as shown in Gong *et al.* [2015b]) such that surrogate model selection is necessary for each problem [Pilát and Neruda, 2013]. Consequently, in this study, we used GLP with RGS de-correlation as the initial sampling method and used GPR as a surrogate model. (2) The development of an adaptive sampling strategy to save computational resources. Nondominated sorting-based algorithms usually use all Pareto optimal points given by surrogate models for the original dynamic model to simulate, whereas the hypervolume-based or Tchebycheff function-based algorithms usually simulate a single point. The former can better maintain the diversity of a population but require more computational resources, whereas the later can archive the Pareto frontier more quickly but suffer from premature convergence. In this study, we only used the most representative subsets of the Pareto optimal points for simulation. This adaptive sampling strategy is simple but effective in balancing convergence and diversity. (3) Use of the default parameterization scheme to constrain the optimization result. Typical dynamic geophysical models, such as land surface models, weather and climate models, have default parameterization schemes that depend on model physics. With parameter optimization, our goal is to improve the performance of all model objectives, and of course, the solutions dominated by the default scheme are not useful in this case. In this study, we developed a mechanism that can make the optimization focus on the region that can improve all of the objectives relative to what can be achieved with the default scheme. This paper is organized as follows: Section 2 details the four components of the MO-ASMO algorithm; section 3 evaluates the effectiveness and efficiency of the algorithm using 13 test functions and a case study of a land surface model; and section 4 provides concluding remarks and discusses plans for future work.

2. Methodology

2.1. Elitist Strategy

As mentioned in the introduction, the goal of multiobjective optimization is to find the Pareto frontier, a surface in the output space on which the solution cannot be improved in one objective without doing harm to another. An effective multiobjective optimization method should be able to find the optimal parameter set that uniformly populates the entire Pareto frontier. The effectiveness of a multiobjective optimization algorithm can be evaluated in two respects: convergence and diversity. Convergence refers to minimizing the average distance between the achieved Pareto optimal points and the true Pareto frontier, whereas diversity refers to the uniformity of distribution matching the true Pareto frontier. The convergence and diversity represent two goals of multiobjective optimization. For all objectives, our goal is to find the optimal solutions that can exhaust the possibility of improving every objective simultaneously while maintaining diversity such that if some objectives can be sacrificed we can achieve the optimal for each of the remaining objectives.

As described by Deb *et al.* [2002], the convergence metric γ measures the fitness between the Pareto optimal points obtained by an optimization procedure and the true Pareto frontier (as the distance c_i in Figure 1a). The average value of the distance c_i between the optimized and true Pareto set is defined as the convergence metric γ :

$$\gamma = \frac{1}{n} \sum_{i=1}^n c_i \tag{1}$$

where n is the number of Pareto optimal points. The best value of γ is 0. The diversity metric Δ measures the span of solutions that can effectively cover the entire Pareto frontier. The diversity metric Δ is defined as

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{n-1} |d_i - \bar{d}|}{d_f + d_l + (n-1)\bar{d}} \tag{2}$$

where d_f and d_l are the distances between the extreme true solutions and the extreme obtained solutions (as shown in Figure 1b). d_i are the distances between consecutive solutions, and their average is \bar{d} . The

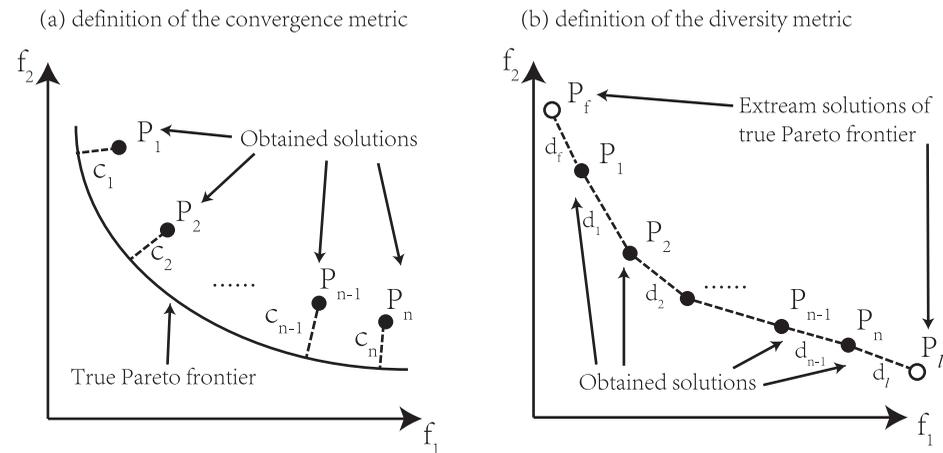


Figure 1. The definition of the convergence metric and the diversity metric.

best value of Δ is 0 when $d_i = \bar{d}$ (uniformly distributed) and $d_f = d_l = 0$ (extending to extreme Pareto solutions). Note that the convergence metric γ can be applied to any multiobjective problem, whereas the diversity metric Δ can only be applied to two-objective problems.

In evolution algorithms, the fittest individuals are able to survive and breed, whereas the least fit individuals are eliminated. The elitist strategy is one way to determine the fitness of each individual. In the NSGA-II algorithm [Deb et al., 2002], the elitist strategy comprises two components: the fast-nondominated-sorting method and the crowding distance. (1) The fast-nondominated-sorting method can compute the Pareto rank of each individual. As shown in Figure 2a, the points with a small Pareto rank dominate those with larger Pareto ranks. For each point with a large rank, we can find at least one point with a smaller rank that surpasses it in every objective. The nondominated-sorting can direct the evolution of a population toward the true Pareto surface, i.e., improve the convergence metric. (2) The crowding distance is defined as the sum of the edge length of the cuboid, as shown in Figure 2b. A large crowding distance indicates that a point is far from its neighbors; thus, the point might be more representative and require further exploration around it. The crowding distance directs the population toward unexplored regions, and this mechanism can effectively improve the diversity of the population. In NSGA-II, the individuals are first sorted using a nondominated-sorting method in ascending order, and within each Pareto rank the points are sorted using crowding distance in descending order (Figure 2c). The individuals near the tail are eliminated, and the survivors have the opportunity to generate offspring.

The goal of parameter calibration of dynamic geophysical models is to improve the models' simulation ability. Consequently, if we already have a parameterization scheme, ideally the optimization can improve all of the objectives. Those Pareto optimal points that improve some objectives but make others worse than the default parameterization are meaningless in this framework. As shown in Figure 2d, the reference point of objectives is given by the default parameter scheme, and the plane is divided into four regions: region 1, the nondominated region of the reference point; regions 2 and 3, dominated by f_1 and f_2 , respectively; and region 4, the dominated region of the reference point. According to the goal of multiobjective optimization of dynamic models, the aim is to improve all of the objectives. Therefore, we are most interested in the Pareto optimal points located in region 1. The points in regions 2 and 3, although possibly "optimal" according to the Pareto rank, are useless because they degrade at least one objective relative to the default parameterization scheme.

To direct the evolution procedure toward the nondominated region (region 1 as in Figure 2d), we define a weighted crowding distance, which replaces the classical crowding distance that maintains the diversity of Pareto optimal points. The weighted crowding distance multiplies the classical crowding distance by a very small weight factor if one of the objectives is worse than the reference point, while the crowding distances in the nondominated region remain unchanged. This mechanism changes the order within each Pareto rank, making the individual that degrades some objectives more likely to be eliminated. The classical crowding distance forces the Pareto optimal points to be uniformly distributed along the Pareto frontier, whereas

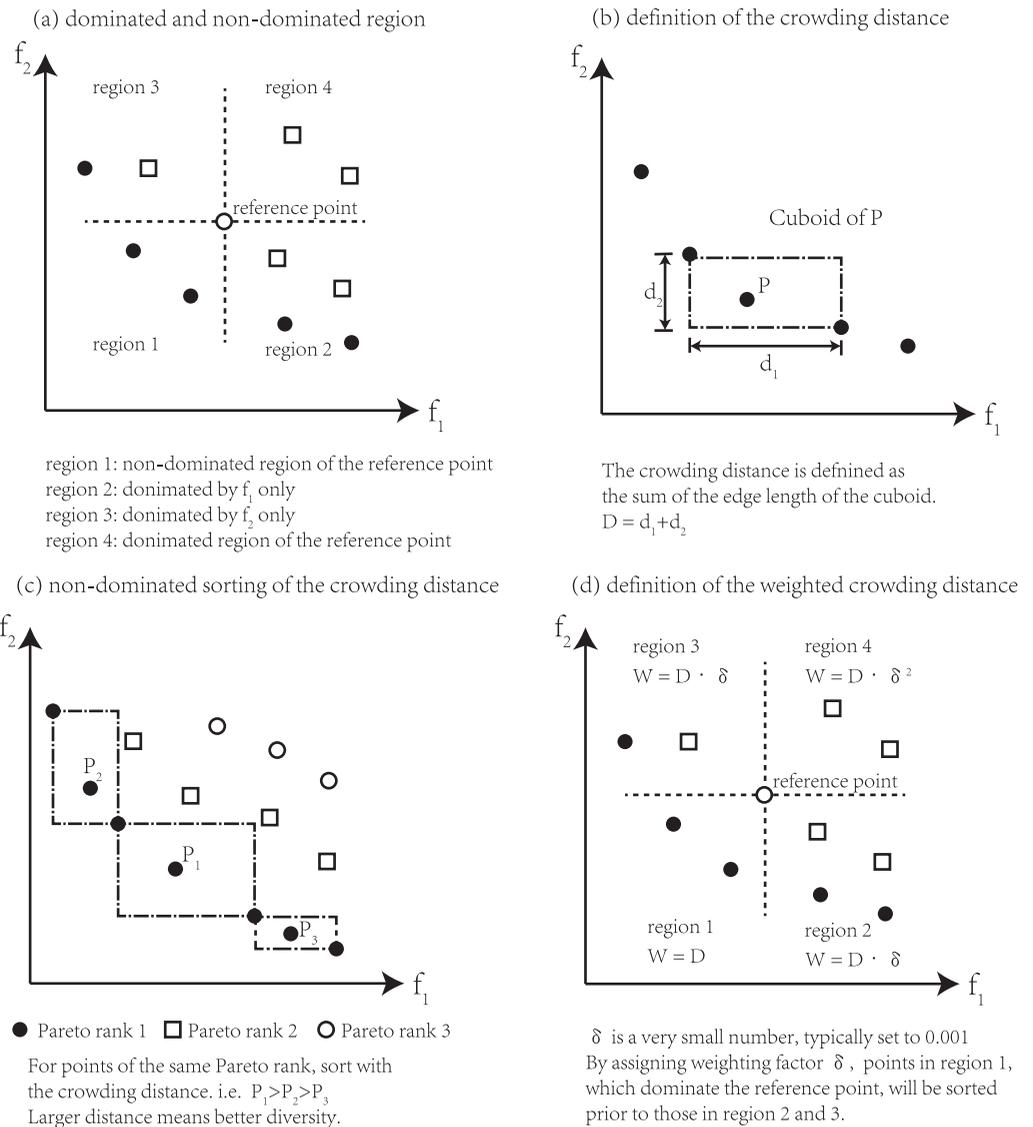


Figure 2. The definition of the crowding distance and the weighted crowding distance.

the weighted crowding distance forces the search to focus on the nondominated region of the reference point and also maintains the uniform distribution along the Pareto frontier within this region. The weighing factor δ is a small positive value less than 1. Setting $\delta=0$ indicates that the optimization is rigorously constrained to the nondominated region of the reference point, whereas $\delta=1$ indicates that there is no constraint from the reference point and the full Pareto frontier can be explored. In this paper, we set $\delta=0.001$. This value can help us search not only the nondominated region 1 but also the half-dominated regions (regions 2 and 3 in Figure 2d), which might provide useful information about the region that we are interested in, i.e., region 1.

2.2. Main Program of MO-ASMO

The MO-ASMO algorithm, as shown in the pseudo code, involves the following steps:

Step 1: Problem definition. First, we must specify the original dynamic model's adjustable parameters (inputs) and objective functions (outputs). If the model has too many adjustable parameters, sensitivity analysis must be used to screen out the most important parameters to reduce the complexity of multiobjective optimization.

Step 2: Initial sampling. A uniformly distributed sample set is generated using a specific sampling method. Based on the study of *Gong et al.* [2015b], we use the GLP with RGS de-correlation to generate initial samples. As suggested by *Wang et al.* [2014], a proper initial sample size should be 15–20 times the number of parameters. Other initial sampling methods that can generate a uniformly distributed point set, such as Latin Hypercube [*McKay et al.*, 1979], Symmetric Latin Hypercube [*Ye et al.*, 2000], Halton sequence [*Halton*, 1964], and Sobol' sequence [*Sobol'*, 1967], are also applicable in this step. The details of initial sampling are presented in section 2.3.

After generating a uniformly distributed sample set, the original dynamic model simulation is run with the generated parameters and objective functions are calculated from the raw outputs of the model.

Step 3: Main loop of MO-ASMO. First, build or update surrogate models for each objective function. Gaussian Processes Regression (GPR) has been demonstrated to outperform other methods [*Gong et al.*, 2015a; *Wang et al.*, 2014]. Moreover, GPR can easily integrate reinforcement learning, which can update the trained surrogate model with an additional sample set as reinforcement. GPR has several hyper-parameters that control the behavior of the Gaussian processes. To obtain a better-suited surrogate model, the hyper-parameters can be optimized using the SCE-UA method. Other surrogate modeling methods, such as MARS [*Friedman*, 1991], SVM [*Vapnik*, 2002], random forests [*Breiman*, 2001], ANN [*Cybenko*, 1989], are also applicable in this step. The details of surrogate modeling are presented in section 2.4.

Then run a multiobjective optimization algorithm on the surrogate models built in the previous step, and Pareto optimal solutions are obtained. In this study, we use NSGA-II [*Deb et al.*, 2002]. If the dynamic model has a default parameterization scheme, use weighted crowding distance to replace the crowding distance in NSGA-II. After running multiobjective optimization on the surrogate models, parts of the Pareto optimal solutions are selected for evaluation using the original dynamic model. To save on the number of original model runs, we do not evaluate all of the Pareto optimal solutions given by NSGA-II optimization, but only those (in this study, 20%) with the largest crowding distances, which can mostly represent the diversity of the Pareto optimal sets.

Finally, we can update the training data set with the dynamic model simulation results obtained by adaptive sampling. Repeat the main loop until the termination conditions are satisfied (iteration limit, total original model evaluation limit, etc.).

The pseudo code of MO-ASMO:

1. Problem definition:

$$\mathbf{y}=f(\mathbf{x})$$

where $f(\cdot)$ is the dynamic model, which has N adjustable parameters and M objective functions, \mathbf{x} is the N dimensional parameter vector and \mathbf{y} is the M dimensional objective vector.

2. Initial sampling:

Generate a $T \times N$ matrix \mathbf{X} using the Good Lattice Points method with RGS de-correlation, where T is the number of sample points.

Run the dynamic model for T times and obtain the multiobjective results \mathbf{Y} .

$$\mathbf{Y}=f(\mathbf{X})$$

where \mathbf{Y} is a $T \times M$ matrix containing the objective functions.

3. Main loop of MO-ASMO:

a. Surrogate model training:

$$\mathbf{y}=f^*(\mathbf{x})$$

where $f^*(\cdot)$ is the surrogate model. The surrogate model is trained with the parameter matrix \mathbf{X} as input and objective matrix \mathbf{Y} as output. Currently, we use Gaussian Processes Regression (GPR) as the surrogate.

b. Run NSGA-II multiobjective optimization on the surrogate model $f^*(\cdot)$, and obtain the Pareto optimal points \mathbf{X}^* . \mathbf{X}^* is a $T_{pop} \times N$ matrix, and T_{pop} is the population size.

c. From the matrix \mathbf{X}^* given by NSGA-II, select a portion of the solutions (typically 20%) that have the largest crowding distances (or the largest weighted crowding distance, if the reference point is provided). Run the dynamic model with the selected optimal points \mathbf{X}' , and obtain the simulation results:

$$\mathbf{Y}' = f(\mathbf{X}')$$

- d. Append \mathbf{X}' and \mathbf{Y}' to the input-output pool: $\mathbf{X} = [\mathbf{X}; \mathbf{X}']; \mathbf{Y} = [\mathbf{Y}; \mathbf{Y}']$.
- e. Return to step a) if termination conditions are not satisfied.

4. End of MO-ASMO

The pseudo code for determining the crowding distance:

D = crowding distance (\mathbf{Y}):

(\mathbf{Y} is the $T \times M$ matrix containing the objectives functions.)

1. Initialize the crowding distance of each point $\mathbf{Y}(i)$.distance = 0.
For each objective $m = 1$ to M ,
 - a. $y_m^{\max} = \max(\mathbf{Y}_m), y_m^{\min} = \min(\mathbf{Y}_m)$
($\mathbf{Y}_m = \mathbf{Y}(:, m)$ is the m -th objective.)
 - b. $\mathbf{Y} = \text{sort}(\mathbf{Y}, m)$.
(Sort the matrix \mathbf{Y} with the m -th objective.)
 - c. $\mathbf{Y}(1)$.distance = inf, $\mathbf{Y}(T)$.distance = inf.
(Set the crowding distance of boundary points to infinity.)
 - d. For each point $i = 2$ to $T - 1$,
 $\mathbf{Y}(i)$.distance = $\mathbf{Y}(i)$.distance + $(\mathbf{Y}(i+1, m) - \mathbf{Y}(i-1, m)) / (y_m^{\max} - y_m^{\min})$
3. Return the distance vector $D = \mathbf{Y}$.distance.

The pseudo code for determining the weighted crowding distance:

W = Weighted crowding distance (\mathbf{Y}, Y_{ref})

1. D = crowding distance (\mathbf{Y}):
2. $W = D$
(Initialize W , the weighted crowding distance, a T -dimensional vector.)
3. For each point $i = 1$ to T ,
If $\mathbf{Y}(i, m) \geq Y_{ref}(m)$:
 $W(i) = W(i) \cdot \delta$

(δ is a very small number; in this study, we set $\delta = 0.001$)
4. Return the weighted crowding distance W .

2.3. Initial Sampling

Based on the findings of Gong et al. [2015b], Good Lattice Points (GLP) [Fang, 1980] with ranked Gram-Schmidt (RGS) de-correlation [Owen, 1994] was demonstrated to be the most uniform method among several uniform sampling methods evaluated. Therefore, the initial sampling method used in MO-ASMO is GLP with de-correlation. For n points s factors initial sampling, we can define a U-array: $U_n(n^s) = [u_{ij}]_{n \times s}$, each column of which is a permutation of $1, \dots, n$. For each element of the U-array, u_{ij} is the j -th factor of the i -th sample. The U-array can be normalized to $[0, 1]$ using the following transformation:

$$x_{ij} = \frac{u_{ij} - 0.5}{n} \quad (3)$$

The U-array is generated as $u_{ij} = i \cdot h_j \pmod{n}$ (and assign $i \cdot n \pmod{n} = n$), where h_j is the j -th element of the powered generating vector H_a . Let integers a and t satisfy the following: (1) $a < n, t \geq s - 1$, (2) $a^{t+1} = 1 \pmod{n}$, (3) a and n are relatively prime, and (4) the elements of sequence $a, a^2, \dots, a^t \pmod{n}$ are different from each other; thus, the subsequence $H_a = \{a^0, a^1, \dots, a^{s-1} \pmod{n}\}$ is defined as the powered generating vector.

In addition to generating a proper sampling design, de-correlation is another way to improve uniformity. De-correlation algorithms include Ranked Cholesky (RC) proposed by Iman and Conover [1982] and Ranked Gram-Schmidt (RGS) proposed by Owen [1994]. These algorithms can remove the correlation between

factors while maintaining the spatial structure of samples. In this study, we use GLP sampling with RGS de-correlation. A more detailed description is provided by Gong *et al.* [2015b].

2.4. Surrogate Model

Based on our previous study and work by other researchers, GPR is one of the effective multidimensional nonlinear regression algorithms [Gong *et al.*, 2015a; Rasmussen and Williams, 2006]. In this paper, we build multiple GPRs for each objective function. A brief introduction is provided below.

Suppose there is a regression model

$$y=f(\mathbf{x})+\varepsilon \tag{4}$$

where $f(\mathbf{x})$ is the regression function, $\mathbf{x}=(x_1, x_2, \dots, x_n)$ is the input vector, y is the output variable, and $\varepsilon \sim N(0, \sigma_n^2)$ is i.i.d. Gaussian noise. Suppose there is an ensemble of functions satisfying equation (2); their mean and covariance satisfy

$$\begin{cases} m(\mathbf{x})=E[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}')=E[(f(\mathbf{x})-m(\mathbf{x}))(f(\mathbf{x}')-m(\mathbf{x}'))] \end{cases} \tag{5}$$

where $m(\mathbf{x})$ is called the mean function and $k(\mathbf{x}, \mathbf{x}')$ is called the covariance function, and the GPR model can be expressed as $f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$.

The training procedure of GPR (building the GPR surrogate model from the samples) is as follows: Suppose X is the input matrix, where each column is an input vector \mathbf{x} ; \mathbf{y} is the output vector, where each element is the original dynamic model's objective function; X_* is the predicting input matrix, namely, the samples that we want to evaluate using the GPR surrogate model; and \mathbf{f}_* is the predicting output. Therefore, the joint distribution of training and predicting inputs and outputs is a joint Gaussian distribution as follows:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} K(X, X)+\sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \tag{6}$$

where $K(X_1, X_2)$ is the covariance matrix of two input matrices, which can be calculated from the covariance function:

$$k_{\nu=5/2}(r)=\left(1+\frac{\sqrt{5}r}{l}+\frac{5r^2}{3l^2}\right)\exp\left(-\frac{\sqrt{5}r}{l}\right) \tag{7}$$

where $r=|\mathbf{x}_1-\mathbf{x}_2|$ is the Euclidian distance between each sample point (i.e., each column) in X_1 and X_2 . This covariance function is the Martérn covariance function with the parameter $\nu=5/2$. The Martérn covariance function is an isotopic function in which the covariance between function values at different points, i.e., $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$, only depends on the distance between \mathbf{x}_1 and \mathbf{x}_2 and the covariance decreases with increasing distance. The predicted mean and variance of $f(\mathbf{x})$ are expressed as follows:

$$\begin{cases} \mathbf{E}(\mathbf{f}_*)=K(X_*, X)[K(X, X)+\sigma_n^2 I]^{-1}\mathbf{y} \\ \mathbf{V}(\mathbf{f}_*)=K(X_*, X_*)-K(X_*, X)[K(X, X)+\sigma_n^2 I]^{-1}K(X, X_*) \end{cases} \tag{8}$$

Training GPR is time consuming because it requires the Cholesky decomposition of a large matrix, $K(X, X)+\sigma_n^2 I$. A positive-definite real matrix A can be decomposed into the product of a lower triangular matrix and its transpose: $A=LL^T$. Cholesky decomposition is formulated as follows:

$$\begin{cases} L_{jj}=\sqrt{A_{jj}-\sum_{k=1}^{j-1}L_{jk}^2} \\ L_{ij}=\frac{1}{L_{jj}}\left(A_{ij}-\sum_{k=1}^{j-1}L_{ik}L_{jk}\right), \text{ for } i > j \end{cases} \tag{9}$$

Cholesky decomposition is numerically very stable and simple to implement. The decomposition starts from the left upper corner and proceeds row by row; thus, if there are any additional samples, the L matrix can be easily updated by adding new rows behind. This procedure is also called "reinforcement learning". The computational burden of surrogate modeling can be significantly reduced by reinforcement learning.

GPR is a highly flexible regression method that has various types of covariance functions. GPR can become Kriging interpolation, Radial Basis Function (RBF) interpolation, Infinite node single layer feed-forward ANN, if proper covariance function is adopt. Furthermore, the hyper-parameter values can also control the behavior of GPR. For instance, if the noise term σ_n^2 is set to 0, GPR will serve as a multidimensional interpolation method. If the characteristic length l is large, GPR will be smooth, but if l is small, GPR will be sensitive to small changes of the outputs. To choose appropriate hyper-parameter values, we maximize the marginal likelihood function

$$\log [p(\mathbf{y}|X)] = -\frac{1}{2} \mathbf{y}^T [K(X, X) + \sigma_n^2 I]^{-1} \mathbf{y} - \frac{1}{2} \log |K(X, X) + \sigma_n^2 I| - \frac{n}{2} \log 2\pi \quad (10)$$

using the SCE-UA optimization algorithm [Duan et al., 1993].

3. Case Study

3.1. Synthetic Studies to Evaluate the Effectiveness and Efficiency of MO-ASMO

We conducted a total of 13 synthetic case studies, including five simple math functions with no more than three inputs and two objectives, five ZDT test problems with two objectives [Zitzler et al., 2000], and three DTLZ test problems with three objectives [Deb et al., 2005]. The math functions' names, abbreviations, references, equations, and true Pareto frontiers are presented below.

The five simple test functions are shown in Table 1. The first two functions SCH1 and SCH2 were proposed by Schaffer [1987]. Both have only one input variable and two outputs. SCH1 is the simplest convex test function, and SCH2 is a convex discontinuous test function. The details of SCH1 and SCH2 can be found in the book by Deb [2001]. The BIN function is test case 2 originally proposed by Binh and Korn [1997] and also used by Binh [1999]. The KIT function was originally proposed by Kita et al. [1996] and used by Binh [1999] as test case 4. Both the BIN and KIT functions have a constrained version, but in this paper, we use the unconstrained versions. The FON function was originally proposed by Fonseca and Fleming [1998a, 1998b]. The function has two or more inputs and two outputs. In this paper, we use the three-input version. The FON function has a nonconvex Pareto frontier.

Moreover, we want to test the algorithm with more complex problems. Zitzler et al. [2000] proposed a method for conveniently constructing test problems for multiobjective optimization and constructed six test functions. In this paper, we only use functions 1 – 4 and 6 (ZDT5 is a Boolean function for bit strings), as shown in Table 2. The ZDT functions are two-objective problems defined within the following common framework:

$$\begin{cases} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) = g(\mathbf{x})h(\mathbf{x}) \end{cases} \quad (11)$$

where $f_1(\mathbf{x})$ controls the difficulty along the Pareto frontier, $g(\mathbf{x})$ controls the difficulty lateral to the Pareto frontier and $h(\mathbf{x})$ controls the miscellaneous properties of the Pareto frontier, such as convex/nonconvex, continuous/discontinuous, and oscillations. The Pareto optimal is achieved if $g(\mathbf{x})=1$. The ZDT function family not only offers a common framework that can develop multiobjective test problems from single-objective test problems but also helps diagnose the issues with a multiobjective optimization method. For more details about the ZDT test functions, please refer to the book written by Deb [2001] and the PhD thesis by Van Veldhuizen [1999].

The ZDT test problems have multiple input parameters and miscellaneous properties but only have two objectives. Deb et al. [2005] proposed DTLZ test problems that have multiple objective functions. The DTLZ test problems have two outstanding advantages: (1) they are scalable and (2) they yield the theoretical true value of the Pareto frontier. With respect to (1), the DTLZ test problems can be easily extended to multiple adjustable parameters and objective functions and have similar expressions. With respect to (2), the DTLZ test problems are built by a bottom-up approach; therefore, the true Pareto frontier can be a hyper-plane, unit sphere, curve, or comet. Among many combinations in the DTLZ family, we adopt three, as shown in Table 3, where n is the number of adjustable parameters, M is the number of objectives, and \mathbf{x}_M is a k -dimensional vector that can compute the theoretical true Pareto frontier ($n=M+k-1$). All of the adopted problems have 6 adjustable parameters and 3 objective functions. The DTLZ1 is the simplest one that has a

Table 1. Simple Multiobjective Optimization Test Functions

Problem and Reference	No. of Inputs	Bounds of Inputs	Objective Functions	Optimal Solutions	Comments
SCH1 [Schaffer, 1987]	1	[-10,10]	$\begin{cases} f_1 = x^2 \\ f_2 = (x-2)^2 \end{cases}$	$x \in [0, 2]$	convex
SCH2 [Schaffer, 1987]	1	[-5,10]	$f_1 = \begin{cases} -x, & \text{if } x \leq 1 \\ x-2, & \text{if } 1 < x \leq 3 \\ 4-x, & \text{if } 3 < x \leq 4 \\ x-4, & \text{if } x > 4 \end{cases}$ $f_2 = (x-5)^2$	$x \in [0, 2]$	convex discontinuous
BIN [Binh and Korn, 1997; Binh, 1999]	2	[-15,30]	$\begin{cases} f_1 = 4x_1^2 + 4x_2^2 \\ f_2 = (x_1 - 5)^2 + (x_2 - 5)^2 \end{cases}$	$x_1, x_2 \in [0, 5]$ $x_1 = x_2$	convex
KIT [Kita et al., 1996; Binh, 1999]	2	[0,7]	$\begin{cases} f_1 = x_1^2 - x_2 \\ f_2 = -0.5x_1 - x_2 - 1 \end{cases}$	$x_1 \in [0, 7]$ $x_2 = 7$	convex
FON [Fonseca and Fleming, 1998a, 1998b]	3	[-4,4]	$\begin{cases} f_1 = 1 - \exp\left(-\sum_{i=1}^3 \left(x_i - \frac{1}{\sqrt{3}}\right)^2\right) \\ f_2 = 1 - \exp\left(-\sum_{i=1}^3 \left(x_i + \frac{1}{\sqrt{3}}\right)^2\right) \end{cases}$	$x_i \in \left[-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right], i=1, 2, 3$ $x_1 = x_2 = x_3$	nonconvex

hyper-plane Pareto frontier. The DTLZ1 has a global Pareto optimal hyper-plane with $11^k - 1$ local Pareto optimal fronts. The DTLZ2 problem has a unit sphere Pareto frontier, which is convex and nonlinear. The DTLZ3 problem, which is the most difficult one, also features a unit sphere Pareto frontier but has $3^k - 1$ local Pareto optimal fronts. More details about the DTLZ test problems can be found in the study by Deb et al. [2005].

To evaluate the effectiveness and efficiency of MO-ASMO, we compared it with the classical NSGA-II algorithm [Deb et al., 2002] as well as the surrogate toolbox SUMO [Couckuyt et al., 2012, 2013; Gorissen, 2010], which uses hypervolume-based Pol and outperforms other surrogate-based algorithms, according to the

Table 2. The Family of ZDT Multiobjective Optimization Test Functions Proposed in Zitzler et al. [2000]

Problem and Reference	No. of Inputs	Bounds of Inputs	Objective Functions	Optimal solutions	Comments
ZDT1	30	[0,1]	$\begin{cases} f_1(\mathbf{x}) = x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - \sqrt{x_1/g(\mathbf{x})}\right] \\ g(\mathbf{x}) = 1 + 9 \left(\sum_{i=2}^n x_i\right) / (n-1) \end{cases}$	$x_1 \in [0, 1]$ $x_i = 0, i=2, \dots, n$	convex
ZDT2	30	[0,1]	$\begin{cases} f_1(\mathbf{x}) = x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - (x_1/g(\mathbf{x}))^2\right] \\ g(\mathbf{x}) = 1 + 9 \left(\sum_{i=2}^n x_i\right) / (n-1) \end{cases}$	$x_1 \in [0, 1]$ $x_i = 0, i=2, \dots, n$	nonconvex
ZDT3	30	[0,1]	$\begin{cases} f_1(\mathbf{x}) = x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - \sqrt{x_1/g(\mathbf{x})}\right] - \frac{x_1}{g(\mathbf{x})} \sin(10\pi x_1) \\ g(\mathbf{x}) = 1 + 9 \left(\sum_{i=2}^n x_i\right) / (n-1) \end{cases}$	$x_1 \in [0, 1]$ $x_i = 0, i=2, \dots, n$	convex, discontinuous
ZDT4	10	$x_1 \in [0, 1],$ $x_i \in [-5, 5],$ $i=2, \dots, 10$	$\begin{cases} f_1(\mathbf{x}) = x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - \sqrt{x_1/g(\mathbf{x})}\right] \\ g(\mathbf{x}) = 1 + 10(n-1) + \sum_{i=2}^n [x_i^2 - 10\cos(4\pi x_i)] \end{cases}$	$x_1 \in [0, 1]$ $x_i = 0, i=2, \dots, n$	nonconvex
ZDT6	10	[0,1]	$\begin{cases} f_1(\mathbf{x}) = 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\ f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2\right] \\ g(\mathbf{x}) = 1 + 9 \left[\left(\sum_{i=2}^n x_i\right) / (n-1)\right]^{0.25} \end{cases}$	$x_1 \in [0, 1]$ $x_i = 0, i=2, \dots, n$	nonconvex

Table 3. The Family of DTLZ Multiobjective Optimization Test Functions Proposed in *Deb et al. [2005]*

Problem and Reference	No. of Inputs/Outputs	Bounds of Inputs	Objective Functions	Optimal Solutions	Comments
DTLZ1	6/3	[0,1]	$\begin{cases} f_1(\mathbf{x}) = \frac{1}{2}x_1x_2 \dots x_{M-1}(1+g(\mathbf{x}_M)) \\ f_2(\mathbf{x}) = \frac{1}{2}x_1x_2 \dots (1-x_{M-1})(1+g(\mathbf{x}_M)) \\ \vdots \\ f_{M-1}(\mathbf{x}) = \frac{1}{2}x_1(1-x_2)(1+g(\mathbf{x}_M)) \\ f_M(\mathbf{x}) = \frac{1}{2}(1-x_1)(1+g(\mathbf{x}_M)) \\ g(\mathbf{x}_M) = 100 \left[(n-M+1) + \sum_{x_i \in \mathbf{x}_M} [(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))] \right] \end{cases}$	$\begin{cases} \mathbf{x}_M = 0.5 \\ \sum_{m=1}^M f_m = 0.5 \end{cases}$	Pareto frontier is a linear hyperplane
DTLZ2	6/3	[0,1]	$\begin{cases} f_1(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cos(x_2\pi/2) \dots \cos(x_{M-2}\pi/2)\cos(x_{M-1}\pi/2) \\ f_2(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cos(x_2\pi/2) \dots \cos(x_{M-2}\pi/2)\sin(x_{M-1}\pi/2) \\ f_3(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cos(x_2\pi/2) \dots \sin(x_{M-2}\pi/2) \\ \vdots \\ f_{M-1}(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\sin(x_2\pi/2) \\ f_M(\mathbf{x}) = (1+g(\mathbf{x}_M))\sin(x_1\pi/2) \\ g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} (x_i-0.5)^2 \end{cases}$	$\begin{cases} \mathbf{x}_M = 0.5 \\ \sum_{m=1}^M f_m^2 = 1 \end{cases}$	Pareto frontier is the unit sphere
DTLZ3	6/3	[0,1]	$\begin{cases} f_1(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cos(x_2\pi/2) \dots \cos(x_{M-2}\pi/2)\cos(x_{M-1}\pi/2) \\ f_2(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cos(x_2\pi/2) \dots \cos(x_{M-2}\pi/2)\sin(x_{M-1}\pi/2) \\ f_3(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cos(x_2\pi/2) \dots \sin(x_{M-2}\pi/2) \\ \vdots \\ f_{M-1}(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\sin(x_2\pi/2) \\ f_M(\mathbf{x}) = (1+g(\mathbf{x}_M))\sin(x_1\pi/2) \\ g(\mathbf{x}_M) = 100 \left[(n-M+1) + \sum_{x_i \in \mathbf{x}_M} [(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))] \right] \end{cases}$	$\begin{cases} \mathbf{x}_M = 0.5 \\ \sum_{m=1}^M f_m^2 = 1 \end{cases}$	Pareto frontier is the unit sphere, has many local optimals

comparison performed by *Tsoukalas and Makropoulos [2015]*. The algorithm setups are shown in Table 4. To provide a fair comparison of efficiency between the three algorithms, we used exactly the same number of original model runs. With the same limited number of original model runs, the effectiveness of the compared algorithms could be evaluated based on the convergence and diversity of the archived Pareto optimal points relative to the theoretical true Pareto frontier. To evaluate the efficiency, we added an additional NSGA-II setup in which the population size (pop) and number of generations (gen) were both set to 100. A surrogate-based algorithm is most efficient when its convergence and diversity is close to, or even better, than the results of large-population NSGA-II. For MO-ASMO, the population size (pop) and number of

Table 4. Optimization Algorithm Setup for the Case Studies in This Paper^a

Test Problem Name	NSGA-II	NSGA-II (Large)	MO-ASMO	SUMO	
Simple test problems	SCH1	11 pop × 10 gen = 110 total	100 pop × 100 gen = 10,000 total	10 init + 20 apt × 5 iter = 110 total	10 init + 100 iter = 110 total
	SCH2	11 pop × 10 gen = 110 total	100 pop × 100 gen = 10,000 total	10 init + 20 apt × 5 iter = 110 total	10 init + 100 iter = 110 total
	BIN	12 pop × 10 gen = 120 total	100 pop × 100 gen = 10,000 total	20 init + 20 apt × 5 iter = 120 total	20 init + 100 iter = 120 total
	KIT	12 pop × 10 gen = 120 total	100 pop × 100 gen = 10,000 total	20 init + 20 apt × 5 iter = 120 total	20 init + 100 iter = 120 total
	FON	13 pop × 10 gen = 130 total	100 pop × 100 gen = 10,000 total	30 init + 20 apt × 5 iter = 130 total	30 init + 100 iter = 130 total
ZDT test problems	ZDT1	40 pop × 10 gen = 400 total	100 pop × 100 gen = 10,000 total	300 init + 20 apt × 5 iter = 300 total	300 init + 100 iter = 400 total
	ZDT2	40 pop × 10 gen = 400 total	100 pop × 100 gen = 10,000 total	300 init + 20 apt × 5 iter = 300 total	300 init + 100 iter = 400 total
	ZDT3	40 pop × 10 gen = 400 total	100 pop × 100 gen = 10,000 total	300 init + 20 apt × 5 iter = 300 total	300 init + 100 iter = 400 total
	ZDT4	20 pop × 10 gen = 200 total	100 pop × 100 gen = 10,000 total	100 init + 20 apt × 5 iter = 200 total	100 init + 100 iter = 200 total
	ZDT6	20 pop × 10 gen = 200 total	100 pop × 100 gen = 10,000 total	100 init + 20 apt × 5 iter = 200 total	100 init + 100 iter = 200 total
	DTLZ test problems	DTLZ1	26 pop × 10 gen = 260 total	100 pop × 100 gen = 10,000 total	60 init + 20 apt × 10 iter = 260 total
DTLZ2		26 pop × 10 gen = 260 total	100 pop × 100 gen = 10,000 total	60 init + 20 apt × 10 iter = 260 total	60 init + 100 iter = 260 total
DTLZ3		26 pop × 10 gen = 260 total	100 pop × 100 gen = 10,000 total	60 init + 20 apt × 10 iter = 260 total	60 init + 100 iter = 260 total

^aAbbreviations: pop: number of population, gen: number of generation, init: number of initial sampling, apt: number of adaptive sampling in each iteration, iter: number of iteration.

Table 5. Mean and Standard Deviation of the Convergence and Diversity Metrics of the Pareto Optimal Solutions Given by NSGA-II and MO-ASMO^a

Test Problem Name	Optimization Algorithm	Mean of Convergence	Standard of Convergence	Mean of Diversity	Standard of Diversity		
Simple test problems	SCH1	NSGA-II	0.0062	2.0279E-03	0.8279	0.2189	
		NSGA-II(large)	0.0056	3.5517E-04	0.4280	0.0331	
		MO-ASMO	0.0056	3.1988E-04	0.6717	0.0487	
	SCH2	NSGA-II	0.0122	8.2872E-03	1.0901	0.1635	
		NSGA-II(large)	0.0085	6.1820E-04	1.0181	0.0228	
		MO-ASMO	0.0088	7.5874E-04	1.1170	0.0429	
	BIN	NSGA-II	38.2874	5.0316E+01	1.1511	0.1567	
		NSGA-II(large)	0.0923	6.8615E-03	0.5566	0.0413	
	KIT	MO-ASMO	0.0890	6.2079E-03	0.7301	0.0495	
		NSGA-II	0.5227	2.1924E-01	0.9752	0.1684	
	FON	NSGA-II(large)	0.0143	9.6561E-04	0.5158	0.0428	
		MO-ASMO	0.0137	1.1487E-03	0.7416	0.0503	
		NSGA-II	0.0151	1.0126E-02	0.9718	0.1022	
	ZDT test problems	ZDT1	NSGA-II(large)	0.0011	1.5653E-04	0.4109	0.0326
			MO-ASMO	0.0017	2.8113E-04	0.5602	0.0480
NSGA-II			1.8454	1.2354E-01	0.9235	0.0429	
ZDT2		NSGA-II(large)	0.7619	7.7708E-02	0.8655	0.0547	
		MO-ASMO	0.7453	6.6852E-02	0.8096	0.0398	
		NSGA-II	2.4971	1.3606E-01	1.0241	0.0411	
ZDT3		NSGA-II(large)	0.9967	1.2450E-01	1.0122	0.0399	
		MO-ASMO	1.1087	8.2943E-02	0.9717 (14)	0.0358	
		NSGA-II	1.7332	1.4593E-01	0.8776	0.0579	
ZDT4		NSGA-II(large)	0.6738	1.0195E-01	0.8843	0.0672	
		MO-ASMO	0.8896	1.2905E-01	0.7926	0.0589	
		NSGA-II	78.2058	9.7432E+00	0.9654	0.0611	
ZDT6		NSGA-II(large)	25.2251	5.9909E+00	1.0490	0.0748	
		MO-ASMO	38.5784	7.5076E+00	0.9204 (86)	0.0697	
		NSGA-II	4.9350	1.3718E-01	1.0093	0.0185	
DTLZ test problems	DTLZ1	NSGA-II(large)	3.2077	3.5467E-01	1.0914	0.0566	
		MO-ASMO	3.5064	5.7325E-01	0.9531	0.0598	
		NSGA-II	43.6747	15.0801			
	DTLZ2	NSGA-II(large)	7.1320	2.4005			
		MO-ASMO	36.7536	11.0271			
		NSGA-II	0.1487	0.0224			
	DTLZ3	NSGA-II(large)	0.0304	0.0054			
		MO-ASMO	0.0272	0.0027			
		NSGA-II	107.4802	32.5433			
DTLZ3	NSGA-II(large)	16.3201	6.2372				
	MO-ASMO	85.6173	21.6109				
	NSGA-II						

^aFor each cases, the optimization was replicated for 100 times. For both of the convergence and diversity metrics, the smaller the better and the best values are both 0.

generations (gen) of the embedded NSGA-II were also set to 100, and the resampling percentage (pct) was set to 20%, so that the adaptive sampling size in each iteration was $100 \times 20\% = 20$. The maximum number of iterations (iter) for simple and ZDT test problems was set to 5. For DTLZ test problems, iter was extended to 10 in order to give a fair comparison with SUMO. The total number of model runs of DTLZ problems was set to 260, which is close to 250 in *Couckuyt et al.* [2013]. Because in SUMO the comparison of hypervolume-based PoI is very time consuming, we did not run SUMO in the DTLZ case but compare with the results given by *Couckuyt et al.* [2013]. For other cases, SUMO was run only once, whereas other algorithms were replicated 100 times.

The optimization results of the test problems are shown in Table 5 and Figures 3 and 4. Table 5 presents the mean and standard deviation of the convergence and diversity metrics computed from 100 replications of NSGA-II and MO-ASMO with exactly the same number of original model runs (labeled NSGA-II and MO-ASMO, respectively) and computed with NSGA-II for a very large number of original model runs (labeled as NSGA-II (large)). As indicated in **equations (1) and (2) in section 2.1**, for both metrics, the smaller the better. The results of SUMO are not shown in this table because doing so would be too time consuming and we executed the algorithm only once for each test problem. The diversity metrics of DTLZ problems are also not shown because the adopted diversity metric currently supports only two-objective problems. As shown in Figures 3 and 4, we randomly selected one of the replications and plotted the objective functions

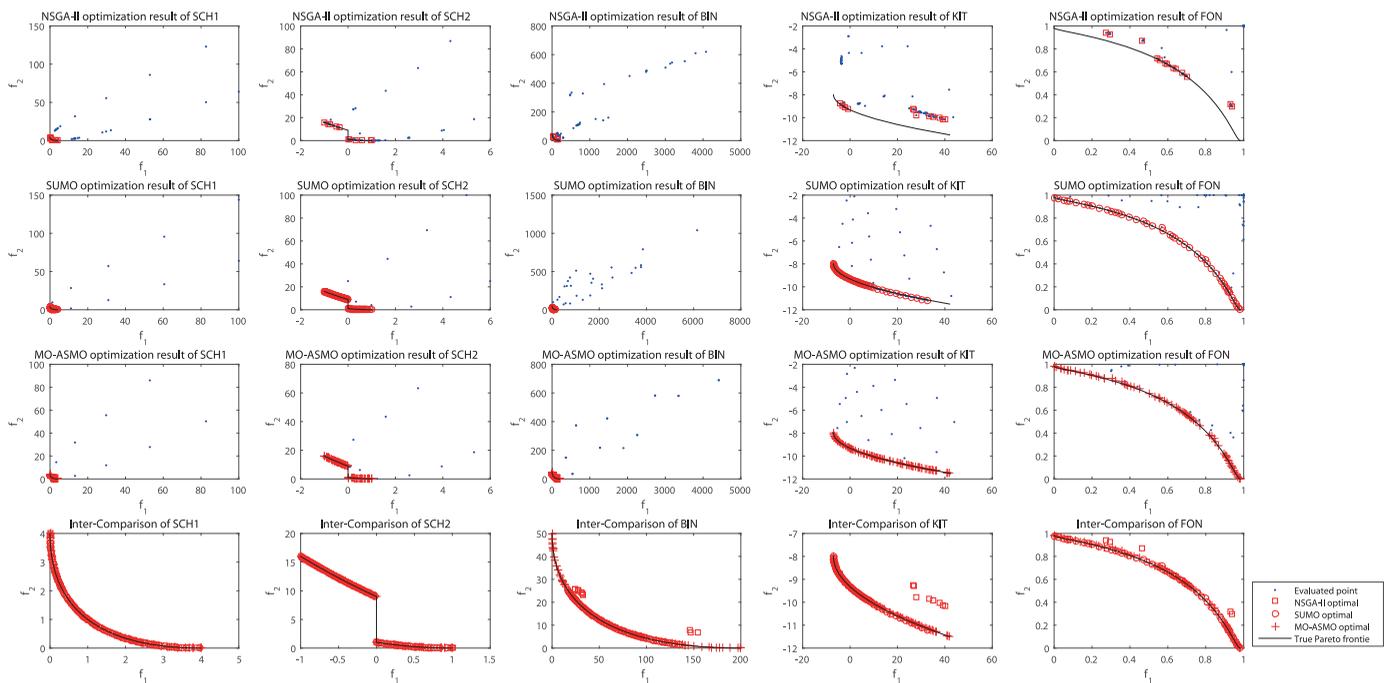


Figure 3. Multiobjective optimization results of simple test problems.

evaluated in the optimization procedure and the Pareto optimal obtained after the optimization was terminated.

As indicated in Table 5, for all of the test problems, the convergence metrics of MO-ASMO are significantly better than those of NSGA-II and very similar to those obtained by NSGA-II (large) except DTLZ1 and DTLZ3, indicating that for most evaluated cases the MO-ASMO algorithm is as effective but more efficient than the classical NSGA-II. For DTLZ1 and DTLZ3, which have many local Pareto-optimal fronts, MO-ASMO falls

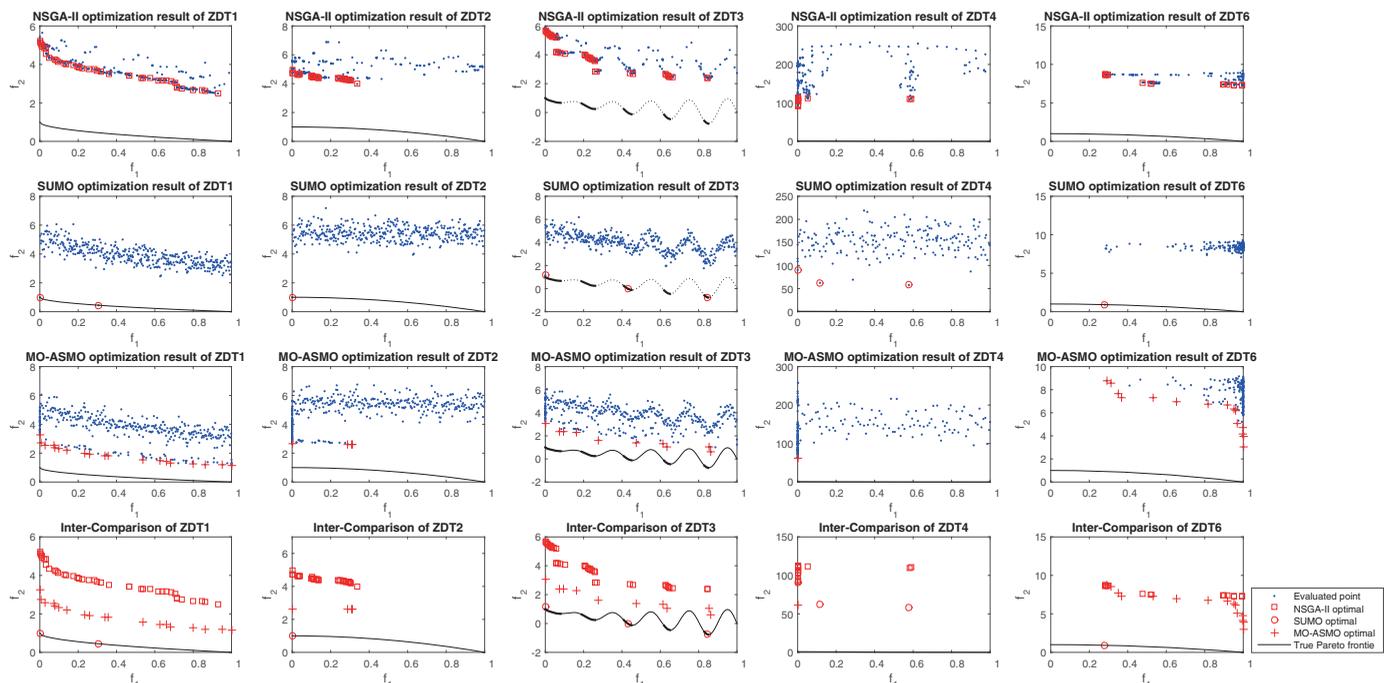


Figure 4. Multiobjective optimization results of ZDT test problems.

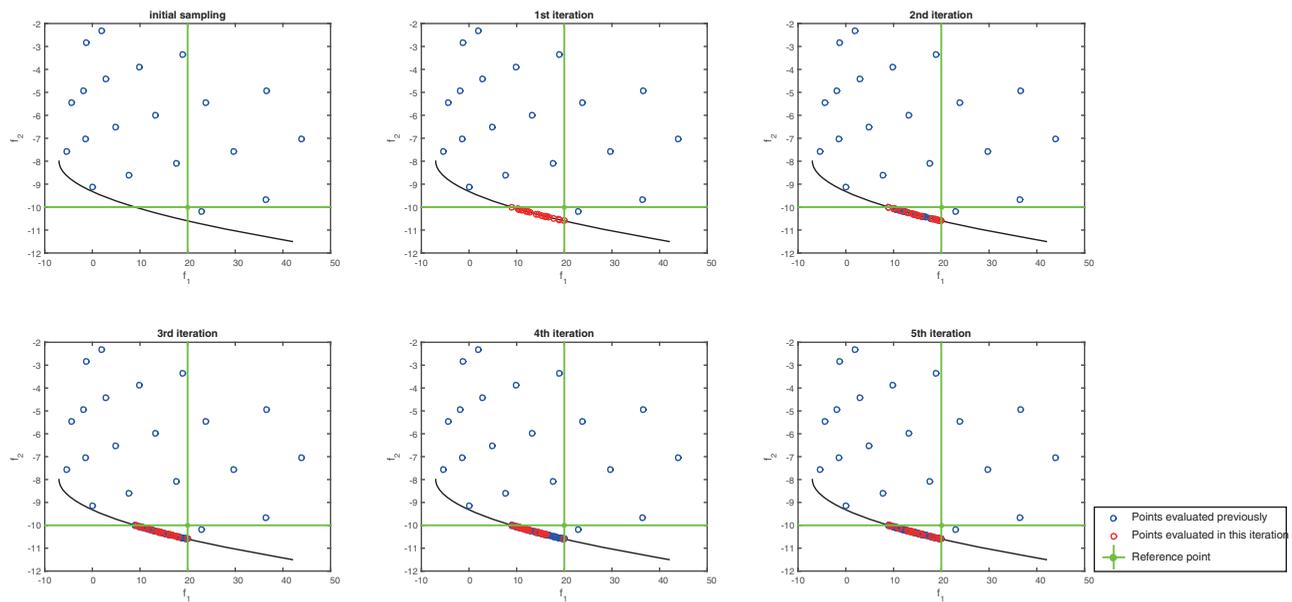


Figure 5. Multiobjective optimization procedure of KIT test problem using WMO-ASMO (with weighted crowding distance).

behind NSGA-II (large) but still significantly better than NSGA-II. The diversity metrics of MO-ASMO are slightly worse but still acceptable compared with those of NSGA-II (large). As shown in Figure 3, for simple test problems with exactly the same number of original model runs, NSGA-II fails to reach the true Pareto frontier, whereas both SUMO and MO-ASMO can almost perfectly find the true optimal with good convergence and diversity. MO-ASMO is as good as SUMO in these simple test cases.

For more complex ZDT test problems, the convergence metrics of MO-ASMO are slightly larger than but still similar to those of NSGA-II (large) and still much better than those of NSGA-II with a small sample size, except for the ZDT4 problem involving many local optimal Pareto frontiers; these results indicate that MO-ASMO can be as effective as NSGA-II with far fewer original model evaluations. For ZDT1, ZDT3, and ZDT6, the diversity metrics of MO-ASMO are better than those of NSGA-II and NSGA-II (larger), indicating that MO-ASMO can maintain diversity as effectively as but more efficiently than NSGA-II. The diversity metrics of

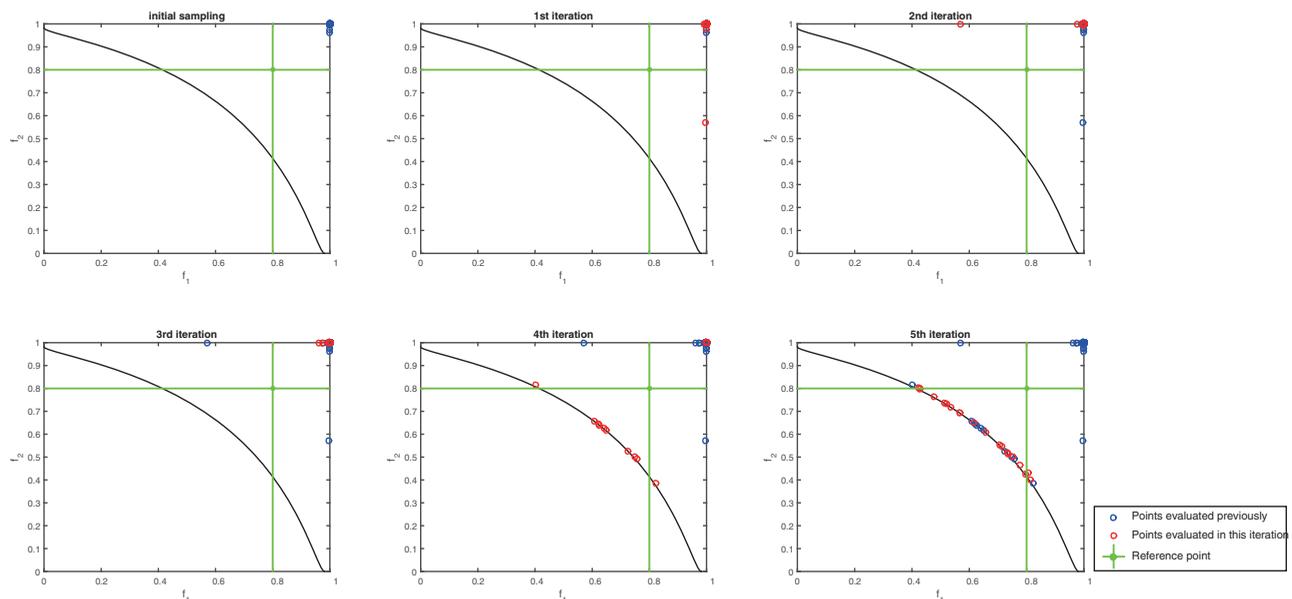


Figure 6. Multiobjective optimization procedure of FON test problem using WMO-ASMO (with weighted crowding distance).

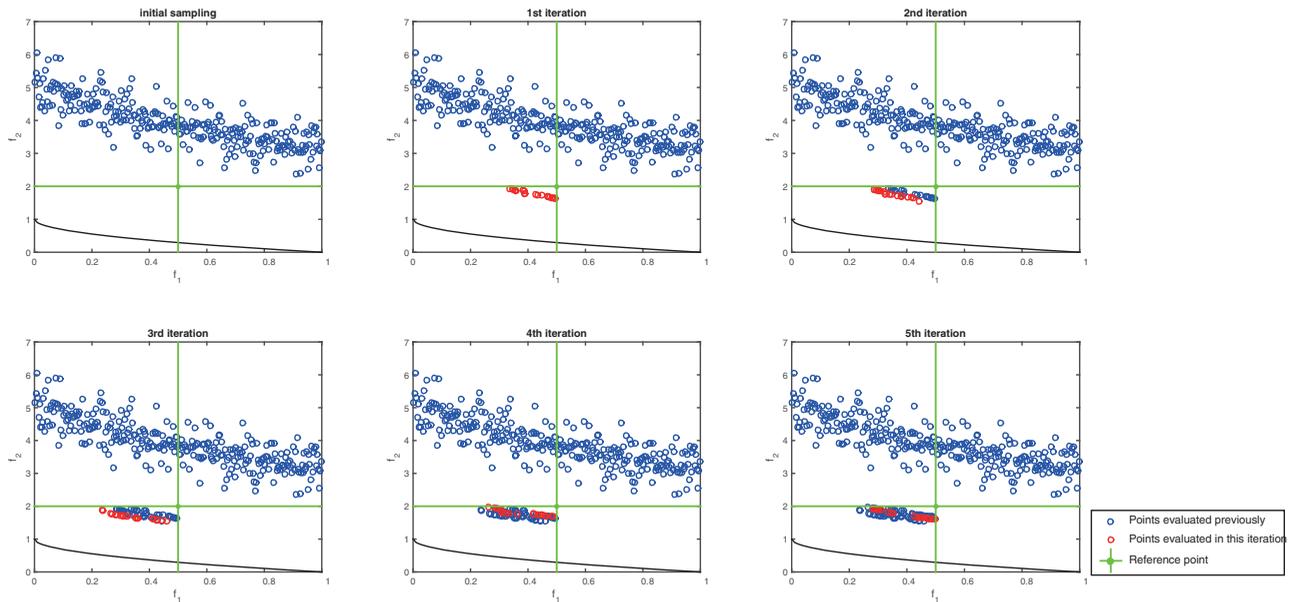


Figure 7. Multiobjective optimization procedure of ZDT1 test problem using WMO-ASMO (with weighted crowding distance).

ZDT2 and ZDT4 were sometimes NaN because only one Pareto optimal solution was found for some replications. The number of replications that yielded only one Pareto optimal solution is shown in brackets in the “mean of diversity” column of Table 5, after the mean value of the diversity metrics of other replications. Interestingly, the mean diversity of MO-ASMO is even better than that of NSGA-II and that of NSGA-II (large). This result may have been obtained because the individuals involved in NSGA-II become similar to each other during the evolution procedure, whereas MO-ASMO adaptively selects one point among very similar

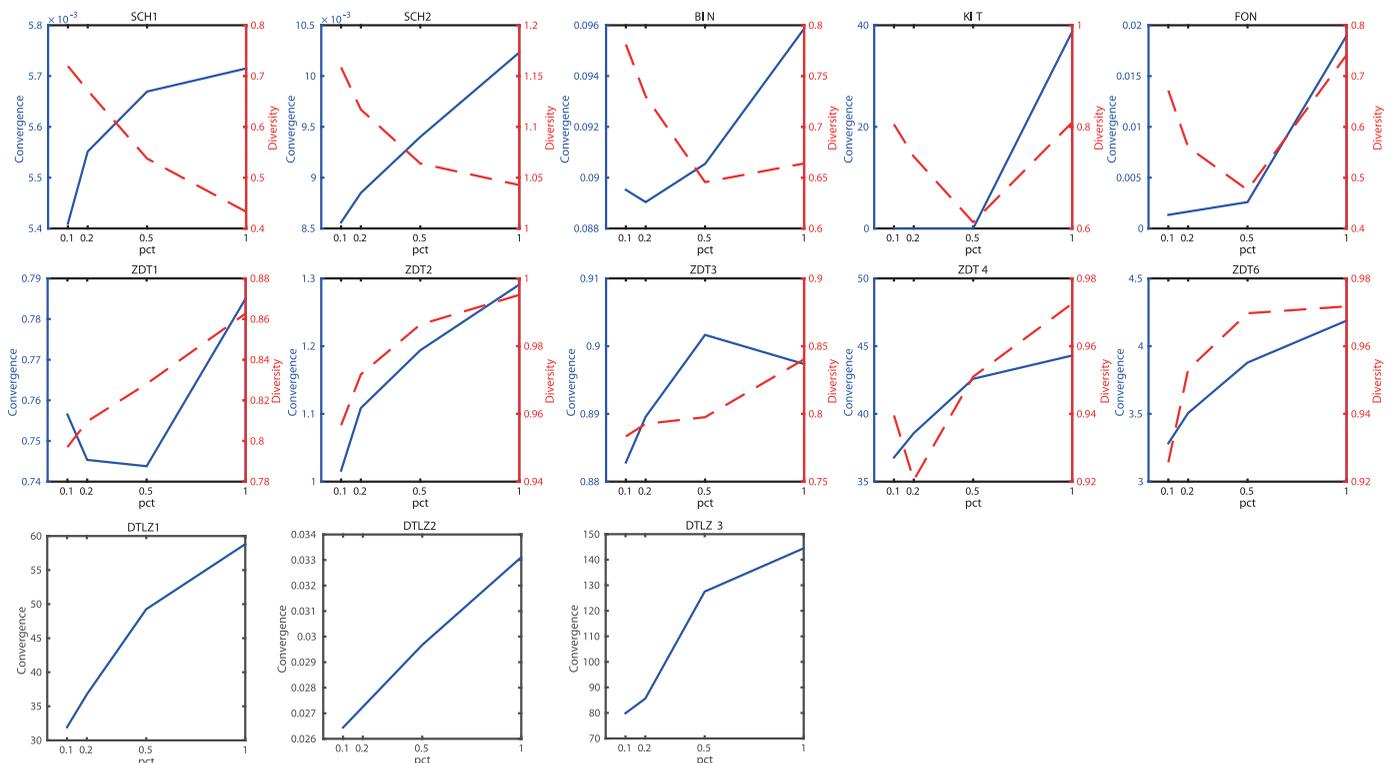


Figure 8. The influence of MO-ASMO's resampling percentage (pct) on the convergence and diversity metrics of test problems.

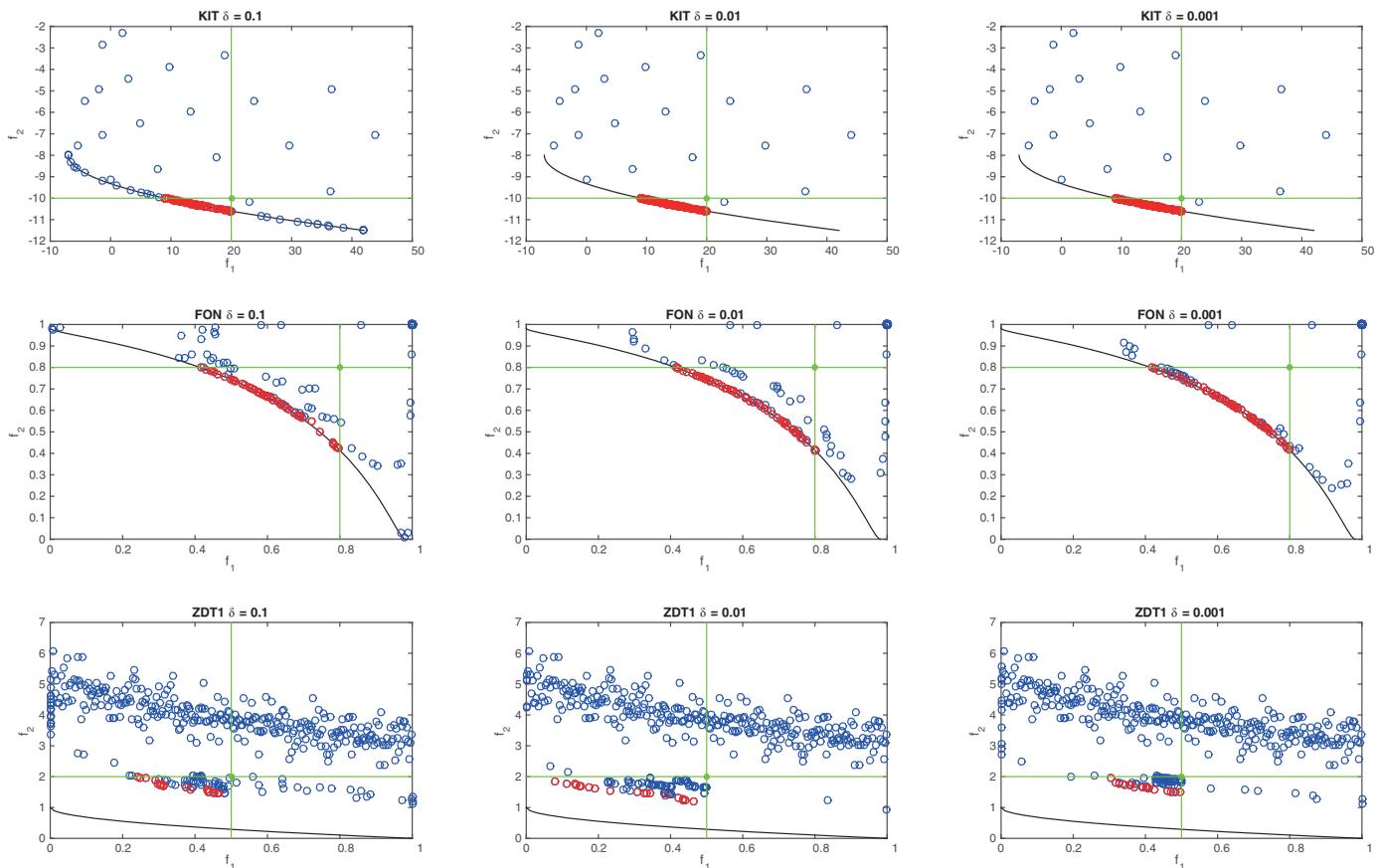


Figure 9. The influence of WMO-ASMO's weight factor δ on the optimal solutions of KIT, FON and ZDT1 test problems.

candidates. As shown in Figure 4, the SUMO toolbox could obtain optimal points that were very close to the true Pareto frontier, except for the toughest ZDT4 problem. However, the algorithm provided only one (ZDT2 and ZDT6), two (ZDT1 and ZDT4) or three (ZDT3) optimal points, which cannot represent the diversity of the Pareto frontier. The results indicated that SUMO is more aggressive in exploiting the global optimal and achieves better convergence metrics but fails to explore the space near the optimal to maintain diversity. The MO-ASMO algorithm can achieve better convergence than NSGA-II and better diversity than SUMO.

For the multiobjective DTLZ test problems, we only evaluated the convergence metric of NSGA-II, NSGA-II (large) and MO-ASMO because the diversity metric defined in **equation** (2) is not currently applicable to more than two objectives and the hypervolume-based Pol used in SUMO is very time consuming for more

Table 6. Screened Parameters for Multiobjective Optimization and Their Categories, Meanings and Ranges

No.	Parameter	Units	Category	Physical Meaning	Feasible Range
P2	hksati	mm/s	soil	maximum hydraulic conductivity	[0.001, 1]
P3	porsl		soil	porosity	[0.25, 0.75]
P4	phi0	mm	soil	minimum soil suction	[50, 500]
P6	bsw		soil	Clapp and Hornberger "b" parameter	[2.5, 7.5]
P17	sqrtdi	$m^{-0.5}$	canopy	the inverse of square root of leaf dimension	[2.5, 7.5]
P18	effcon	mol CO_2 /mol quanta	canopy	quantum efficiency of vegetation photosynthesis	[0.035, 0.35]
P19	vmax25	mol CO_2 /m ² s	canopy	maximum carboxylation rate at 25°C	[10^{-6} , 200^{-6}]
P23	trop		canopy	temperature coefficient of conductance-photosynthesis model	[250, 300]
P25	binter		canopy	intercept of conductance-photosynthesis model	[0.01, 0.04]
P30	ref(2,1)		canopy	NIR reflectance of living leaf	[0.35, 0.58]
P34	tran(2,1)		canopy	NIR transmittance of living leaf	[0.1, 0.3]
P36	z0m	m	canopy	aerodynamic roughness length	[0.05, 0.3]

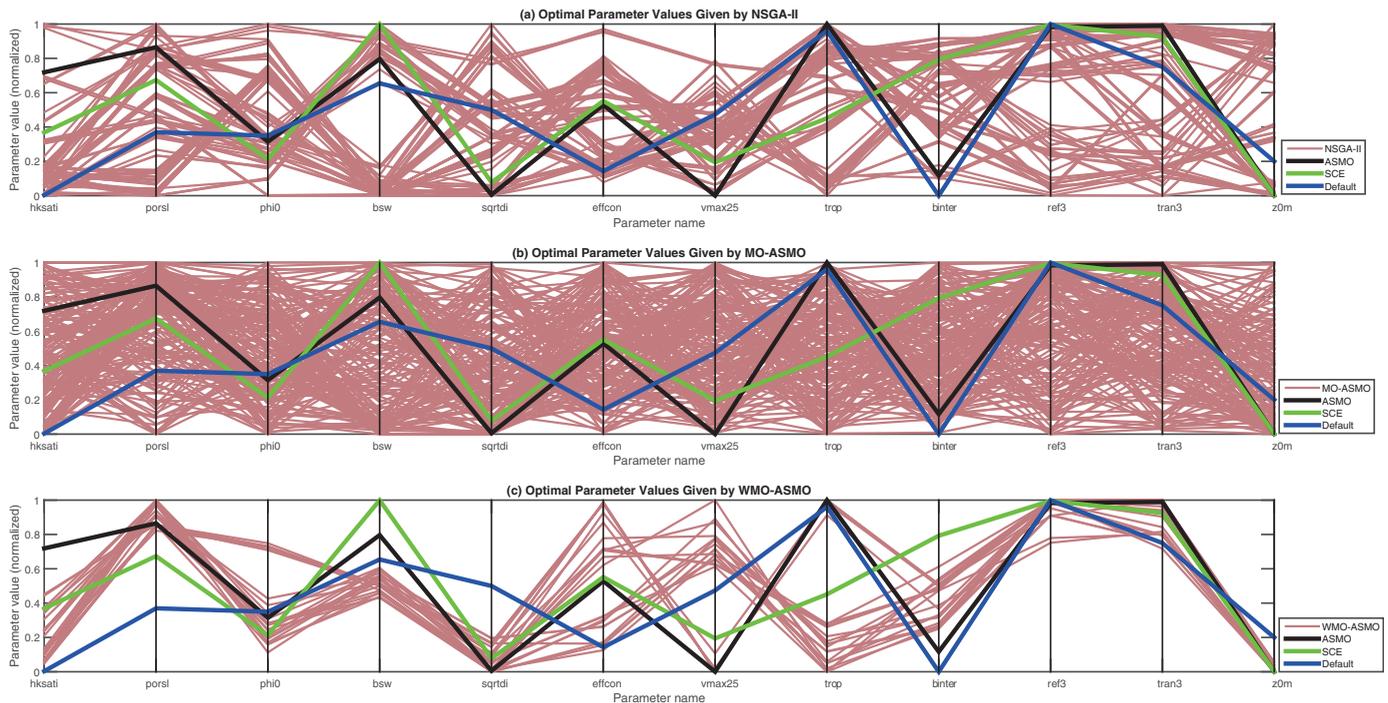


Figure 10. Multiobjective optimization results of CoLM: optimal parameter values given by NSGA-II, MO-ASMO and WMO-ASMO.

than two objectives. Although a fast calculation method for hypervolume-based PoI developed by *Couckuyt et al.* [2013] has been tested with DTLZ problems, computing the hypervolume-based PoI of a three-objective problems with 100 Pareto points will cost approximately 0.03 s, which is still too long because the computation must be repeated thousands to millions of times during the optimization procedure. As shown in Table 5, MO-ASMO could yield better convergence metrics than NSGA-II. Moreover, the convergence metrics of DTLZ1 and DTLZ2 given by MO-ASMO are also close to the results given by SUMO in *Couckuyt et al.* [2013], indicating that MO-ASMO is as effective as SUMO, and both of them are more efficient than NSGA-II.

For parameter optimization problems of large, complex geophysical models, we are more interested in finding the parameter sets that can improve all of the objectives than in those that can only improve some but degrade others. Consequently, we developed the “weighted crowding distance” to replace the “crowding distance” in NSGA-II and MO-ASMO. The MO-ASMO algorithm using the weighted crowding distance is referred to as WMO-ASMO. To evaluate the optimization procedure of WMO-ASMO, we selected three cases from previous test problems—KIT, FON, and ZDT1—and the reference points were set to [20, −10], [0.8 0.8], and [0.5, 2], respectively. The optimization procedures are shown in Figures 5–7, respectively. For the KIT problem, the true Pareto frontier located in the nondominated region of the reference point could be found during the first iteration, and in the following iterations, the algorithm continued to search within this region. For the more complex FON problem, the Pareto frontier could be reached in the fourth iteration. For the challenging ZDT1 problem, the Pareto frontier could not be reached over the limited number of iterations, but the obtained Pareto optimal points were located in the nondominated region of the reference point, indicating that the weighted crowding distance can steer the evolution procedure toward the right direction.

3.2. Optimal Use of (W)MO-ASMO

The (W)MO-ASMO algorithm has the following tunable meta-parameters: the population size (pop) and number of generations (gen) of the embedded NSGA-II, the resampling percentage (pct) and the weight factor δ in the weighted crowding distance. Pop and gen should be large enough to make sure the embedded NSGA-II can find the global Pareto optimal frontier. The total number of model runs (maxn) is limited by the available computational resources, and the maximum number of iterations (iter) is then determined

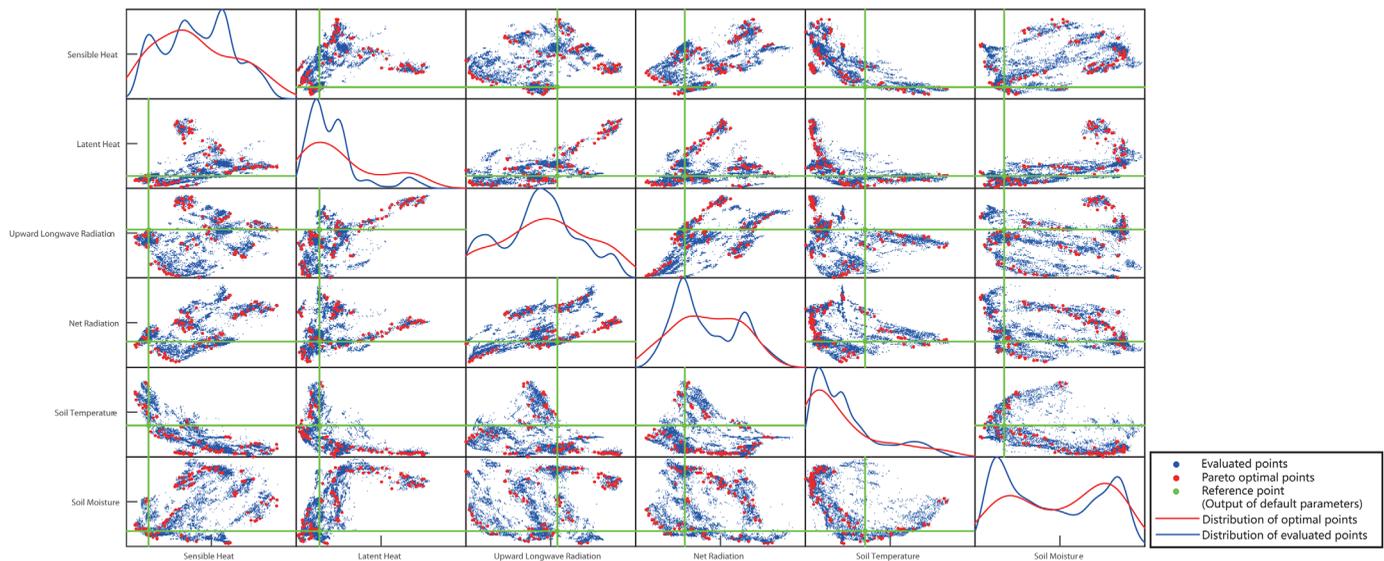


Figure 11. Multiobjective optimization results of CoLM: optimal objectives given by NSGA-II.

as: $iter = (maxn - initial\ sample\ size) / (pop * pct)$. Consequently, for MO-ASMO the user only needs to assign the value of pct , and for WMO-ASMO the weight factor δ as well. In this section we will show the results of test problems obtained with different meta-parameters pct and δ , and discuss how to determine their values.

First we evaluated the influence of pct . The convergence and diversity metrics of simple, ZDT and DTLZ test problems obtained by MO-ASMO with different pct values are presented in Figure 8. The value of pct was assigned to 0.1, 0.2, 0.5 and 1.0, and to give a fair comparison the total number of model runs was remain unchanged. Obviously the influence of pct is problem dependent. For most cases, the convergence metric increases with the growth of pct , except BIN, ZDT1 and ZDT3. For BIN and ZDT1, the convergence metric decreases first and then increases, whereas for ZDT3, increases first and then decreases. The influence to diversity metric is more complicated. For ZDT1, ZDT2, ZDT3 and ZDT6, the diversity metric is proportional

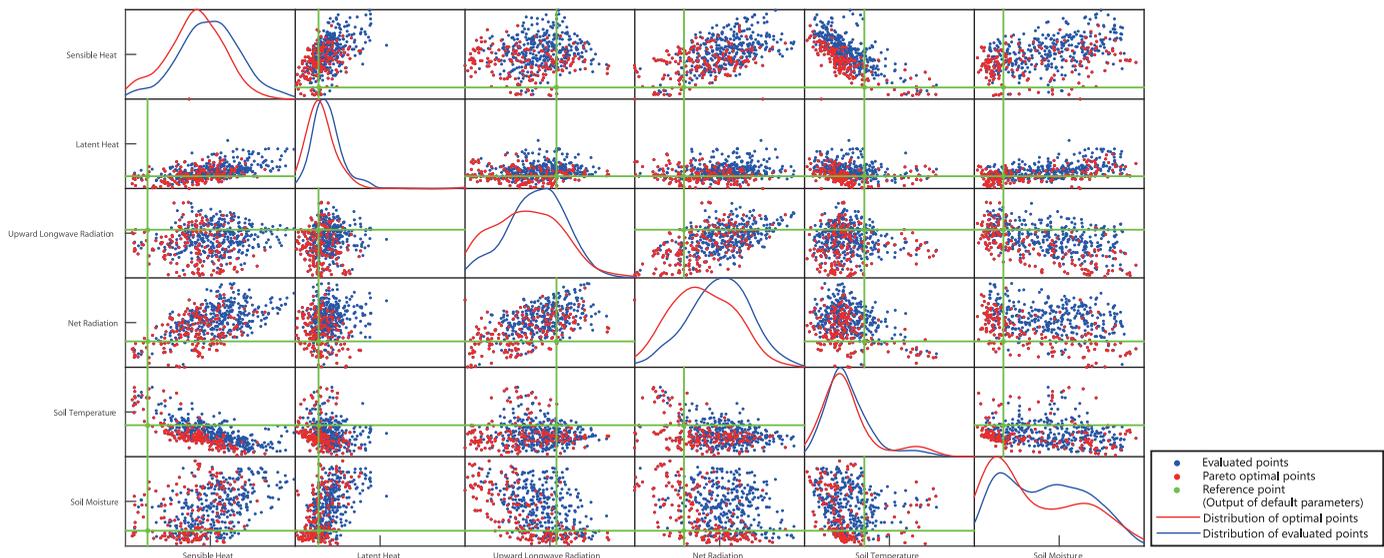


Figure 12. Multiobjective optimization results of CoLM: optimal objectives given by MO-ASMO.

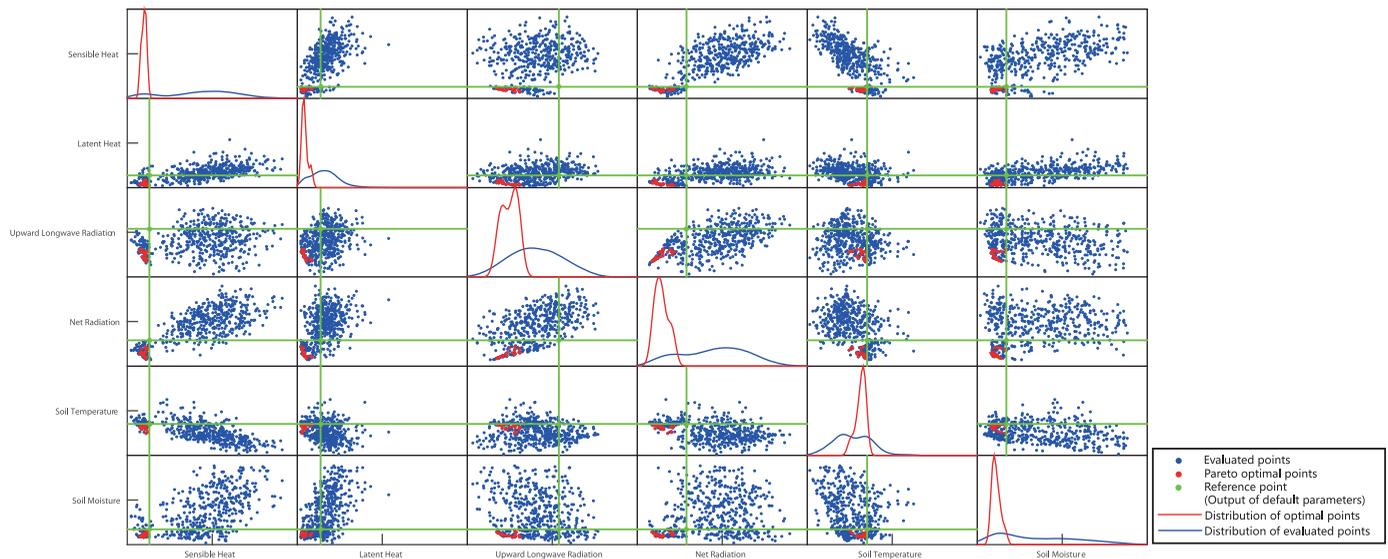


Figure 13. Multiobjective optimization results of CoLM: optimal objectives given by WMO-ASMO.

to pct, whereas for SCH1 and SCH2, inversely proportional to pct. For BIN, KIT, FON and ZDT4, the diversity metric decreases first and then increases with the growth of pct. Generally speaking, a smaller pct leads to better convergence, but for some problems there is a risk of losing diversity. Consequently, the users are suggested to begin with a small to medium value of pct, such as 0.1 or 0.2, to balance the performance of convergence and diversity, and increase pct if losing diversity.

To discuss the influence of weight factor, the optimal solutions of KIT, FON and ZDT1 obtained by WMO-ASMO with $\delta = 0.1, 0.01, 0.001$ are presented in Figure 9. A smaller weight factor means the search procedure is strongly driven toward the reference point, whereas larger weight factor allows searching other regions, such as the extreme optimal values of each objective. If the computational resources are very limited, a small value of weight factor is preferred in order to focus on the nondominated region of the reference point.

3.3. Evaluation of MO-ASMO on the CoLM Land Surface Model

To evaluate the effectiveness and efficiency of MO-ASMO for practical dynamic geophysical models, we used the Common Land Model (CoLM) developed by Dai *et al.* [2003, 2004] and Ji and Dai [2010], which can simulate biophysical, biochemical, ecological and hydrological dynamic processes and output energy-water transmission fluxes among soil, snow, vegetation and the atmosphere. We obtained freeze-thaw

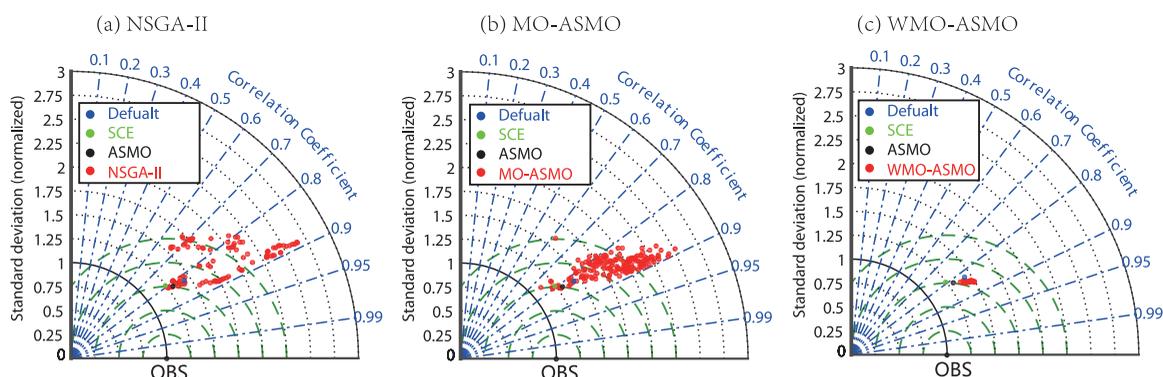


Figure 14. Taylor diagram of the Pareto optimal solution: sensible heat.

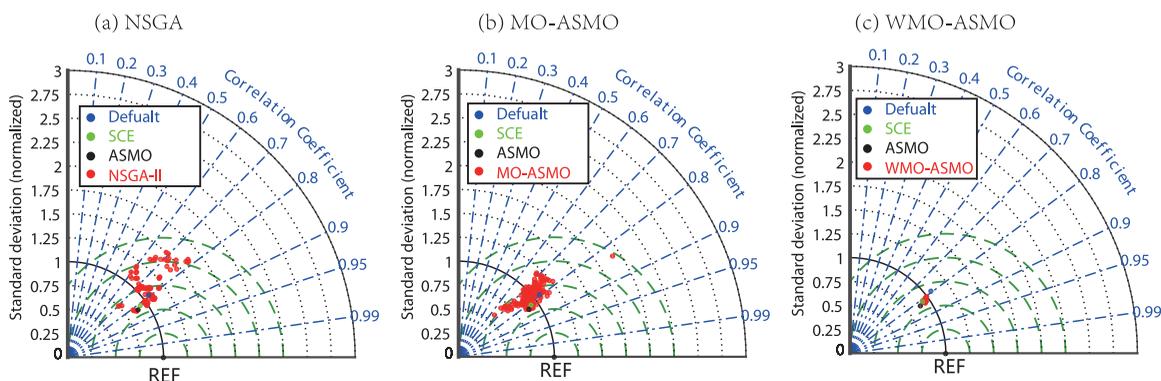


Figure 15. Taylor diagram of the Pareto optimal solution: latent heat.

observations from the A'rou station located in the Heihe River basin in northwest China (100°28'E, 38°03'N, altitude 3032.8 m above sea level). Seven forcing data sets of land surface modeling, namely, data pertaining to downward shortwave and longwave radiation, precipitation, air temperature, relative humidity, air pressure and wind speed [Hu et al., 2003], were provided by the Environmental & Ecological Science Data Center for West China (<http://westdc.westgis.ac.cn>), and six output data sets pertaining to sensible heat, latent heat, upward longwave radiation, net radiation, soil temperature and soil moisture were provided by Prof. Shaomin Liu at Beijing Normal University. In a previous study, we selected 40 adjustable parameters, defined six objective functions as the normalized root mean squared error (RMSE normalized by the means of data), screened the most sensitive parameters [Li et al., 2013] and performed parameter calibration [Gong et al., 2015a].

In the study by Gong et al. [2015a], multiobjective optimization was transformed into a single objective optimization with a weighting function. In this study, we conducted multiobjective optimization that could obtain the Pareto frontier in three different ways: NSGA-II, MO-ASMO and WMO-ASMO. The CoLM model setup was exactly the same as that reported by Gong et al. [2015a]. The 12 screened parameters and their units, meanings and ranges are shown in Table 6. For NSGA-II, both the population size and the number of generations were set to 100. For MO-ASMO and WMO-ASMO, 400 initial samples were used in the optimization, the maximum iteration number was set to 5, and the resample proportion was set to 0.2. The integrated NSGA-II featured 100 individuals evolved over 100 generations. Thus, the maximum numbers of CoLM evaluations of MO-ASMO and WMO-ASMO were $400 + 100 \times 0.2 \times 5 = 500$, whereas the maximum number of evaluations for NSGA-II was $100 \times 100 = 10,000$. The reference point of WMO-ASMO was set to the NRMSEs obtained by the default parameterization scheme [Gong et al., 2015a].

The Pareto optimal parameter values found by NSGA-II, MO-ASMO and WMO-ASMO are presented in Figure 10. The optimal parameter sets obtained by SCE-UA, ASMO and the default parameters are also plotted in the same figure. The optimal objective values provided by NSGA-II, MO-ASMO and WMO-ASMO are shown

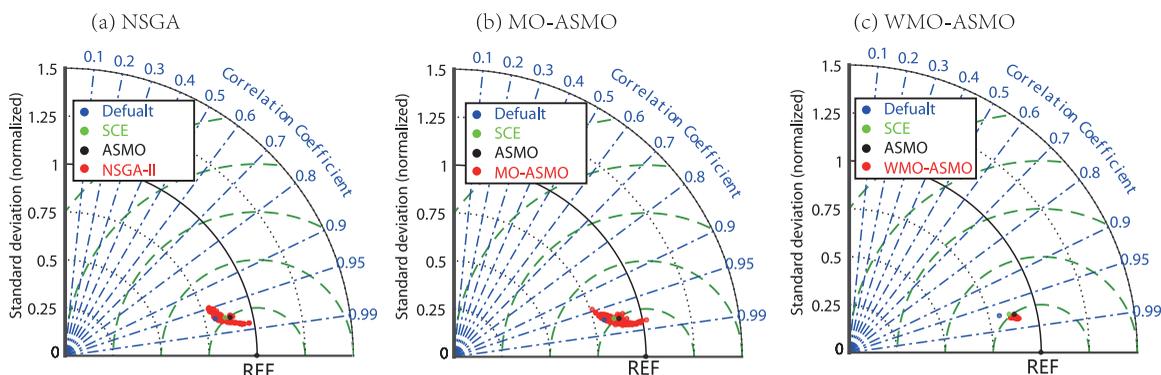


Figure 16. Taylor diagram of the Pareto optimal solution: upward longwave radiation.

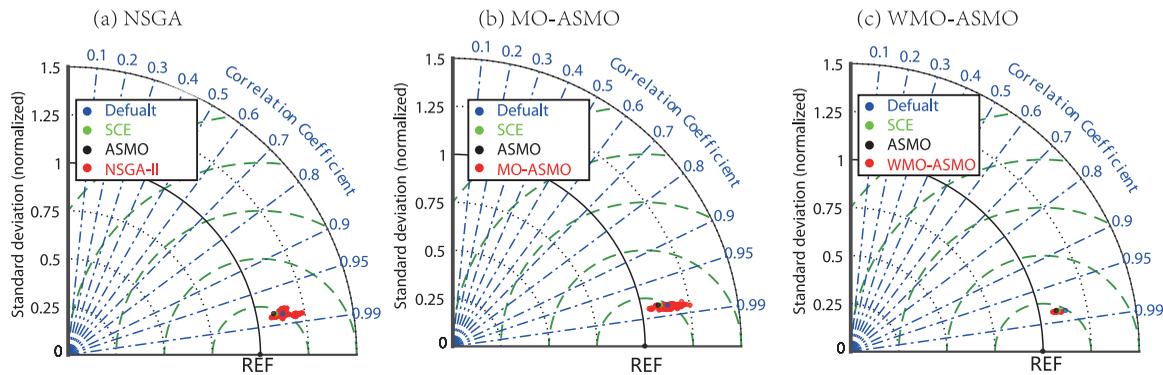


Figure 17. Taylor diagram of the Pareto optimal solution: net radiation.

in Figures 11–13 (red points), together with the points evaluated during optimization (blue points). Clearly, the distributions of the objective values and parameters given by the three methods are significantly different. The optimal objectives obtained by NSGA-II shows many detail structures of the Pareto frontier, and these details are provided by the very large number of evaluated points. For MO-ASMO, however, the distributions of objectives are similar but many details were lost, and the parameter distributions are also smoother than those of NSGA-II because NSGA-II used 10,000 original model runs to explore the details, whereas MO-ASMO only used 500 original model runs and used GPR surrogate model to predict the objectives in other regions. The surrogate model can mimic the behavior of the original model with only a few sample points, but it inevitably lost some details because of the limited number of samples. In the CoLM calibration case, many Pareto optimal solutions could improve one or two objectives but degraded others. These half-dominated solutions spread across the objective space, and describing their distribution would require a large number of original model runs. For WMO-ASMO, the search region was concentrated in the nondominated region of the reference point such that the limited number of original model runs performed could be concentrated to the region in which all of the objectives could be improved. The optimal parameters given by WMO-ASMO are also concentrated in a relatively small region compared with the distributions observed for the other two methods.

Furthermore, as shown in the Taylor diagrams [Taylor, 2001] in Figures 14–19, the optimal objectives yielded by MO-ASMO and NSGA-II are quite similar. The results indicate that MO-ASMO is as effective as NSGA-II and is much more efficient because it is rendered less computationally expensive by using fewer CoLM evaluations. Because the Pareto frontier of CoLM includes a vast half-dominated region in which only some of the objectives are improved, leaving others deteriorated, we must use WMO-ASMO, which uses weighted crowding distance to concentrate the search direction toward a small region that can simultaneously improve all of the objectives. As shown in the Taylor diagram, the vast half-dominated region was trimmed, leaving only the elite nondominated region. For all six evaluated fluxes, the variation, correlation coefficient, and RMSE could be more or less improved by the optimal solutions provided by the WMO-ASMO.

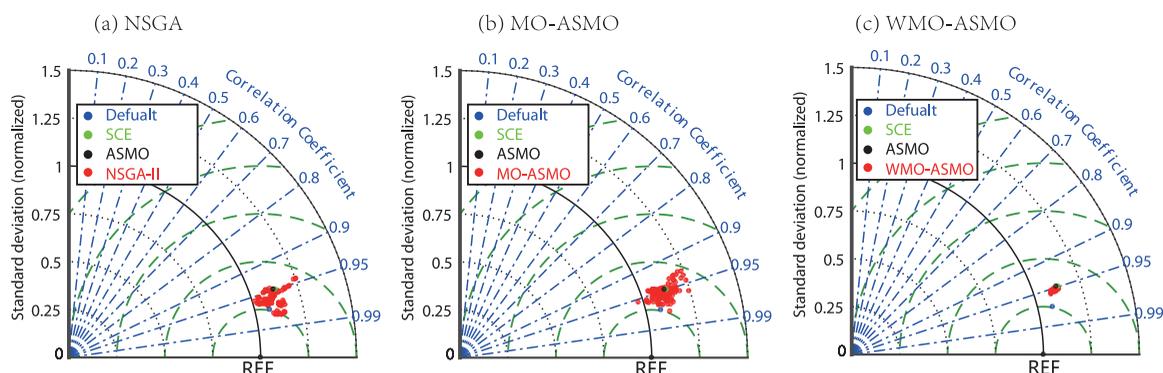


Figure 18. Taylor diagram of the Pareto optimal solution: soil temperature.

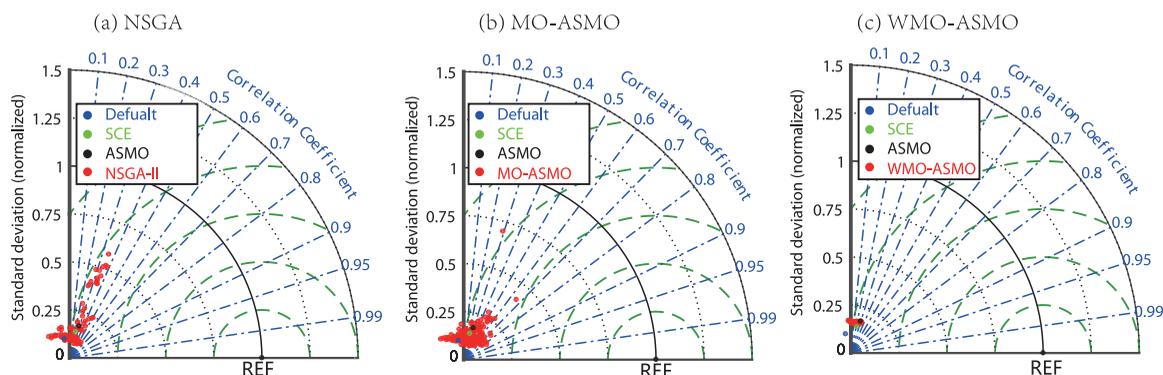


Figure 19. Taylor diagram of the Pareto optimal solution: soil moisture.

It is also worth noting the limitation of parameter optimization. As shown in Figure 19, the soil moisture simulation results yielded by CoLM with the original parameterization scheme are not satisfactory, and both the multiobjective optimization (NSGA-II, MO-ASMO and WMO-ASMO) and single-objective optimization (SCE-UA and ASMO) algorithms could not significantly improve the results. As discussed by Gong *et al.* [2015a], simulating freezing/thawing processes is one of the most challenging tasks in land surface modeling, especially for the A'rou station, where the freezing and thawing cycles are very fast in the top layer. To date, we still lack knowledge of the details of the freezing/thawing processes. To substantially improve the soil moisture simulation, it is necessary to thoroughly understand the physical processes of frozen soil, revise the program of land surface model and validate it in different areas. Parameter optimization is effective only if the model structure is consistent with the physical processes involved and the initial/boundary condition data set and surface data sets are of good quality. Optimization can help improve model performance and determine the most suitable parameter values, but it cannot compensate for the effects of model structure and data quality.

4. Discussion and Conclusions

In this study, we proposed a multiobjective optimization algorithm for expensive large, complex dynamic geophysical models, MO-ASMO, and a variation thereof, WMO-ASMO, which uses weighted crowding distance. In comparing 13 test problems, MO-ASMO showed similar effectiveness but was much more efficient than the classical evolution method NSGA-II. For most test problems, MO-ASMO yielded similar results with no more than 10% of the original model runs compared with NSGA-II. For the land surface model CoLM, MO-ASMO yielded a similar Pareto optimal with only 5% of the original model runs used by NSGA-II. We also incorporated the surrogate-based multiobjective optimization toolbox SUMO, which uses hypervolume-based PoI to transform multiobjective problems into single-objective problems. In our comparison, SUMO was more aggressive in finding the Pareto optimal and obtaining better convergence metrics but failed to maintain the diversity of optimal solutions, whereas MO-ASMO was more balanced in maintaining both convergence and diversity. In the comparison reported by Tsoukalas and Makropoulos [2015], SUMO outperformed SmsEGO and ParEGO, but the authors' test problem involved only four parameters and two objectives, which is too small a system to provide a more general conclusion. In this study, we found that for ZDT problems featuring more than 10 adjustable parameters, SUMO could only find 1, 2, or 3 Pareto optimals, implying that the algorithm does not have a robust mechanism for maintaining diversity, such as the crowding distance used in NSGA-II and MO-ASMO. Despite having used 13 various test problems, we admit that this comparative study was still very limited because only three algorithms were involved, and the algorithm setups were also not sufficiently diverse. In performing comprehensive comparisons of various types of optimization methods, the community should develop a common framework online that includes many components of the existing optimization algorithms and use various canonical real-world applications to test these algorithms [Maier *et al.*, 2014]. Toward this long-term target, we have developed an open-source software framework called UQ-PyL that can be deployed on both personal computers and super-computers and allows users to test their own problem with various types of optimization algorithms.

To sum up, compared with other similar methods, the novel contribution of (W)MO-ASMO lies in the following aspects:

1. *Initial sampling*: A uniform sampling design is applied for initial sampling. In the study by Gong *et al.* [2015b], the Good Lattice Points method with RGS de-correlation was shown to be one of the most uniform sampling methods. A more uniform initial sampling can potentially improve the convergence speed of optimization.
2. *Surrogate modeling*: The Gaussian Processes Regression method is adopted as the surrogate model. In various previous studies, similar methods such as Kriging interpolation and Radial Basis Functions have been successfully used in surrogate-based optimization. GPR is a flexible nonlinear regression method that can mimic their behavior by selecting a corresponding covariance function. In this study, we used the general-purpose Matérn covariance function. The hyper-parameters of the covariance function significantly affect the performance of GPR. To adaptively choose the hyper-parameters, we used the SCE-UA optimization method to adaptively maximize the marginal likelihood function. We also applied reinforcement learning technologies to save time when training the GPR.
3. *Adaptive sampling*: To effectively use the information in the surrogate model, we applied NSGA-II optimization to the GPR surrogate model for 100 generations to sufficiently explore the Pareto frontier and select a portion of (i.e., 20% in this paper) most representative subsets of the Pareto optimal solutions to run the original dynamic model. Crowding distance was adopted as a metric of the representativeness of optimal points [Deb *et al.*, 2002]. A large distance indicates that a point is far from other optimal solutions and thus more representative of its local region. In this manner, we could reduce the total number of dynamic model runs while maintaining the diversity of the population.
4. *Weighted crowding distance*: The goal of multiobjective optimization is to find a sufficiently large point set to represent the entire nondominating Pareto frontier. However, for certain real-world applications, such as the use of geophysical dynamic models, e.g., land surface models, weather and climate models, the goal is to find parameter sets that can improve all objectives relative to the default parameterization scheme, and other half-dominated solutions should be trimmed from the optimal solution set. The weighted crowding distance, which can replace the crowding distance in NSGA-II, assigns a small positive weight factor to the points in the half-dominated region to make them more likely to be eliminated by fast nondominated sorting. By using the weighted crowding distance, the WMO-ASMO can focus on the nondominated region of the reference point without wasting valuable computational resources on the half-dominated region, in which we have no interest. In the case study of CoLM, WMO-ASMO was able to find the Pareto optimal under the limit of 500 original model runs. Compared with the single-objective ASMO, WMO-ASMO can produce an ensemble parameter set in which all of the members are better than those under the default parameterization with respect to all objectives. Such an ensemble set will be very useful in producing ensemble hydro-meteorology forecasts because each member of the ensemble is better than in the default parameterization, and the members can also represent the diversity of possible future scenarios.

Acknowledgments

This research is supported by the National Basic Research Program of China (No. 2015CB953703), the Natural Science Foundation of China (No. 51309011 and No. 41375139), the Strategic Priority Research Program for Space Sciences (No. XDA04061200) of the Chinese Academy of Sciences, the Special Fund for Meteorological Scientific Research in Public Interest (GYHY201506002, CRA-40: the 40-year CMA global atmospheric reanalysis), and the Fundamental Research Fund for the Central Universities – Beijing Normal University Research Fund (No. 2013YB47). Special thanks to “Environmental & Ecological Science Data Center for West China, National Natural Science Foundation of China” (<http://westdc westgis.ac.cn>) for providing the meteorological forcing data and to the group of Prof. Shaomin Liu at the State Key Laboratory of Remote Sensing Science, School of Geography and Remote Sensing Science of Beijing Normal University for providing the surface flux validation data. The forcing data is available from Data Center for West China (<http://westdc westgis.ac.cn>), the surface flux validation data is available from Prof. Shaomin Liu (smliu@bnu.edu.cn), the CoLM code (version 2005) is available from the homepage of Prof. Yongjiu Dai, GCESS, BNU (<http://globalchange.bnu.edu.cn/research/models/>), and the source code of MO-ASMO and simulation/optimization results can be accessed from the first author upon request.

References

- Binh, T. T. (1999), A multiobjective evolutionary algorithm: The study cases, paper presented at The 1999 Genetic and Evolutionary Computation Conference, International Society for Genetic Algorithms, Inc. and Genetic Programming Conferences Inc., Orlando, Fla.
- Binh, T. T., and U. Korn (1997), MOBES: A multiobjective evolution strategy for constrained optimization problems, paper presented at The Third International Conference on Genetic Algorithms, International Society for Genetic Algorithms, Inc., Brno, Czech Republic.
- Breiman, L. (2001), Random forests, *Mach. Learning*, 45(1), 5–32.
- Corne, D. W., N. R. Jerram, J. D. Knowles, and M. J. Oates (2001), PESA-II: Region-based selection in evolutionary multiobjective optimization, paper presented at Genetic and Evolutionary Computation Conference, GECCO-2001, San Francisco.
- Couckuyt, I., D. Deschrijver, and T. Dhaene (2012), Towards efficient multiobjective optimization: multiobjective statistical criteria, paper presented at 2012 IEEE Congress on Evolutionary Computation (CEC), IEEE Computational Intelligence Society, Brisbane, Australia.
- Couckuyt, I., D. Deschrijver, and T. Dhaene (2013), Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization, *J. Global Optim.*, 60(3), 575–594.
- Cybenko, G. (1989), Approximation by superpositions of a sigmoidal function, *Math. Control Signals Syst.*, 2(4), 303–314.
- Dai, Y. J., et al. (2003), The Common Land Model, *Bull. Am. Meteorol. Soc.*, 84(8), 1013–1023.
- Dai, Y. J., R. E. Dickinson, and Y. P. Wang (2004), A two-big-leaf model for canopy temperature, photosynthesis, and stomatal conductance, *J. Clim.*, 17(12), 2281–2299.
- Deb, K. (2001), *Multi-Objective Optimization Using Evolutionary Algorithms*, 518 pp., John Wiley, Chichester, U. K.
- Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (2002), A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.*, 6(2), 182–197.
- Deb, K., L. Thiele, M. Laumanns, and E. Zitzler (2005), Scalable test problems for evolutionary multiobjective optimization, in *Evolutionary Multiobjective Optimization*, edited by A. Abraham, L. Jain, and R. Goldberg, pp. 105–145, Springer, London, U. K.

- Di, Z., Q. Duan, W. Gong, C. Wang, Y. Gan, J. Quan, J. Li, A. Ye, C. Miao, and C. Tong (2015), Assessing WRF model parameter sensitivity: A case study with 5 day summer precipitation forecasting in the Greater Beijing Area, *Geophys. Res. Lett.*, *42*, 579–587, doi:10.1002/2014GL061623.
- di Pierro, F., S.-T. Khu, D. Savić, and L. Berardi (2009), Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms, *Environ. Modell. Software*, *24*(2), 202–213.
- Duan, Q. Y., S. Sorooshian, and V. K. Gupta (1992), Effective and efficient global optimization for conceptual rainfall-runoff models, *Water Resour. Res.*, *28*(4), 1015–1031.
- Duan, Q. Y., V. K. Gupta, and S. Sorooshian (1993), Shuffled complex evolution approach for effective and efficient global minimization, *J. Optim. Theory Appl.*, *76*(3), 501–521.
- Emmerich, M., N. Beume, and B. Naujoks (2005), An EMO algorithm using the hypervolume measure as selection criterion, in *Evolutionary Multi-Criterion Optimization*, edited by C. Coello Coello, A. Hernández Aguirre, and E. Zitzler, pp. 62–76, Springer, Berlin.
- Emmerich, M. T. M., K. C. Giannakoglou, and B. Naujoks (2006), Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodelling, *IEEE Trans. Evol. Comput.*, *10*(4), 421–439.
- Fang, K. (1980), The uniform design: Application of number-theoretic methods in experimental design, *Acta Math. Appl. Sin.*, *3*(4), 363–372.
- Fleischer, M. (2003), The measure of Pareto optima applications to multi-objective metaheuristics, in *Evolutionary Multi-Criterion Optimization*, edited by C. Fonseca et al., pp. 519–533, Springer, Berlin.
- Fonseca, C. M., and P. J. Fleming (1993), Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, paper presented at The Fifth International Conference on Genetic Algorithms, International Society for Genetic Algorithms, Inc., Univ. of Ill. at Urbana-Champaign, Champaign.
- Fonseca, C. M., and P. J. Fleming (1998a), Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part II: Application example, *IEEE Trans. Syst. Man Cybernetics Part A Syst. Hum.*, *28*(1), 38–47.
- Fonseca, C. M., and P. J. Fleming (1998b), Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part I: A unified formulation, *IEEE Trans. Syst. Man Cybernetics Part A Syst. Hum.*, *28*(1), 26–37.
- Friedman, J. H. (1991), Multivariate adaptive regression splines, *Ann. Stat.*, *19*(1), 1–14.
- Gong, W., Q. Duan, J. Li, C. Wang, Z. Di, Y. Dai, A. Ye, and C. Miao (2015a), Multi-objective parameter optimization of common land model using adaptive surrogate modeling, *Hydrol. Earth Syst. Sci.*, *19*(5), 2409–2425.
- Gong, W., Q. Y. Duan, J. D. Li, C. Wang, Z. H. Di, A. Z. Ye, C. Y. Miao, and Y. J. Dai (2015b), An intercomparison of sampling methods for uncertainty quantification of environmental dynamic models, *J. Environ. Informatics*, doi:10.3808/jei.201500310, in press.
- Gorissen, D. (2010), *Grid-enabled Adaptive Surrogate Modeling for Computer Aided Engineering*, Ghent Univ., Ghent, Belgium.
- Halton, J. H. (1964), Algorithm 247: Radical-inverse quasi-random point sequence, *Commun. ACM*, *7*(12), 701–702.
- Horn, J., N. Nafpliotis, and D. E. Goldberg (1994), A niched Pareto genetic algorithm for multiobjective optimization, paper presented at 1994 IEEE World Congress on Computational Intelligence, IEEE Computational Intelligence Society, Orlando, Fla.
- Hu, Z. Y., M. G. Ma, R. Jin, W. Z. Wang, G. H. Huang, Z. H. Zhang, and J. L. Tan (2003), *WATER: Dataset of Automatic Meteorological Observations at the A'rou Freeze/Thaw Observation Station*, Cold and Arid Reg. Environ. and Eng. Res. Inst., Chin. Acad. of Sci., Lanzhou, China.
- Iman, R. L., and W. J. Conover (1982), A distribution-free approach to inducing rank correlation among input variables, *Commun. Stat. Simul. Comput.*, *11*(3), 311–334.
- Ji, D., and Y. Dai (2010), *The Common Land Model (CoLM) Technical Guide*, GCESS, Beijing Normal Univ., Beijing, China.
- Jones, D. R. (2001), A taxonomy of global optimization methods based on response surfaces, *J. Global Optim.*, *21*(4), 345–383.
- Jones, D. R., M. Schonlau, and W. J. Welch (1998), Efficient global optimization of expensive black-box functions, *J. Global Optim.*, *13*(4), 455–492.
- Jourdan, L., D. W. Corne, D. A. Savić, and G. Walters (2006), LEMMO: Hybridising rule induction and NSGA II for Multi-Objective Water Systems design, paper presented at The Eighth International Conference on Computing and Control for the Water Industry, Univ. of Exeter, Exeter, U. K.
- Kita, H., Y. Yabumoto, N. Mori, and Y. Nishikawa (1996), Multi-objective optimization by means of the thermodynamical genetic algorithm, in *Parallel Problem Solving From Nature—PPSN IV*, edited by H.-M. Voigt et al., pp. 504–512, Springer, Berlin.
- Knowles, J. (2006), ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems, *IEEE Trans. Evol. Comput.*, *10*(1), 50–66.
- Knowles, J., and D. Corne (1999), The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation, paper presented at 1999 Congress on Evolutionary Computation CEC 99, Washington, D. C., 1 Jan.
- Korobov, N. M. (1959a), The approximate computation of multiple integrals, *Dokl. Akad. Nauk. SSSR*, *124*, 1207–1210.
- Korobov, N. M. (1959b), Computation of multiple integrals by the method of optimal coefficients, *Vestnik Moskov Univ. Sec. Math. Astr. Fiz. Him.*, *4*, 19–25.
- Kourakos, G., and A. Mantoglou (2013), Development of a multi-objective optimization algorithm using surrogate models for coastal aquifer management, *J. Hydrol.*, *479*, 13–23.
- Li, J., Q. Y. Duan, W. Gong, A. Ye, Y. Dai, C. Miao, Z. Di, C. Tong, and Y. Sun (2013), Assessing parameter importance of the Common Land Model based on qualitative and quantitative sensitivity analysis, *Hydrol. Earth Syst. Sci.*, *17*(8), 3279–3293.
- Li, M., D. Yang, J. Chen, and S. S. Hubbard (2012), Calibration of a distributed flood forecasting model with input uncertainty using a Bayesian framework, *Water Resour. Res.*, *48*, W08510, doi:10.1029/2010WR010062.
- Liu, Y. Q., H. V. Gupta, S. Sorooshian, L. A. Bastidas, and W. J. Shuttleworth (2004), Exploring parameter sensitivities of the land surface using a locally coupled land-atmosphere model, *J. Geophys. Res.*, *109*, D21101, doi:10.1029/2004JD004730.
- Liu, Y. Q., H. V. Gupta, S. Sorooshian, L. A. Bastidas, and W. J. Shuttleworth (2005), Constraining land surface and atmospheric parameters of a locally coupled model using observational data, *J. Hydrometeorol.*, *6*(2), 156–172.
- Maier, H. R., et al. (2014), Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions, *Environ. Modell. Software*, *62*, 271–299.
- McKay, M. D., R. J. Beckman, and W. J. Conover (1979), 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.
- Michalski, R. S. (2000), Learnable Evolution Model: Evolutionary processes guided by machine learning, *Mach. Learning*, *38*(1–2), 9–40.
- Nain, P. K. S., and K. Deb (2005), *A Multi-Objective Optimization Procedure With Successive Approximate Models*, Indian Inst. of Technol., Kanpur, India.
- Neelin, J. D., A. Bracco, H. Luo, J. C. McWilliams, and J. E. Meyerson (2010), Considerations for parameter optimization and sensitivity in climate models, *Proc. Natl. Acad. Sci. U. S. A.*, *107*(50), 21,349–21,354.
- Owen, A. B. (1994), Controlling correlations in Latin hypercube samples, *J. Am. Stat. Assoc.*, *89*(428), 1517–1522.

- Pilát, M., and R. Neruda (2013), Surrogate model selection for evolutionary multiobjective optimization, paper presented at 2013 IEEE Congress on Evolutionary Computation (CEC), IEEE Computational Intelligence Society, Cancun, Mexico.
- Ponweiser, W., T. Wagner, D. Biermann, and M. Vincze (2008), Multiobjective optimization on a limited budget of evaluations using model-assisted S-metric selection, in *Parallel Problem Solving From Nature—PPSN X*, edited by G. Rudolph et al., pp. 784–794, Springer, Berlin.
- Rasmussen, C. E., and C. K. I. Williams (2006), *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, Mass.
- Razavi, S., B. A. Tolson, and D. H. Burn (2012), Review of surrogate modeling in water resources, *Water Resour. Res.*, *48*, W07401, doi: 10.1029/2011WR011527.
- Schaffer, J. D. (1987), Multiple objective optimization with vector evaluated genetic algorithms, paper presented at ICGA, Lawrence Erlbaum Assoc., Hillsdale, N. J.
- Sobol', I. M. (1967), On the distribution of points in a cube and the approximate evaluation of integrals, *USSR Comput. Math. Math. Phys.*, *7*(4), 86–112.
- Syberfeldt, S., H. Grimm, A. Ng, and R. I. John (2008), A parallel surrogate-assisted multi-objective evolutionary algorithm for computationally expensive optimization problems, paper presented at 2008 IEEE Congress on Evolutionary Computation (CEC 2008), IEEE Computational Intelligence Society, Hongkong, China.
- Taylor, K. E. (2001), Summarizing multiple aspects of model performance in a single diagram, *J. Geophys. Res.*, *106*(D7), 7183–7192.
- Tsoukalas, I., and C. Makropoulos (2015), Multiobjective optimisation on a budget: Exploring surrogate modelling for robust multi-reservoir rules generation under hydrological uncertainty, *Environ. Modell. Software*, *69*, 396–413.
- Van Veldhuizen, D. A. (1999), *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, pp. 249–249, Air Force Inst. of Technol., Ann Arbor, Mich.
- Vapnik, V. N. (2002), *The Nature of Statistical Learning Theory*, 2nd ed., Springer, N. Y.
- Vrugt, J. A., H. V. Gupta, W. Bouten, and S. Sorooshian (2003a), A Shuffled Complex Evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters, *Water Resour. Res.*, *39*(8), 1201, doi:10.1029/2002WR001642.
- Vrugt, J. A., H. V. Gupta, L. A. Bastidas, W. Bouten, and S. Sorooshian (2003b), Effective and efficient algorithm for multiobjective optimization of hydrologic models, *Water Resour. Res.*, *39*(8), 1214, doi:10.1029/2002WR001746.
- Wang, C., Q. Duan, W. Gong, A. Ye, Z. Di, and C. Miao (2014), An evaluation of adaptive surrogate modeling based optimization with two benchmark problems, *Environ. Modell. Software*, *60*, 167–179.
- Wang, Y., and K. Fang (1981), A note on uniform distribution and experimental design, *KeXue TongBao*, *26*(6), 495–489.
- Ye, K. Q., W. Li, and A. Sudjianto (2000), Algorithmic construction of optimal symmetric Latin hypercube designs, *J. Stat. Plann. Inference*, *90*(1), 145–159.
- Zitzler, E., and L. Thiele (1998), Multiobjective optimization using evolutionary algorithms—A comparative case study, in *Parallel Problem Solving From Nature*, edited by A. E. Eiben et al., pp. 292–301, Springer, Berlin.
- Zitzler, E., K. Deb, and L. Thiele (2000), Comparison of multiobjective evolutionary algorithms: Empirical results, *Evol. Comput.*, *8*(2), 173–195.