Generalizing Surrogate-Assisted Evolutionary Computation

Dudy Lim, Yaochu Jin, *Senior Member, IEEE,* Yew-Soon Ong, *Member, IEEE,* and
Bernhard Sendhoff, *Senior Member, IEEE*

***Abstract*—Using surrogate models in evolutionary search provides an efficient means of handling today's complex applications plagued with increasing high-computational needs. Recent surrogate-assisted evolutionary frameworks have relied on the use of a variety of different modeling approaches to approximate the complex problem landscape. From these recent studies, one main research issue is with the choice of modeling scheme used, which has been found to affect the performance of evolutionary search significantly. Given that theoretical knowledge available for making a decision on an approximation model *a priori* is very much limited, this paper describes a generalization of surrogate-assisted evolutionary frameworks for optimization of problems with objectives and constraints that are computationally expensive to evaluate. The generalized evolutionary framework unifies diverse surrogate models synergistically in the evolutionary search. In particular, it focuses on attaining reliable search performance in the surrogate-assisted evolutionary framework by working on two major issues: 1) to mitigate the *'curse of uncertainty'* robustly, and 2) to benefit from the '*bless of uncertainty*.' The backbone of the generalized framework is a surrogate-assisted memetic algorithm that conducts simultaneous local searches using *ensemble* and *smoothing* surrogate models, with the aims of generating reliable fitness prediction and search improvements simultaneously. Empirical study on commonly used optimization benchmark problems indicates that the generalized framework is capable of attaining reliable, high quality, and efficient performance under a limited computational budget.**

***Index Terms*—Approximation models, computationally expensive problems, memetic algorithms, metamodels, surrogate models, surrogate-assisted evolutionary algorithms.**

I. Introduction

OVER the years, evolutionary algorithms (EAs) have become one of the well-established optimization techniques, especially in the fields of art and design, business and finance, science and engineering. Many successful applications of EAs have been reported, ranging from music composition [1] to financial forecasting [2], aircraft design [3], job-shop scheduling [4], and drug design [5]. Although well established as credible and powerful optimization tools, researchers in this area are now facing new challenges of increasing computational needs by today's applications. For instance, a continuing trend in science and engineering is the use of increasingly high-fidelity accurate analysis codes in the design and simulation process. Modern computational structural mechanics (CSM), computational electro-magnetics (CEM), computational fluid dynamics (CFD) and first principle simulations have been shown to be reasonably accurate. Such analysis codes play a central role in the design process since they aid designers and scientists in validating new designs and studying the effect of altering key parameters on product and/or system performance. However, such moves may prove to be cost-prohibitive or impractical in the evolutionary design optimization process, leading to intractable design cycle times.

An intuitive way to reduce the search time of evolutionary optimization algorithms when dealing with computationally expensive solver, is the use of high-performance computing technologies and/or computationally efficient surrogate models. In recent years, there have been increasing research activities in the design of surrogate-assisted evolutionary frameworks for handling complex optimization problems with computationally expensive objective functions and constraints. In particular, since the modeling and design optimization cycle time is roughly proportional to the number of calls to the computationally expensive solver, many evolutionary frameworks have turned to the deployment of computationally cheap approximation models in the search to replace in part the original solvers [6]–[8]. Using approximation models also known as surrogates or metamodels, the computational burden can be greatly reduced since the efforts required to build the surrogates and to use them are much lower than those in the standard approach that directly couples the EA with the expensive solvers. Among the approximation models, polynomial regression (PR), also known as response surface methodology (RSM), support vector machine (SVM), artificial neural networks (ANNs), radial basis function (RBF), and Gaussian process (GP), also referred to as Kriging or design and analysis of computer experiment (DACE) models, are the most prominent and commonly used [9]–[11].

Manuscript received January 30, 2008; revised May 14, 2008. First version published December 11, 2009; current version published May 28, 2010. This paper was supported by Honda Research Institute Europe GmbH, Germany.

D. Lim is with Center for Computational Intelligence, School of Computer Engineering, Nanyang Technological University, 639798, Singapore (e-mail: dlim@ntu.edu.sg).

Y. Jin and B. Sendhoff are with Honda Research Institute Europe GmbH, Offenbach 63073, Germany (e-mail: yaochu.jin@honda-ri.de; bernhard.sendhoff@honda-ri.de).

Y. S. Ong is with the Division of Information Systems, School of Computer Engineering, Nanyang Technological University, 639798, Singapore. He is also with the Center for Computational Intelligence (C2I), Nanyang Technological University, 639798, Singapore (e-mail: asysong@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2009.2027359

In the context of EA, various approaches for working with computationally expensive problems using surrogate models have been reported. Early techniques include the use of fitness inheritance or imitation [12], [13], where the fitness of an individual is defined by either the parents or other individuals previously encountered along the search. Another common approach is to preselect a subset of individuals that would undergo exact function evaluations, while all others are predicted based on surrogate models. Some of the simple schemes introduced are based on random individual selection [14] or selecting the best/most promising individuals based on the predictions made by the surrogate models [7], [11], [15], [16]. Other schemes include identifying some cluster centers [17], [18], or uncertain individuals that are predicted to have poor estimates [19] as representatives that will undergo exact function evaluations subsequently. Such forms of model management schemes are termed as *'evolution control'* in [7] and [20]. An alternative approach adopted in [21] involves the refinement of the surrogate used from coarse-to-fine grained models as the search evolves. Online localized surrogate models are also deployed within the local search phase of memetic algorithms (MAs) [8], [22]. The synergy of online global and local surrogate in the memetic search was also investigated in [11]. To enhance the prediction accuracy of fitness predictions based on surrogates, the inclusion of gradient information in surrogate building was also studied in [23] and [24], independently. More recently, Schmidt and Lipson [25] proposed the use of coevolution technique to address issues such as level of approximation and accuracy of fitness predictors.

More recently, the idea of using surrogate to speed up evolutionary search process has found its way into the field of evolutionary multiobjective optimization (MOO). Many of the schemes introduced in the context of single-objective optimization (SOO) have been extended to their corresponding MOO variants. The Kriging-based surrogate-assisted evolutionary multiobjective algorithm in [26] represents an extension of the efficient global optimization framework [27] introduced for handling SOO problems, whereas [28] and [29] extended the coarse-to-fine grained approximation and preselection schemes to its MOO variants, respectively. The coevolution of genetic algorithms (GAs) for multiple objectives based on online surrogates was introduced in [30]. After some fixed search intervals, the surrogates produced that represent the different objectives are then exchanged and shared among multiple GAs. In [31], a multiobjective EA is run for a number of iterations on a surrogate model before the model is updated using exact evaluation from some selected points. For greater details on surrogate-assisted EAs for handling optimization problems with computationally expensive objective/constraint functions, the readers are referred to [9] and [32].

In spite of the extensive research efforts on this topic, existing surrogate-assisted evolutionary frameworks remains open for further improvement. Jin *et al.* [14] have shown that existing surrogate-assisted evolutionary frameworks proposed are often flawed by introduction of false optima since the parametric approximation technique used may not be capable of modeling the problem landscapes accurately, thus producing unreliable search. Generally, the *'curse of dimensionality'* creates significant difficulties in the construction of accurate surrogate models for fitness prediction. Further, recent studies have shown that the choice of approximation technique used affects the performance of evolutionary searches [33]. On the other hand, it is worth keeping in mind that approximation error in the surrogate model does not always harm. A surrogate model capable of smoothing the multimodal or noisy landscape of the complex problem may contribute more beneficially to the evolutionary search than one that models the original fitness function accurately. For instance, the study in [44] has emphasized the importance of predicting search improvement as opposed to the usual practice of improving only the quality of the surrogate in the context of evolutionary optimization. Based on these recent studies, it is worth highlighting the influence of the approximation method used on the performance of any surrogate-assisted evolutionary search. The greatest barrier to further progress is that, with so many approximation techniques available in the literature, it is almost impossible to know which is most relevant for modeling the problem landscape or generating reliable fitness predictions when one has only limited knowledge of its fitness space before the search starts. Moreover, approximation techniques by themselves may model differently on different problem landscapes. Depending on the complexity of a design problem, a single approximation model that may have proven to be successful in an instance might not work so well, or at all, on others. In the field of multidisciplinary optimization, such observations have also been reported [34]–[41]. In those studies, this issue is commonly handled by performing multiple optimization runs, each on different surrogate model or ensemble model. In [34], [35] and [39], a set of surrogate models consisting Kriging, PR, RBF, and weighted average ensemble is used to demonstrate that multiple surrogates can improve robustness of optimization at minimal cost. Similarly, [36] uses PR and RBF surrogate models in the context of multiobjective optimization and shows that each of the models performs better at different region of the Pareto front. Others in [37], [38], [40] and [41] resolve this issue by introducing various ensemble model building techniques. It is shown from these studies that ensemble models generally outperform most of the individual surrogates.

This paper introduces a generalized framework for unifying diverse surrogate models synergistically in the evolutionary search. In contrast to existing efforts, we focus on predicting search improvement in the context of optimization as opposed to solely on improving the prediction quality of the approximation. In particular, we generalize the problem to attain reliable search improvement in surrogate-assisted evolutionary framework as two major goals: 1) to mitigate the *'curse of uncertainty'*; 2) to benefit from the *'bless of uncertainty'*. The *'curse of uncertainty'*[1] refers to the negative consequences introduced by the approximation error of the surrogate models

[1] In the present context, the definition of 'uncertainty' refers to the approximation errors in the fitness function due to the use of surrogate models based on the definitions given in [45].

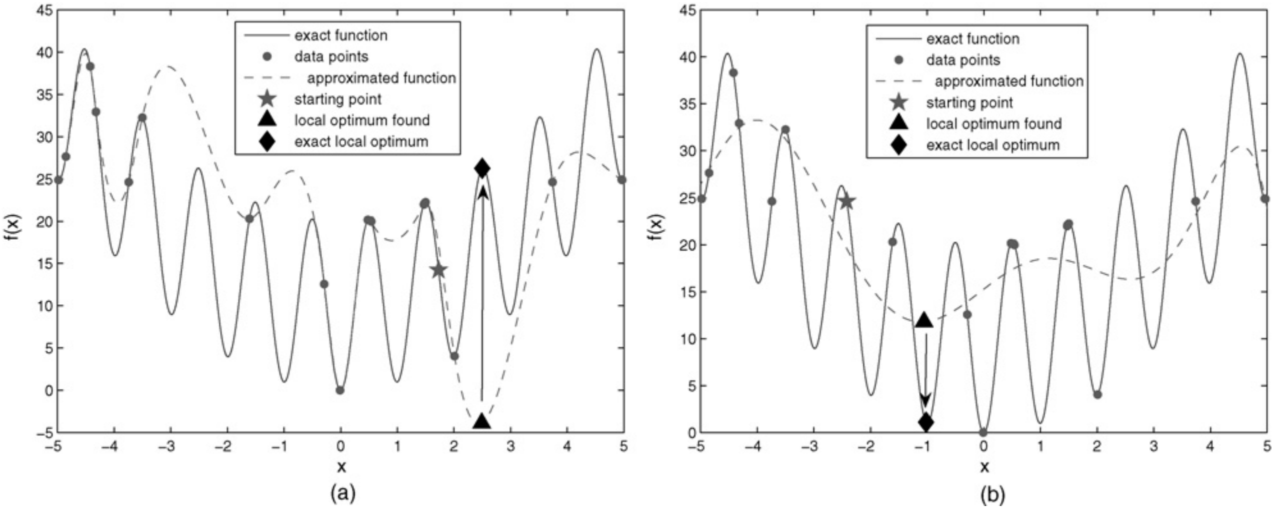


Fig. 1. Curse and bless of uncertainty in single-objective EA using surrogates. (a) *'Curse of uncertainty'* in single-objective EA using surrogates. Approximated function in the figure is obtained using spline interpolation technique. (b) *'Bless of uncertainty'* in single-objective EA using surrogates. Approximated function in the figure is obtained using a low order polynomial regression.

used. On the other hand, *'bless of uncertainty'* refers to the benefits attained by the use of surrogate models. Particularly, we seek for surrogate models that are capable of generating reliable fitness predictions on diverse problems of different landscapes to mitigate the *'curse of uncertainty'* on one hand, and on the other hand surrogate models that are capable of smoothing rugged fitness landscapes to prevent the search from getting stuck in local optima [44]. Previous studies by Yao *et al.* [42], [43] have also confirmed that smoothed landscape of rugged fitness landscape can lead the search to optimum solutions easier than using the exact fitness landscape.

The rest of this paper is organized as follows. Section II discusses the impacts of uncertainty due to approximation errors in evolutionary frameworks that employ surrogates. Based on the discussion, Section III provides a generalization of surrogate-assisted evolutionary search for both SOO and MOO subsequently. We summarize the empirical studies on some popular SOO and MOO benchmark problems in Section IV. Finally, Section V concludes this paper.

II. Impacts of Approximation Errors in Surrogate-Assisted Evolutionary Algorithms

In this section, we briefly discuss the effects of uncertainty introduced by inaccurate approximation models on surrogate-assisted evolutionary algorithms (SAEA) search performance. Without loss of generality, here we consider computationally expensive minimization problems under limited computational budget with bound constraints of the following form:

$$\begin{aligned} \text{minimize:} \quad & f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_r(\mathbf{x}) \\ \text{subject to:} \quad & x_i^l \le x_i \le x_i^u \end{aligned} \tag{1}$$

where $i = 1, 2, \ldots, d$, d is the dimensionality of the search problem, r is the number of objective functions, and x_i^l, x_i^u are the lower and upper bounds of the ith dimension of vector $\mathbf{x}$, respectively.

Note that when more than one objective is involved for approximation, there are two commonly adopted strategies, i.e., 1) one approximation model per objective function; 2) one approximation model for an aggregated (linear or nonlinear combination) objective function, $f_{\text{aggr}}(\mathbf{x})$. In this paper, we consider the second strategy. Since in single-objective context, $f_{\text{aggr}}(\mathbf{x}) = f(\mathbf{x}) = f_1(\mathbf{x})$, the term $f(\mathbf{x})$ might be used interchangeably to $f_{\text{aggr}}(\mathbf{x})$ for brevity purposes when only single-objective context is considered.

If $f_{\text{aggr}}(\mathbf{x})$ denotes the original fitness function and the approximated function is $\hat{f}_{\text{aggr}}(\mathbf{x})$, the approximation errors at any solution vector $\mathbf{x}$ is $e(\mathbf{x})$, i.e., the uncertainty introduced by the surrogate at $\mathbf{x}$, may then be defined as

$$e(\mathbf{x}) = |f_{\text{aggr}}(\mathbf{x}) - \hat{f}_{\text{aggr}}(\mathbf{x})|. \tag{2}$$

Here, we highlight the negative and positive impacts introduced by the approximation inaccuracies of the surrogates on SAEA search [44]. The negative impact or otherwise known as the *'curse of uncertainty'* on SAEA search can be briefly defined as the phenomenon where the inaccuracies of the surrogates used results in the SAEA search to stall or converge to false optimum. To illustrate the *'curse'* effect, we refer to Fig. 1(a), where the SAEA is likely to converge to the false optimum of the spline interpolation model due to inaccuracy. On the other hand, the positive impact, i.e., the *'bless of uncertainty'* in SAEA materializes when the use of surrogate(s) brings about greater search improvements over the use of original exact objective/fitness function. For instance, the surrogate can help to traverse the search across valleys and hills of local optima by smoothing the ruggedness/multimodality of the problem landscape. To illustrate the blessing effect, we refer to the example in Fig. 1(b), where a low-order polynomial regression scheme is used to approximate the exact objective function. Due to

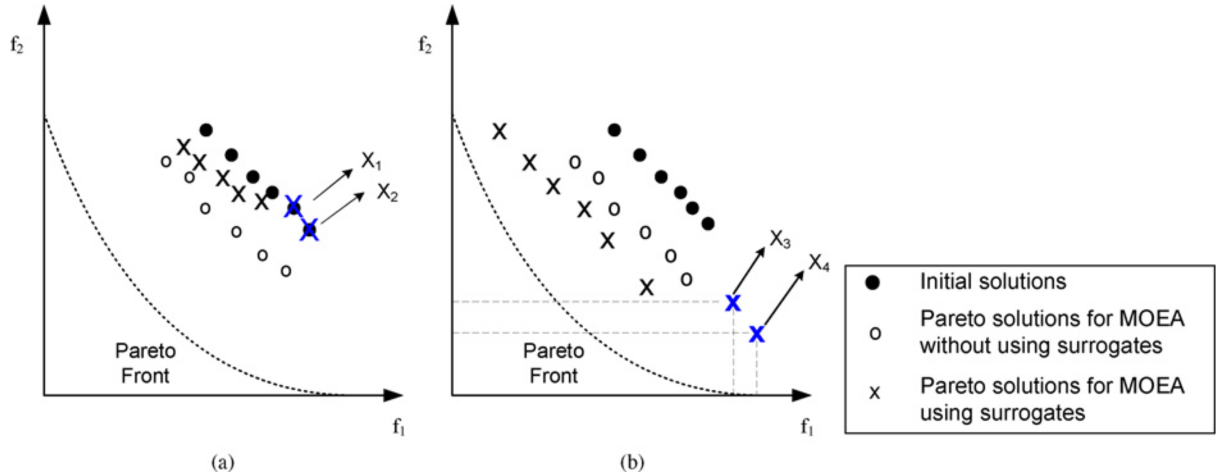


Fig. 2. Curse and bless of uncertainty in multiobjective EA using surrogates. (a) *'Curse of uncertainty'* in multiobjective EA using surrogates. (b) *'Bless of uncertainty'* in multiobjective EA using surrogates.

the smoothing effect of the polynomial surrogate, the search leads to an improved solution that is unlikely to be attained even if the exact objective function is used. Hence, the *'bless of uncertainty'* brings about possible acceleration in the search. Besides a faster convergence, recent study in [32] revealed that the *'bless of uncertainty'* in SAEA also exists in the form of improving evolutionary search diversity through the use of surrogate model.

Next, to illustrate *'curse and bless of uncertainty'* in the context of multiobjective optimization, we refer to the examples in Fig. 2(a) and (b). Fig. 2(a) depicts the effect of *'curse of uncertainty'* in MOEA search due to the presence of inaccurate surrogate models. In Fig. 2(a), the surrogate-assisted MOEA search is observed to be evolving toward poor nondominated solutions in comparison to that based on exact fitness functions. Moreover, those labeled as $\mathbf{x}_1$ and $\mathbf{x}_2$ in Fig. 2(a) suggest that some solutions might stall, while others fail to converge optimally. On the other hand, Fig. 2(b) illustrates the presence of *'bless of uncertainty'* where the errors in the surrogate used is observed to improve the MO evolutionary search in both convergence and diversity measures. Particularly, some improved solutions of the surrogate-assisted search is shown to dominate at least one of its initial solutions, while others such as $\mathbf{x}_3$ and $\mathbf{x}_4$ are newly found nondominated solutions.

III. Generalizing Surrogate-Assisted Evolutionary Search

In this section, we present a generalization of surrogate-assisted evolutionary frameworks for optimization of problems with objective(s) and constraint(s) that are computationally expensive to evaluate. The generalized framework illustrated here for unifying diverse approximation concept synergistically is a surrogate-assisted memetic algorithm that conducts simultaneous local searches on separate *ensemble* and *smoothing* surrogate models. MAs are population-based metaheuristic search methods that are inspired by Darwinian principles of natural evolution and Dawkins notion of a meme defined as a unit of cultural evolution capable of local refinements [46].[2] For example, the brief outline of a traditional MA is provided in Algorithm 1.

Algorithm 1 Memetic Algorithm (for SOO)

```
Initialization: Generate and evaluate a population of design vectors.
while computational budget is not exhausted do
    Apply evolutionary operators (selection, crossover, mutation) to create a new population.

    / * * * * Local Search Phase * * * * /

    for each individual x in current population do
        • Apply local search to find an improved solution, $\mathbf{x}_{\text{opt}}$.
        • Perform replacement using Lamarckian learning, i.e.,
        if $f(\mathbf{x}_{\text{opt}}) < f(\mathbf{x})$ then
            $\mathbf{x} = \mathbf{x}_{\text{opt}}$
        end if
    end for

    / * * End of Local Search Phase * * /

end while
```

In the generalized framework, we introduce first the idea of employing online local ensemble surrogate models constructed from diverse approximation concepts using data points that lie in the vicinity of an initial guess. The surrogate or approxi-

[2] Note that the rationale behind using a memetic framework over a traditional evolutionary framework is multifold [46], [50]. First, we aim to exploit MAs' capability of locating the local and global optima efficiently. Second, a memetic model of adaptation exhibits the plasticity of individuals that a pure genetic model fails to capture. Further, by limiting the use of surrogate models within the local search procedures, the global convergence property of EAs can be ensured. For a greater exposition of local metaheuristics in optimization, the reader is referred to [47]–[49].

mation models are then used to replace the expensive function evaluations performed in the local search phase. The improved solution generated by the local search procedure then replaces the genotype and/or fitness of the original individual.[3]

A. Ensemble Model

To mitigate the *'curse of uncertainty'* caused by the effect of using imperfect surrogate models, we seek for surrogate models that are capable of generating reliable fitness predictions on diverse problems. In particular, since it is almost impossible to know in advance which approximation technique best suits the optimization problem at hand, we consider a synergy of diverse approximation methods through the use of ensemble models to generate reliable accurate predictions across problems of differing problem landscapes [18], [37], [51], as opposed to single surrogate models created by specific approximation scheme that may not be appropriate for the problem at hand. In what follows, we consider online local weighted average ensembles. For instance, in the single-objective context, the predicted ensemble output of $f(\mathbf{x})$ is formulated as

$$\begin{aligned} \hat{f}_{\text{ens}}(\mathbf{x}) &= \sum_{i=1}^{n} c_i \hat{f}_i(\mathbf{x}) \\ \sum_{i=1}^{n} c_i &= 1 \end{aligned} \tag{3}$$

where $\hat{f}_{\text{ens}}(\mathbf{x})$ and $\hat{f}_i(\mathbf{x})$ are the fitness prediction made by the ensemble and ith surrogate model, respectively. The same formulation applies in the multiobjective context where $f_{\text{aggr}}(\mathbf{x})$ is considered. c_i is the weight coefficient associated with the ith surrogate model. A model can be assigned a larger weight if it is found or deemed to be more accurate. Hence, the weighting function becomes

$$c_i = \frac{\sum_{j=1, j\neq i}^{n} \varepsilon_j}{(n-1)\sum_{j=1}^{n} \varepsilon_j} \tag{4}$$

where ε_j is the error measurement for the jth surrogate model. Here, the root mean square error (RMSE) is used as the error measurement. The RMSE of each surrogate model is then of the form

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{m} e^2(\mathbf{x}_i)}{m}} \tag{5}$$

where m is the number of data samples compared, $e(\mathbf{x}_i)$ is the error of prediction for data point $\mathbf{x}_i$, as shown in (2). For greater details on other ensemble model building techniques, interested readers are referred to [37], [38], [40], [41] and [51].

B. Landscape Smoothing Model

Meanwhile, to benefit from the *'bless of uncertainty,'* smoothing techniques including global convex underestimation, tunneling and filling methods are some appropriate alternatives [52] that may be used. Given a problem landscape, smoothing methods transform the function into one with noticeably fewer minima, thus speeding up the evolutionary search. In the generalized framework, global convex underestimation is used for successive smoothing of the problem landscape within the local search phase which is realized through low-order polynomial regression (PR). Besides the generalization property of PR models on rugged landscape, the low-computational costs incurred makes them very efficient as online surrogate models. Note that the PR model may be used in both ensemble and the smoothing models, hence only a one-time model building cost is involved.

C. GSM Framework for Single-Objective Optimization

In this subsection, we describe the generalized surrogate memetic framework for single-objective optimization. A brief outline of the generalized surrogate single-objective memetic algorithm (GS-SOMA) is presented in Algorithm 2. Note that the difference between the GS-SOMA and a traditional MA lies in the local search phase of the algorithms.

GS-SOMA begins with the initialization of a population of design points. During the database building phase, the search operates like a traditional evolutionary algorithm based on the original exact fitness function for some initial G_{db} generations. Up to this stage, no form of surrogates are used, and all exact fitness function evaluations made are archived in a central database. Subsequently, the algorithm proceeds into the local search phase. For each individual $\mathbf{x}$, n online surrogates that model the fitness function are created dynamically using m training data points, which lie in the vicinity of $\mathbf{x}$, extracted from the archived database of previously evaluated design points. From the n surrogates, an ensemble model is built. From here, two separate local searches are conducted on: 1) M_1, the ensemble of n surrogate models, and 2) M_2, a low-order PR model. If improved solutions are achieved, GS-SOMA proceeds with the individual replacement scheme. Since we adopt the Lamarckian scheme here, the genotype/phenotype of the initial individual is then replaced by the higher quality solutions among the two that are locally improved based on M_1 and M_2, i.e., $\mathbf{x}^1_{\text{opt}}$ or $\mathbf{x}^2_{\text{opt}}$. The search cycle is then repeated until the allowed maximum computational budget is exhausted.

D. GSM Framework for Multiobjective Optimization

Next, we describe the generalized surrogate memetic framework in the context of multiobjective optimization (MOO). In MOO, a solution $\mathbf{x}^{(1)}$ is said to dominate solution $\mathbf{x}^{(2)}$ in the objective space, i.e., $\mathbf{x}^{(1)} \preceq \mathbf{x}^{(2)}$ if the following two conditions hold:

1) $\mathbf{x}^{(1)}$ is no worse than $\mathbf{x}^{(2)}$ on all objectives or $f_j(\mathbf{x}^{(1)}) \leq f_j(\mathbf{x}^{(2)})$ for all $j = 1, 2, \ldots, r$;
2) $\mathbf{x}^{(1)}$ is strictly better than $\mathbf{x}^{(2)}$ on at least one objective, or $f_j(\mathbf{x}^{(1)}) < f_j(\mathbf{x}^{(2)})$ for at least one $j \in 1, 2, \ldots, r$.

[3] There are two basic replacement strategies in MAs [50].

1) *Lamarckian learning* forces the genotype to reflect the result of improvement in local search by placing the locally improved individual back into the population to compete for reproductive opportunities.
2) *Baldwinian learning* only alters the fitness of the individuals and the improved genotype is not encoded back into the population.

For the sake of brevity, we consider Lamarckian learning in this paper.

Algorithm 2 Generalized Surrogate Single-Objective Memetic Algorithm (GS-SOMA)

```
initialization: Generate and evaluate a database containing a population of designs, archive all exact evaluations into the database.
while computational budget is not exhausted do
  if generation count $<$ database building phase ($G_{\text{db}}$) then
    Evolve the population using exact fitness function evaluations, archive all exact evaluations into the database.
  else
    Apply evolutionary operators (selection, crossover, mutation) to create a new population.

    /**** Local Search Phase ****/

    for each individual $\mathbf{x}$ in the population do
      • Find $m$ nearest points to $\mathbf{x}$ in database as training points for surrogate models.
      • Build model-1: $M_1$, as an ensemble of all $M'_j$ for $j = 1, \ldots, n$ where $n$ is the number of surrogate models used.
      • Build model-2: $M_2$, which is a low-order PR model.
      • Apply local search in $M_1$ to arrive at $\mathbf{x}^1_{\text{opt}}$, and $M_2$ to arrive at $\mathbf{x}^2_{\text{opt}}$.
      • Replace $\mathbf{x}$ with the locally improved solution, i.e.,
      if $f(\mathbf{x}^1_{\text{opt}}) < f(\mathbf{x}^2_{\text{opt}})$ then
        $\mathbf{x} = \mathbf{x}^1_{\text{opt}}$
      else
        $\mathbf{x} = \mathbf{x}^2_{\text{opt}}$
      end if
      • Archive all new exact function evaluations into the database.
    end for

    /** End of Local Search Phase **/

  end if
end while
```

Algorithm 3 Multiobjective Memetic Algorithm

```
initialization: Generate and evaluate a population of design vectors.
while computational budget is not exhausted do
  Apply MO evolutionary operators (selection, crossover, mutation) to create a new population.

  /**** Local Search Phase ****/

  for each individual $\mathbf{x}$ in the population do
    • Generate a random weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_r)$, $\sum_{i=1}^{r} w_i = 1$ where $r$ is the number of objectives.
    • Apply local search in $f_{\text{aggr}} = \sum_{i=1}^{r} w_i f_i(\mathbf{x})$ to find an improved solution, $\mathbf{x}^{\text{opt}}$.
    • Perform Lamarckian learning, i.e.,
    if $f_{\text{aggr}}(\mathbf{x}_{\text{opt}}) < f_{\text{aggr}}(\mathbf{x})$ then
      $\mathbf{x} = \mathbf{x}_{\text{opt}}$
    end if
  end for

  /** End of Local Search Phase **/

end while
```

If set P is the entire feasible search space, the nondominated set P^* is labeled as the *Pareto-optimal set*. Any two solutions in P^* must nondominate each other, i.e., $\mathbf{x}^{(1)} \sim \mathbf{x}^{(2)}$. On the other hand, Pareto front (PF^*) is the image of the Pareto-optimal set in objective space. The brief outline of a typical multiobjective memetic algorithm (MOMA) using weighting (scalarization) technique [58]–[60] is illustrated in Algorithm 3. In contrast, the studied GSM framework for multiobjective optimization (GS-MOMA) is outlined in Algorithm 4. Note that the key differences of the two algorithms lie in the local search phase and selection pool forming phase.

GS-MOMA begins with the population initialization phase and evolutionary search based on exact fitness function for a number of early generations, G_{db}, before entering the local search phase. In the local search phase, independent local searches are conducted on: 1) M_1, the ensemble of n surrogate models; 2) M_2, the smoothing low-order PR model on each individual of the generated offspring population. For the sake of brevity, the core distinguishing feature of GS-MOMA can be noted in line 17 of Algorithm 4, i.e., the existence of the *Replace&Archive* procedure.

The *Replace&Archive* procedure performs replacements based on domination between the original offspring and the two local optima found. The original offspring will only be replaced by one dominating optimum found. Any other local optima are then saved into the learning archive, A_l. Note that the result of GS-MOMAs local searches is either $\mathbf{x}_{\text{opt}} \preceq \mathbf{x}$ or $\mathbf{x}_{\text{opt}} \sim \mathbf{x}$. Otherwise, there is no improvement to the original offspring, and hence we get $\mathbf{x}_{\text{opt}} == \mathbf{x}$.

Based on the procedure in Algorithm 5, the possible local search outcomes and corresponding actions taken by the scheme are summarized in Table I. Note that there exist six possible actions to be taken by GS-MOMA which are summarized as follows:

1) replacement is performed once [e.g., Fig. 3(a)];
2) two subsequent replacements are performed [e.g., Fig. 3(b)];
3) both replacement and archiving are performed [e.g., Fig. 3(c)];
4) archiving is performed once [e.g., Fig. 3(d)];
5) archiving is performed twice [e.g., Fig. 3(e)];
6) neither replacement nor archiving is performed [e.g., Fig. 3(f)].

TABLE I

ACTIONS TAKEN BY THE *Replace&Archive* SCHEME IN GS-MOMA FOR CORRESPONDING RESULTS OF LOCAL SEARCHES. NOTE THAT IRRELEVANT CASES HAVE BEEN EXCLUDED FOR BREVITY

$\mathbf{x}^1_{\text{opt}}$ vs $\mathbf{x}$	$\mathbf{x}^2_{\text{opt}}$ vs $\mathbf{x}$	$\mathbf{x}^1_{\text{opt}}$ vs $\mathbf{x}^2_{\text{opt}}$	Actions taken by GS-MOMA
$\preceq$	$\preceq$	$\preceq$	$\mathbf{x} = \mathbf{x}^1_{\text{opt}}$
$\preceq$	$\preceq$	$\succ$	$\mathbf{x} = \mathbf{x}^2_{\text{opt}}$
$\preceq$	$\preceq$	$\sim$	$\mathbf{x} = \mathbf{x}^1_{\text{opt}}$, archive $\mathbf{x}^2_{\text{opt}}$
$\preceq$	$\preceq$	==	$\mathbf{x} = \mathbf{x}^1_{\text{opt}}$
$\preceq$	==	$\preceq$	$\mathbf{x} = \mathbf{x}^1_{\text{opt}}$
$\preceq$	$\sim$	$\preceq$	$\mathbf{x} = \mathbf{x}^1_{\text{opt}}$
$\preceq$	$\sim$	$\sim$	$\mathbf{x} = \mathbf{x}^1_{\text{opt}}$, archive $\mathbf{x}^2_{\text{opt}}$
==	$\preceq$	$\succ$	$\mathbf{x} = \mathbf{x}^2_{\text{opt}}$
==	==	==	No changes
==	$\sim$	$\sim$	Archive $\mathbf{x}^2_{\text{opt}}$
$\sim$	$\preceq$	$\succ$	$\mathbf{x} = \mathbf{x}^2_{\text{opt}}$
$\sim$	$\preceq$	$\sim$	$\mathbf{x} = \mathbf{x}^2_{\text{opt}}$, archive $\mathbf{x}^1_{\text{opt}}$
$\sim$	==	$\sim$	Archive $\mathbf{x}^1_{\text{opt}}$
$\sim$	$\sim$	$\preceq$	Archive $\mathbf{x}^1_{\text{opt}}$
$\sim$	$\sim$	$\succ$	Archive $\mathbf{x}^2_{\text{opt}}$
$\sim$	$\sim$	$\sim$	Archive $\mathbf{x}^1_{\text{opt}}$ and $\mathbf{x}^2_{\text{opt}}$
$\sim$	$\sim$	==	Archive $\mathbf{x}^1_{\text{opt}}$

At the end of each GS-MOMA generation, A_l is combined with the current parent population, P_c, and the offspring population, P_o to form the entire pool of individuals, P_s that will then undergo the MOEA selection mechanism, i.e., $P_s = P_c \bigcup P_o \bigcup A_l$. From here, the process described repeats until the maximum computational budget of the GS-MOMA is exhausted.

E. Local Search Scheme

In the GSM framework for SO/MOO, a trust-region-regulated search strategy is utilized to ensure convergence to some local optimum or the global optimum of the exact computationally expensive fitness function [8], [53], [61], even though surrogate models are deployed in the local search. For each individual in the GS-SO/MOMA population, the local search (refer to line 14 of Algorithm 2 and line 16 of Algorithm 4) proceeds with a sequence of trust-region subproblems of the form

$$\begin{aligned} \text{minimize}: &\quad \hat{f}^k(\mathbf{x}_c^k + \mathbf{s}) \\ \text{subject to}: &\quad \|\mathbf{s}\| \leq \Omega^k \end{aligned} \tag{6}$$

where $k = 0, 1, 2, \ldots, k_{\max}$, $\hat{f}(\mathbf{x})$ is the approximation function corresponding to the objective function $f(\mathbf{x})$. Meanwhile, $\mathbf{x}_c^k$, $\mathbf{s}$, and Ω^k represent the initial guess (current best solution) at iteration k, an arbitrary step, and the trust-region radius at iteration k, respectively. In our experiments, the Sequential Quadratic Programming (SQP) [54] is used to minimize the sequence of subproblems on the approximated landscape.

During the local search, the initial trust-region radius Ω is initialized based on the minimum and maximum values of the m design points used to construct the surrogate model (refer to line 11 of Algorithm 2 and line 13 of Algorithm 4). The trust-region radius for iteration k, i.e., Ω^k is updated based on a measure which indicates the accuracy of the surrogate model at the kth local optimum, $\mathbf{x}^k_{\text{opt}}$. This measure, ρ^k, provides a measure of the actual versus predicted change in the exact fitness function values at the kth local optimum and is calculated as

$$\rho^k = \frac{f(\mathbf{x}_c^k) - f(\mathbf{x}^k_{\text{opt}})}{\hat{f}(\mathbf{x}_c^k) - \hat{f}(\mathbf{x}^k_{\text{opt}})}. \tag{7}$$

The value of ρ^k is then used to update the trust-region radius as follows [61]:

$$\begin{aligned} \Omega^{k+1} &= C_1\Omega^k, && \text{if } \rho^k \leq C_2 \\ &= \Omega^k, && \text{if } C_2 < \rho^k \leq C_3 \\ &= C_4\Omega^k, && \text{if } \rho^k > C_3 \end{aligned} \tag{8}$$

where C_1, C_2, C_3, and C_4 are constants. Typically, $C_1 \in (0, 1)$ and $C_4 \geq 1$ for the scheme to work efficiently. From experience, we set $C_1 = 0.25$, $C_2 = 0.25$, $C_3 = 0.75$, and $C_4 = 2$, if $||\mathbf{x}^k_{\text{opt}} - \mathbf{x}^k_c||_\infty = \Omega^k$ or $C_4 = 1$, if $||\mathbf{x}^k_{\text{opt}} - \mathbf{x}^k_c||_\infty < \Omega^k$.

The trust-region radius for the next iteration, Ω^{k+1}, is reduced if the accuracy of the surrogate, measured by ρ^k is low. On the other hand, Ω^k is doubled if the surrogate is found to be accurate and the kth local optimum, $\mathbf{x}^k_{\text{opt}}$, lies on the trust-region bounds. Otherwise the trust-region radius remains unchanged.

The initial guess of the optimum at iteration $k + 1$ becomes

$$\begin{aligned} \mathbf{x}_c^{k+1} &= \mathbf{x}^k_{\text{opt}}, && \text{if } \rho^k > 0 \\ &= \mathbf{x}_c^k, && \text{if } \rho^k \leq 0. \end{aligned} \tag{9}$$

The trust-region process for an individual terminates when the termination condition is satisfied. For instance, this termination condition could be when the trust-region radius Ω approaches ε, where ε represents some small trust-region radius, or when a maximum number of iteration k_{term} is reached.

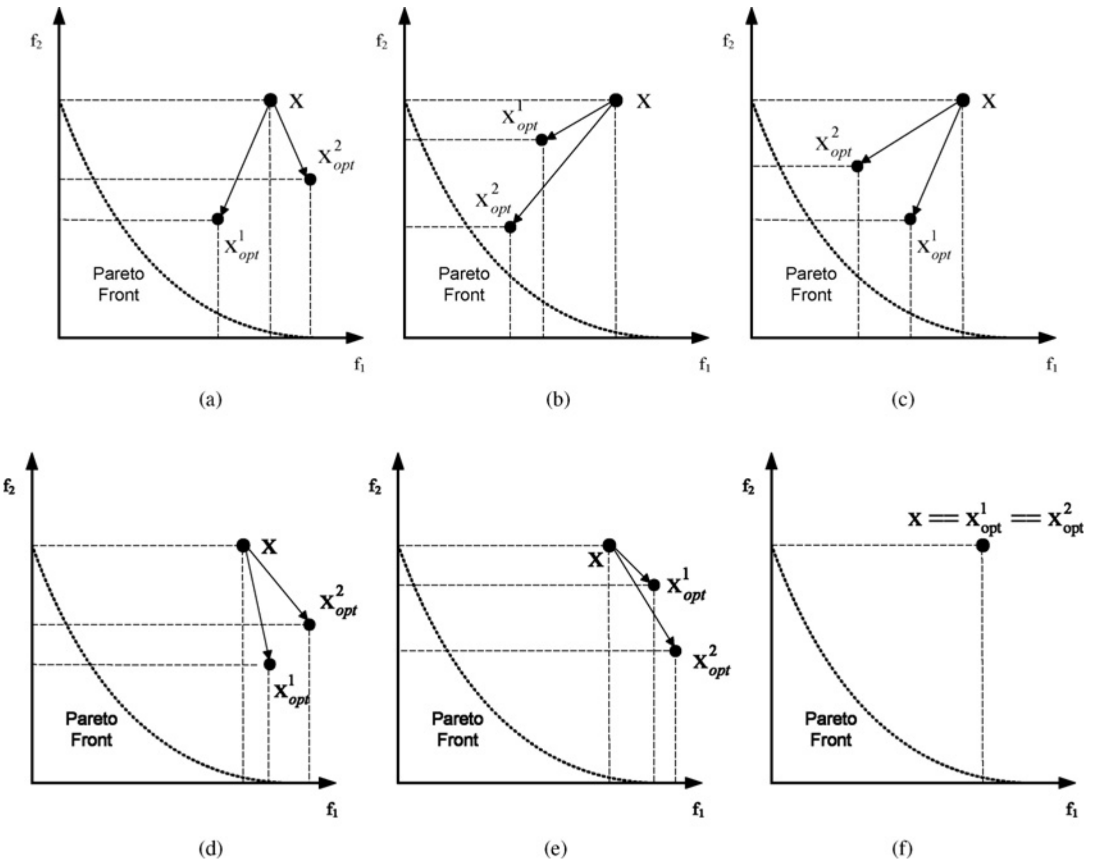


Fig. 3. Examples of the six different actions taken by the *Replace&Archive* scheme in GS-MOMA for corresponding results of local searches. (a) An example of the case where replacement is performed only once by GS-MOMA. $(\mathbf{x}^1_{\text{opt}} \preceq \mathbf{x}) \wedge (\mathbf{x}^1_{\text{opt}} \preceq \mathbf{x}^2_{\text{opt}}) \wedge (\mathbf{x} \sim \mathbf{x}^2_{\text{opt}})$. $\mathbf{x}^1_{\text{opt}}$ replaces $\mathbf{x}$. (b) An example of the case where two subsequent replacements are performed by GS-MOMA. $(\mathbf{x}^1_{\text{opt}} \preceq \mathbf{x}) \wedge (\mathbf{x}^2_{\text{opt}} \preceq \mathbf{x}^1_{\text{opt}})$. $\mathbf{x}^1_{\text{opt}}$ replaces $\mathbf{x}$, followed by $\mathbf{x}^2_{\text{opt}}$ replaces $\mathbf{x}$. (c) An example of the case where both replacement and archiving are performed by GS-MOMA. $(\mathbf{x}^1_{\text{opt}} \preceq \mathbf{x}) \wedge (\mathbf{x}^2_{\text{opt}} \preceq \mathbf{x}) \wedge (\mathbf{x}^1_{\text{opt}} \sim \mathbf{x}^2_{\text{opt}})$. $\mathbf{x}^1_{\text{opt}}$ replaces $\mathbf{x}$, $\mathbf{x}^2_{\text{opt}}$ is archived in A_l. (d) An example of the case where archiving is performed only once by GS-MOMA. $(\mathbf{x} \sim \mathbf{x}^1_{\text{opt}}) \wedge (\mathbf{x} \sim \mathbf{x}^2_{\text{opt}}) \wedge (\mathbf{x}^1_{\text{opt}} \preceq \mathbf{x}^2_{\text{opt}})$. $\mathbf{x}^1_{\text{opt}}$ is archived in A_l. (e) An example of the case where archiving is performed twice by GS-MOMA. $(\mathbf{x} \sim \mathbf{x}^1_{\text{opt}}) \wedge (\mathbf{x} \sim \mathbf{x}^2_{\text{opt}}) \wedge (\mathbf{x}^1_{\text{opt}} \sim \mathbf{x}^2_{\text{opt}})$. Both $\mathbf{x}^1_{\text{opt}}$ and $\mathbf{x}^2_{\text{opt}}$ are archived in A_l. (f) An example of the case where neither replacement nor archiving is performed. No new optimum is found.

IV. Empirical Study

In this section, we present an empirical study on the GSM framework for solving single and multiobjective optimization problems. In the present study, we considered a diverse set of exact interpolating and generalizing approximation techniques for constructing the local surrogate models, i.e., M_1 and M_2. These include the interpolating Kriging/Gaussian process (GP), interpolating linear spline radial basis function (RBF) and second-order polynomial regression (PR). For greater details on GP, PR, and RBF, the reader is referred to [55]–[57] and Appendix A.

A. Parameters of GSM Framework

In this subsection, we discuss the user-specified parameters of the GSM framework. Apart from the parameters of the underlying SO/MOEA, the generalized framework has three additional user-specified parameters: m, G_{db}, and k_{term}.

Since model accuracy is highly dependent on the sufficiency of the m data points used for model building, the size of nearest neighboring points used (based on Euclidean distance) is defined by $d+(d+1)(d+2)/2$, where d is the dimensionality of the optimization problem. It is worth noting that the complexity for identifying these m points is negligible compared to the cost of surrogate model building. Moreover, since our emphasis here is with regard to a framework that is tailored for solving computationally expensive problems, i.e., problems that may cost from minutes to hours of computational time per evaluation, such overheads are considered to be insignificant. From these m data points, as many as $(d+1)(d+2)/2$ among them[4] are chosen uniformly as the training data for building the surrogates, the remaining data points then form the set for validating the prediction quality of the surrogate.

Parameter G_{db}, on the other hand, defines the period of the database building phase (refer to lines 3–5 in Algorithms 2 and 4) before the core operation of the GSM framework begins to take effect. Hence, G_{db} can be adapted for different optimization problems according to the fulfillment on the requirement of parameter m. The lower bound of G_{db} is defined by the period to acquire a minimum of m data points for construction of reliable surrogate models.

[4] This amount corresponds to the minimum number of data points required for building quadratic regression models.

Algorithm 4 Generalized Surrogate Multiobjective Memetic Algorithm (GS-MOMA)

```
initialization: Generate and evaluate an initial population with $N_{pop}$ individuals, archive all exact evaluations into a database.
while computational budget is not exhausted do
    if generation count < database building phase ($G_{db}$) then
        Evolve the population using exact fitness function evaluations, archive all exact evaluations into the database.
    else
        Generate the offspring population, $P_o$ using MO evolutionary operators (selection, crossover, mutation) on the selection pool.

        /**** Local Search Phase ****/

        Initialize the learning archive, $A_l$ to empty state.
        for each individual x in the offspring population do
            • Generate a random weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_r)$, $\sum_{i=1}^{r} w_i = 1$ where $r$ is the number of objectives.
            • Find $m$ nearest points to x in database as training points for surrogate models.
            • Build model-1: $M_1$, as an ensemble of all $M'_j$ for $j = 1, \ldots, n$ where $n$ is the number of surrogate models used of $f_{aggr} = \sum_{i=1}^{r} w_i f_i(\mathbf{x})$
            • Build model-2: $M_2$, which is a low-order PR model, of $f_{aggr} = \sum_{i=1}^{r} w_i f_i(\mathbf{x})$
            • Apply local search in $M_1$ to arrive at $\mathbf{x}^1_{opt}$, and $M_2$ to arrive at $\mathbf{x}^2_{opt}$
            • Replace&Archive( x, $\mathbf{x}^1_{opt}$, $\mathbf{x}^2_{opt}$, $A_l$ )
        end for

        /** End of Local Search Phase **/


        /**** Selection pool forming ****/

        Form selection pool, $P_s = P_c \bigcup P_o \bigcup A_l$.

        /** End of selection pool forming **/

    end if
end while
```

Algorithm 5 Procedure *Replace&Archive*(**x**, $\mathbf{x}^1_{opt}$, $\mathbf{x}^2_{opt}$, A_l)

```
if $\mathbf{x}^1_{opt} \preceq \mathbf{x}$ then
    $\mathbf{x} = \mathbf{x}^1_{opt}$
    if $\mathbf{x}^2_{opt} \preceq \mathbf{x}^1_{opt}$ then
        $\mathbf{x} = \mathbf{x}^2_{opt}$
    else if $\mathbf{x}^2_{opt} \sim \mathbf{x}^1_{opt}$ then
        Archive $\mathbf{x}^2_{opt}$ in $A_l$
    end if
else if $\mathbf{x}^2_{opt} \preceq \mathbf{x}$ then
    $\mathbf{x} = \mathbf{x}^2_{opt}$
    if $\mathbf{x}^2_{opt} \sim \mathbf{x}^1_{opt}$ then
        Archive $\mathbf{x}^1_{opt}$ in $A_l$
    end if
else if $(\mathbf{x}^1_{opt} \sim \mathbf{x}) \wedge (\mathbf{x}^2_{opt} == \mathbf{x})$ then
    Archive $\mathbf{x}^1_{opt}$ in $A_l$
else if $(\mathbf{x}^2_{opt} \sim \mathbf{x}) \wedge (\mathbf{x}^1_{opt} == \mathbf{x})$ then
    Archive $\mathbf{x}^2_{opt}$ in $A_l$
else if $(\mathbf{x}^1_{opt} \sim \mathbf{x}) \wedge (\mathbf{x}^2_{opt} \sim \mathbf{x})$ then
    if $(\mathbf{x}^1_{opt} \preceq \mathbf{x}^2_{opt}) \parallel (\mathbf{x}^1_{opt} == \mathbf{x}^2_{opt})$ then
        Archive $\mathbf{x}^1_{opt}$ in $A_l$
    else if $\mathbf{x}^2_{opt} \preceq \mathbf{x}^1_{opt}$ then
        Archive $\mathbf{x}^2_{opt}$ in $A_l$
    else
        Archive $\mathbf{x}^1_{opt}$ and $\mathbf{x}^2_{opt}$ in $A_l$
    end if
end if
```

Theoretically, the trust-region local search scheme terminates when the trust-region radius, Ω approaches ε, where ε represents some very small value for termination condition (refer to Section III-E). Nevertheless, for practical reason, under limited computational budget, it is more appropriate to derive a suitable value for k_{term} as the termination condition in the trust-region local search. In what follows, we present a theoretical bound for k_{term}

$$\Omega^1_{min} (C_1)^{k_{min}} \leq \varepsilon \tag{10}$$

$$\Rightarrow (C_1)^{k_{min}} \leq \frac{\varepsilon}{\Omega^1_{min}} \tag{11}$$

$$\Rightarrow k_{min} \log C_1 \leq \log \frac{\varepsilon}{\Omega^1_{min}}. \tag{12}$$

Since $C_1 \in (0, 1) \rightarrow \log C_1 < 0$, we arrive at

$$\Rightarrow k_{min} \geq \left(\log\left(\frac{\varepsilon}{\Omega^1_{min}}\right)\right) / (\log C_1) \tag{13}$$

$$\Rightarrow k_{min} \geq \log_{C_1}\left(\frac{\varepsilon}{\Omega^1_{min}}\right). \tag{14}$$

Similarly, the maximum number of trust-region iterations in the local search, i.e., k_{max}, is estimated by

$$k_{max} < N^{max}_{succ} + N^{max}_{succ} \log_{C_1}\left(\frac{\varepsilon}{\Omega^1_{max}}\right) \tag{15}$$

$$\Rightarrow k_{max} < N^{max}_{succ}\left(1 + \log_{C_1}\left(\frac{\varepsilon}{\Omega^1_{max}}\right)\right). \tag{16}$$

Note that N^{max}_{succ} is the maximum number of successful iterations, while Ω^1_{min} and Ω^1_{max} are the lower and upper bounds of the initial trust-region radius. In effect, the bounds for k_{term} as the termination condition can be derived as

$$\log_{C_1}\left(\frac{\varepsilon}{\Omega^1_{min}}\right) \leq k_{term} < N^{max}_{succ}\left(1 + \log_{C_1}\left(\frac{\varepsilon}{\Omega^1_{max}}\right)\right). \tag{17}$$

In the trust-region-regulated local search, Ω^1 depends on the local region of interest where the initial m nearest neighbors are located. Hence, it is not possible to define this term precisely for any new optimization problem. For instance, if $\Omega^1_{min} \approx 10\varepsilon$ and $C_1 = 0.25$, we arrive at

$$k_{term} \geq \frac{\log 0.1}{\log 0.25}$$

$$k_{term} \geq 1.66. \tag{18}$$

TABLE II
THE BENCHMARK PROBLEMS USED (F1 − F10) FOR THE EMPIRICAL STUDY OF SINGLE-OBJECTIVE OPTIMIZATION

Benchmark Problem	Description	Global Optimum $f(x^*)$
F1	Ackley	0.0
F2	Griewank	0.0
F3	Rosenbrock	0.0
F4	Shifted Rotated Rastrigin (F10 in [63])	−330.0
F5	Shifted Rotated Weierstrass (F11 in [63])	90.0
F6	Shifted Expanded Griewank plus Rosenbrock (F13 in [63])	−130.0
F7	Hybrid Composition Function (F15 in [63])	120.0
F8	Rotated Hybrid Composition Function (F16 in [63])	120.0
F9	Rotated Hybrid Composition Function with Narrow Basin Global Optimum (F19 in [63])	10.0
F10	Noncontinuous Rotated Hybrid Composition Function (F23 in [63])	360.0

TABLE III
DEFINITION OF THE SINGLE-OBJECTIVE MAS (SOMAS) COMPARED

Algorithms	Definition
GA	No surrogate is used
SS-SOMA-GP	Single surrogate SOMA with M_1: GP
SS-SOMA-PR	Single surrogate SOMA with M_1: PR
SS-SOMA-RBF	Single surrogate SOMA with M_1: RBF
SS-SOMA-Perfect	Single surrogate SOMA with M_1: Perfect model
GS-SOMA	Generalized surrogate SOMA with M_1: *Weighted-average* ensemble of GP, PR, and RBF M_2: PR

TABLE IV
SETTING OF EXPERIMENTS FOR GA, SS-SOMA, SS-SOMA-PERFECT, AND GS-SOMA

Parameters Setting	
Population size (N_{pop})	100
Crossover probability (P_{cross})	0.9
Mutation probability (P_{mut})	0.1
Maximum number of exact evaluations	8000
Evolutionary operators	Uniform crossover & mutation, elitism and ranking selection
Number of trust-region iteration(k_{term}) for SS-SOMA and GS-SOMA	3
Database building phase (G_{db}) for SS-SOMA and GS-SOMA (in number of generations)	20
Number of independent runs	20

As opposed to using $k_{\text{term}} = 1$ which translates to a single iteration local search, a minimum value of $k_{\text{term}} \geq 2$ is more practical to allow the mechanisms of the trust-region-regulated local search to take effect.

B. Single-Objective Optimization

Empirical study on the GS-SOMA is performed using 10 benchmark problems (F1–F10) reported in [62], [63] and summarized here in Table II. More detailed description of the problems is also provided in Appendix B. It consists of problems with diverse properties in terms of separability, multimodality, and continuity.

In this paper, all the benchmark problems are configured with a dimensionality of $d = 30$ for SOO. Performance comparisons are then made between the GA, SS-SOMA-GP, SS-SOMA-PR, SS-SOMA-RBF, SS-SOMA-Perfect, and GS-SOMA (refer to Table III for the definition of the algorithms investigated here). Note that to facilitate a fair comparison, the surrogate memetic variants are built on top of the same GA used in the study, which ensures that any improvement observed is a direct contribution of the surrogate framework considered. SS-SOMA-*XXX* refers to the different surrogate-assisted single-objective MA variants, each with a unique approximation method used to generate the surrogate model. For instance, *XXX* in SS-SOMA-*XXX* refers to GP, PR, or RBF. On the other hand, SS-SOMA-Perfect refers to an SS-SOMA that employs an imaginary approximation technique that generates error-free surrogates,[5] i.e., RMSE = 0. Hence, the notion of 'curse or blessing of uncertainty' does not exist in the SS-SOMA-Perfect search. As such, any SS-SOMA-*XXX* that under/out-performs SS-SOMA-Perfect is clearly attributed to the effects of curse and bless of uncertainty, respectively. Last but not least, GS-SOMA refers to the Generalized Surrogate framework for single-objective optimization. The common parameter settings of the algorithms used in the present experimental study are summarized in Table IV.

1) *Experimental Results:* In Tables V–XIV, the detailed statistical results of 20 independent runs for SS-SOMAs, GS-SOMA, and GA are presented. The GS-SOMA and best performing SS-SOMA are highlighted in the tables. Note that none of the SS-SOMAs always dominates in performance on all 10 benchmark problems. This makes good sense since the performance of any surrogate-assisted evolutionary search would depend on the match between the characteristics of the problem landscape and approximation scheme used. For instance, in the tables, it is shown that SS-SOMA-PR serves to be best suited for F1, F5, and F9 since it outperforms all other algorithms on these problems. Similarly, this also applies to SS-SOMA-GP which excels on F3. On the other hand, SS-SOMA-RBF, though not superior, performs relatively well on F3, F4, F7, and F8. Moreover, it is worth noting that the SS-SOMAs are observed to have performed much poorly on several occasions. For instance, SS-SOMA-PR fares badly on F3, F4, F7, and F8. The same is true for SS-SOMA-GP on F1, F4–F8, and F10, and SS-SOMA-RBF on F1, F2, F5, F6, F9, and F10.

On the other hand, the results in Tables V–XIV, indicate that GS-SOMA consistently performs well on all the benchmark problems. The *t*-test results, i.e., at 95% confidence level, for the different algorithms as reported in Table XV confirm that

[5] An error-free surrogate model can be realized by using exact fitness function in the portion of SS-SOMA where a surrogate model should be used, but the incurred fitness evaluation is counted only as many as in SS-SOMA.

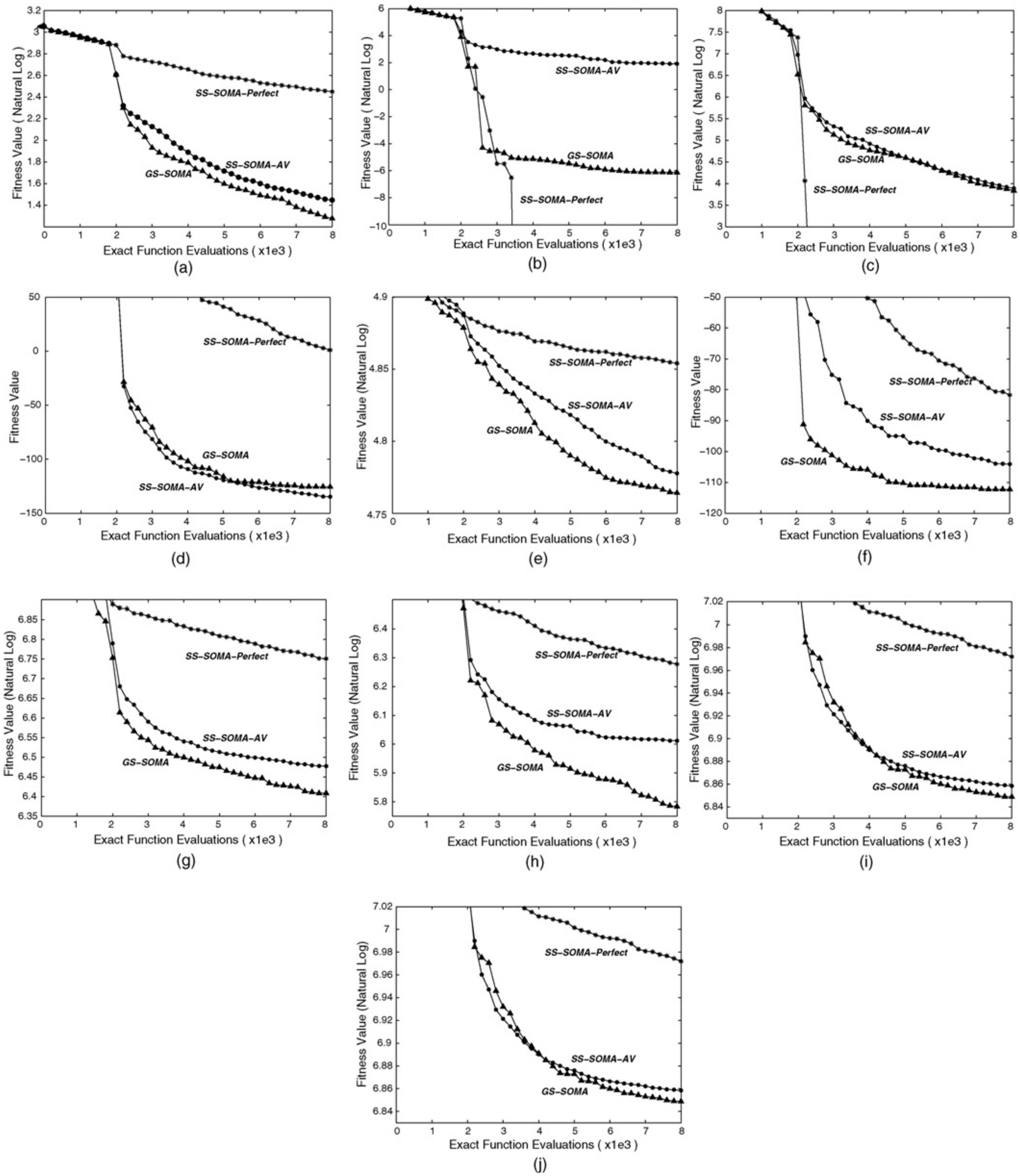


Fig. 4. Convergence trends for F1–F10 obtained from GS-SOMA, SS-SOMA-Perfect, and SS-SOMA-AV.

TABLE V

STATISTICS OF THE FINAL SOLUTION QUALITY AT THE END OF 8000 EXACT FUNCTION EVALUATIONS FOR F1 USING GA, SS-SOMA-GP, SS-SOMA-PR, SS-SOMA-RBF, AND GS-SOMA

Optimization	Statistical Values				
Algorithm	Mean	Std. Dev.	Median	Best	Worst
GA	1.24e+01	9.50e−01	1.23e+01	1.12e+01	1.42e+01
SS-SOMA-GP	6.43e+00	9.73e−01	3.98e+00	2.87e+00	1.56e+01
SS-SOMA-PR	1.39e+00	1.93e−01	1.36e+00	1.14e+00	1.75e+00
SS-SOMA-RBF	4.91e+00	7.57e−01	4.86e+00	3.78e+00	6.09e+00
GS-SOMA	3.58e+00	5.09e−01	3.67e+00	2.87e+00	4.28e+00

TABLE VI

STATISTICS OF THE FINAL SOLUTION QUALITY AT THE END OF 8000 EXACT FUNCTION EVALUATIONS FOR F2 USING GA, SS-SOMA-GP, SS-SOMA-PR, SS-SOMA-RBF, AND GS-SOMA

Optimization	Statistical Values				
Algorithm	Mean	Std. Dev.	Median	Best	Worst
GA	4.58e+01	8.61e+00	4.67e+01	2.15e+01	6.19e+01
SS-SOMA-GP	1.79e+01	8.58e+00	1.07e+01	5.15e−09	3.00e+01
SS-SOMA-PR	1.18e−02	2.78e−02	4.29e−08	7.48e−10	1.19e−01
SS-SOMA-RBF	7.49e−01	8.98e−02	7.51e−01	6.02e−01	8.72e−01
GS-SOMA	2.2e−03	4.60e−03	8.95e−09	1.40e−10	1.54e−02

TABLE VII

STATISTICS OF THE FINAL SOLUTION QUALITY AT THE END OF 8000 EXACT FUNCTION EVALUATIONS FOR F3 USING GA, SS-SOMA-GP, SS-SOMA-PR, SS-SOMA-RBF, AND GS-SOMA

Optimization	Statistical Values				
Algorithm	Mean	Std. Dev.	Median	Best	Worst
GA	4.10e+02	1.01e+02	3.85e+02	2.33e+02	5.73e+02
SS-SOMA-GP	2.99e+01	7.73e−01	3.00e+01	2.87e+01	3.11e+01
SS-SOMA-PR	6.73e+01	2.55e+01	5.62e+01	3.72e+01	1.04e+02
SS-SOMA-RBF	4.90e+01	2.92e+01	3.97e+01	2.92e+01	1.57e+02
GS-SOMA	4.63e+01	2.92e+01	3.02e+01	2.83e+01	1.26e+02

TABLE VIII

STATISTICS OF THE FINAL SOLUTION QUALITY AT THE END OF 8000 EXACT FUNCTION EVALUATIONS FOR F4 USING GA, SS-SOMA-GP, SS-SOMA-PR, SS-SOMA-RBF, AND GS-SOMA

Optimization	Statistical Values				
Algorithm	Mean	Std. Dev.	Median	Best	Worst
GA	−5.46e+01	3.01e+01	−5.48e+01	−1.11e+02	5.19e−01
SS-SOMA-GP	−1.19e+02	1.87e+01	−1.17e+02	−1.50e+02	−8.71e+01
SS-SOMA-PR	−1.19e+02	1.23e+01	−1.21e+02	−1.43e+02	−9.01e+01
SS-SOMA-RBF	−1.65e+02	1.86e+01	−1.66e+02	−1.91e+02	−1.36e+02
GS-SOMA	−1.26e+02	1.60e+01	−1.23e+02	−1.64e+02	−9.97e+01

TABLE IX

STATISTICS OF THE FINAL SOLUTION QUALITY AT THE END OF 8000 EXACT FUNCTION EVALUATIONS FOR F5 USING GA, SS-SOMA-GP, SS-SOMA-PR, SS-SOMA-RBF, AND GS-SOMA

Optimization	Statistical Values				
Algorithm	Mean	Std. Dev.	Median	Best	Worst
GA	1.26e+02	2.85e+00	1.26e+02	1.20e+02	1.32e+02
SS-SOMA-GP	1.19e+02	4.29e+00	1.20e+02	1.12e+02	1.25e+02
SS-SOMA-PR	1.16e+02	3.79e+00	1.16e+02	1.13e+02	1.25e+02
SS-SOMA-RBF	1.21e+02	2.61e+00	1.21e+02	1.18e+02	1.24e+02
GS-SOMA	1.19e+02	3.05e+00	1.19e+02	1.14e+02	1.24e+02

TABLE X
STATISTICS OF THE FINAL SOLUTION QUALITY AT THE END OF 8000 EXACT FUNCTION EVALUATIONS FOR F6 USING GA, SS-SOMA-GP, SS-SOMA-PR, SS-SOMA-RBF, AND GS-SOMA

Optimization Algorithm	Statistical Values				
	Mean	Std. Dev.	Median	Best	Worst
GA	−9.57e+01	9.43e+00	−9.79e+01	−1.06e+02	−7.28e+01
SS-SOMA-GP	−1.02e+02	2.99e+00	−1.03e+02	−1.05e+02	−9.74e+02
SS-SOMA-PR	−1.06e+02	2.45e+00	−1.07e+02	−1.09e+02	−1.02e+02
SS-SOMA-RBF	−1.03e+02	2.43e+00	−1.03e+02	−1.07e+02	−9.96e+01
GS-SOMA	−1.12e+02	1.05e+00	−1.23e+02	−1.13e+02	−1.11e+02

TABLE XI
STATISTICS OF THE FINAL SOLUTION QUALITY AT THE END OF 8000 EXACT FUNCTION EVALUATIONS FOR F7 USING GA, SS-SOMA-GP, SS-SOMA-PR, SS-SOMA-RBF, AND GS-SOMA

Optimization Algorithm	Statistical Values				
	Mean	Std. Dev.	Median	Best	Worst
GA	7.29e+02	5.92e+01	7.27e+02	6.43e+02	8.21e+02
SS-SOMA-GP	6.81e+02	7.23e+01	6.95e+02	6.02e+02	8.23e+02
SS-SOMA-PR	6.42e+02	5.80e+01	6.34e+02	5.73e+02	7.09e+02
SS-SOMA-RBF	6.27e+02	7.93e+01	5.99e+02	5.95e+02	8.49e+02
GS-SOMA	6.07e+02	3.06e+01	6.00e+02	5.79e+02	6.59e+02

TABLE XII
STATISTICS OF THE FINAL SOLUTION QUALITY AT THE END OF 8000 EXACT FUNCTION EVALUATIONS FOR F8 USING GA, SS-SOMA-GP, SS-SOMA-PR, SS-SOMA-RBF, AND GS-SOMA

Optimization Algorithm	Statistical Values				
	Mean	Std. Dev.	Median	Best	Worst
GA	4.83e+02	6.3e+01	4.62e+02	4.19e+02	6.06e+02
SS-SOMA-GP	4.52e+02	9.66e+01	4.35e+02	3.40e+02	5.63e+02
SS-SOMA-PR	3.94e+02	4.41e+01	3.75e+02	3.43e+02	4.52e+02
SS-SOMA-RBF	3.79e+02	3.3e+01	3.69e+02	3.51e+02	4.41e+02
GS-SOMA	3.25e+02	1.17e+02	2.86e+02	2.32e+02	5.54e+02

TABLE XIII
STATISTICS OF THE FINAL SOLUTION QUALITY AT THE END OF 8000 EXACT FUNCTION EVALUATIONS FOR F9 USING GA, SS-SOMA-GP, SS-SOMA-PR, SS-SOMA-RBF, AND GS-SOMA

Optimization Algorithm	Statistical Values				
	Mean	Std. Dev.	Median	Best	Worst
GA	1.02e+03	2.35e+01	1.02e+03	9.86e+02	1.08e+03
SS-SOMA-GP	9.42e+02	1.71e+01	9.37e+02	9.25e+02	9.81e+02
SS-SOMA-PR	9.32e+02	8.26e+00	9.31e+02	9.22e+02	9.48e+02
SS-SOMA-RBF	9.81e+02	1.43e+01	9.80e+02	9.67e+02	1.00e+03
GS-SOMA	9.42e+02	1.75e+01	9.37e+02	9.30e+02	9.86e+02

TABLE XIV
STATISTICS OF THE FINAL SOLUTION QUALITY AT THE END OF 8000 EXACT FUNCTION EVALUATIONS FOR F10 USING GA, SS-SOMA-GP, SS-SOMA-PR, SS-SOMA-RBF, AND GS-SOMA

Optimization Algorithm	Statistical Values				
	Mean	Std. Dev.	Median	Best	Worst
GA	1.51e+03	5.52e+01	1.52e+03	1.40e+03	1.58e+03
SS-SOMA-GP	1.26e+03	1.88e+02	1.22e+03	1.03e+03	1.54e+03
SS-SOMA-PR	1.07e+03	1.07e+02	1.04e+03	9.42e+02	1.29e+03
SS-SOMA-RBF	1.12e+03	1.16e+02	1.15e+03	9.59e+02	1.28e+03
GS-SOMA	1.01e+03	7.85e+01	9.53e+02	9.09e+02	1.51e+03

TABLE XV

RESULT OF t-TEST WITH 95% CONFIDENCE LEVEL COMPARING STATISTICAL VALUES FOR GS-SOMA AND THOSE OF SS-SOMA-GP, SS-SOMA-PR, SS-SOMA-RBF, SS-SOMA-PERFECT ON F1−F10 ($s+$, $s-$, AND $\approx$ INDICATES THAT GS-SOMA IS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, AND INDIFFERENT, RESPECTIVELY)

	GA	SS-SOMA-GP	SS-SOMA-PR	SS-SOMA-RBF	SS-SOMA-Perfect
F1	$s+$	$s+$	$s-$	$s+$	$s+$
F2	$s+$	$s+$	$\approx$	$s+$	$s-$
F3	$s+$	$s-$	$s+$	$\approx$	$s-$
F4	$s+$	$\approx$	$\approx$	$s-$	$s+$
F5	$s+$	$\approx$	$\approx$	$s+$	$s+$
F6	$s+$	$s+$	$s+$	$s+$	$s+$
F7	$s+$	$s+$	$s+$	$\approx$	$s+$
F8	$s+$	$s+$	$s+$	$\approx$	$s+$
F9	$s+$	$\approx$	$s-$	$s+$	$s+$
F10	$s+$	$s+$	$\approx$	$s+$	$s+$

GS-SOMA outperforms or is competitive to the SS-SOMAs on 44/50 cases. On the remaining six cases, GS-SOMA also displays solution qualities close to that of the superior SS-SOMA, see the highlighted results in Tables V–XIV. Note that this is a significant achievement considering that no *a priori* knowledge is available to select an appropriate surrogate modeling scheme for the problems considered. This highlights the reliability of the generalized framework.

The search convergence trends of GS-SOMA, SS-SOMA-AV, and SS-SOMA-Perfect are also plotted in Fig. 4. Note that SS-SOMA-AV represents the estimated performance one might expect to get when an approximation technique is randomly chosen for use. Hence, SS-SOMA-AV is generated from the average of the results obtained by all three SS-SOMAs, i.e., SS-SOMA-GP, SS-SOMA-PR, and SS-SOMA-RBF. It is evident from the search convergence trends that GS-SOMA is superior over SS-SOMA-AV on the 10 benchmark problems. This indicates that the generalized framework is more reliable when one has no knowledge about the suitability of the approximation scheme for the problem at hand.

2) *Analyzing the Generalized Evolutionary Framework in Single-Objective Optimization:* To gain a better understanding of the generalized framework, we further analyze the reliability and effectiveness of the ensemble (M_1) and smoothing (M_2) surrogate models in contributing to the evolutionary search.

To facilitate the analysis, the normalized root mean square errors (N-RMSE) of fitness predictions based on the ensemble surrogate model, i.e., M_1 in GS-SOMA search, for the benchmark problems are presented in Fig. 5. The N-RMSE of model i is determined as follows:

$$\text{Normalized RMSE}_i = \frac{\text{RMSE}_i}{\sum_{j=1}^{n} \text{RMSE}_j} \tag{19}$$

where n is the total approximation methods used in shaping the ensemble. From this figure, the consistently low N-RMSE of the ensemble model generated in the GS-SOMA search across all benchmark problems demonstrates the high reliability of the fitness prediction generated by M_1 across the different optimization problems over any single surrogates.

Further, it is worth noting that the use of M_2 contributes to the fitness improvement in GS-SOMA, which confirms

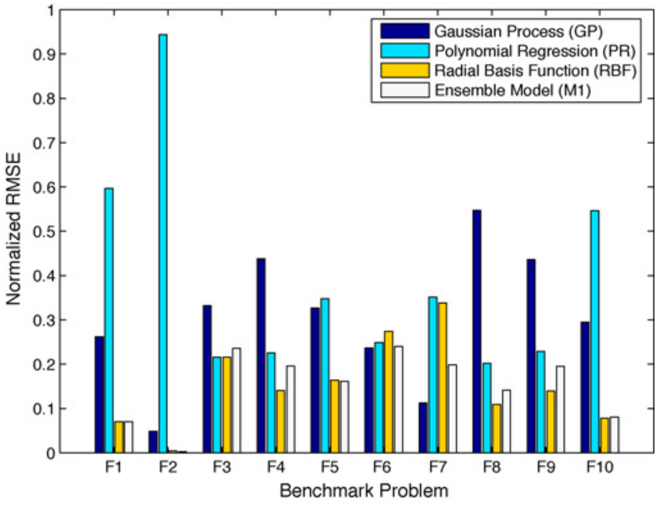


Fig. 5. Normalized RMSE by GP, PR, RBF, and weighted average ensemble.

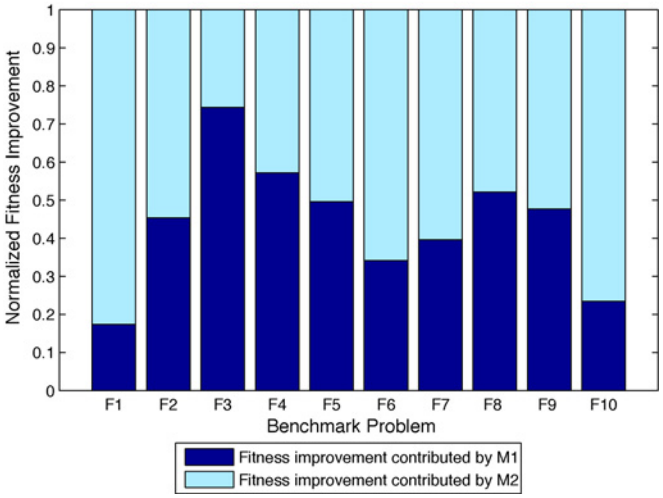


Fig. 6. Normalized fitness improvement during the runs of GS-SOMA contributed by M_1 (Imp_{M_1}) and M_2 (Imp_{M_2}).

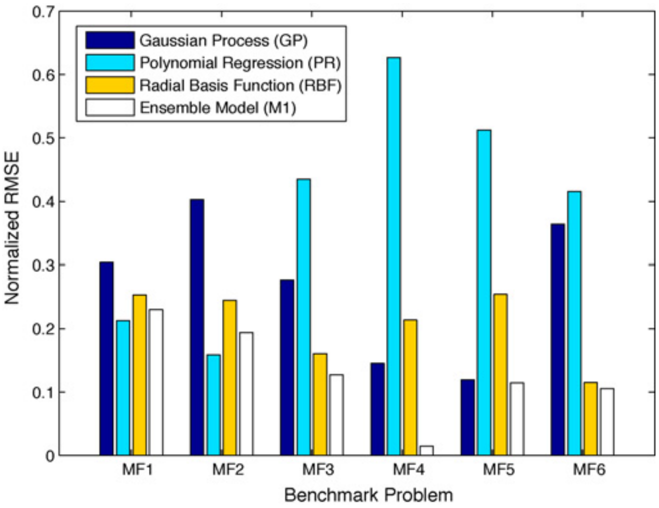


Fig. 7. Normalized RMSE by GP, PR, RBF, and weighted average ensemble on MF1–MF6.

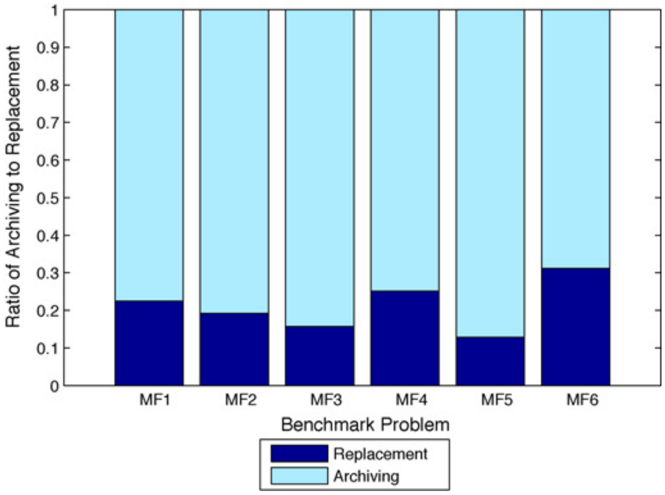


Fig. 8. Archiving to replacement ratio of GS-MOMA on MF1–MF6.

the possible benefits of bless of uncertainty in surrogate model. The normalized average fitness improvement of the local searches contributed via the use of M_1 (Imp_{M_1}) and M_2 (Imp_{M_2}) during the GS-SOMA searches are summarized in Fig. 6 and is defined by

$$\text{Normalized Imp}_{M_1} = \frac{\text{Imp}_{M_1}}{\text{Imp}_{M_1} + \text{Imp}_{M_2}}$$

$$\text{Normalized Imp}_{M_2} = \frac{\text{Imp}_{M_2}}{\text{Imp}_{M_1} + \text{Imp}_{M_2}}. \tag{20}$$

Imp_{M_1} is the total fitness improvements attained by local refinements, i.e., through Lamarckian learning, when $f(\mathbf{x}^1_{\text{opt}}) < f(\mathbf{x}^2_{\text{opt}})$, while Imp_{M_2} is the total fitness improvements when $f(\mathbf{x}^2_{\text{opt}}) < f(\mathbf{x}^1_{\text{opt}})$.

From the statistical results given in Fig. 6, it is notable that M_1 and M_2 surrogates have contributed to the surrogate-assisted memetic search in their unique ways. This provides a means for explaining the results that were obtained in Fig. 4 and Tables V–XIV. In particular, the reason for the fact that all surrogate-assisted SOMAs outperform SS-SOMA-Perfect on F1 (Ackley) suggests the presence of *'bless of uncertainty'* through the use of surrogate(s), since the notion of 'curse or bless of uncertainty' cannot exist in the latter. Further, the fact that SS-SOMA-PR is the most superior on F1 (Ackley) highlights the strength of the PR model in contributing to the search via smoothing the rugged landscape of the Ackley function. This hypothesis is clearly supported by the large portion of fitness improvements that are contributed by $M2$ (i.e., the PR model) on F1, see Fig. 6. On the other hand, neither SS-SOMAs nor GS-SOMA manage to outperform the SS-SOMA-Perfect on F3(Rosenbrock), suggesting the presence of *'curse of uncertainty'* due to the surrogate(s). Further, the results in F3 of Fig. 6 also indicate that M_2 (i.e., the smoothing PR model) did not contribute significantly to the search since the problem landscape of this function is originally smooth. Rather, the use of ensemble model in GS-SOMA had contributed to reliable fitness improvement on F3(Rosenbrock) by generating reliable prediction accuracy. On the other test problems, both M_1 and M_2 surrogates were shown to contribute significantly to GS-SOMA in their own unique ways.

C. Multiobjective Optimization

In this subsection, we present the empirical study of the GS-MOMA on six moderate- to high-dimensional MO benchmark problems, labeled here as MF1–MF6 [64]. The MO benchmark problems used in the study are summarized in Table XVI.

Performance comparisons are then made between the standard nondominated sorting genetic algorithm-II (NSGA-II) [65] and variants of MOMA. For fair comparison, we compare GS-MOMA with several SS-MOMAs and the NSGA-II since the formers are demonstrated with NSGA-II as the baseline by building on top of it. Hence, all algorithms compared inherit the same evolutionary operators as the NSGA-II used in our experiment. In SS-MOMAs, an offspring will be replaced in the spirit of Lamarckian learning during local search if its aggregated fitness function is found to be better than the original offspring. Similarly, SS-MOMA-Perfect is introduced here to assess the effects of approximation error on surrogate-assisted evolutionary search performance. For the sake of brevity, the notations and definitions of the MO algorithms studied are tabulated in Table XVII while the common parameter settings of the MO algorithms used in the experimental study are defined in Table XVIII.[6]

Many performance indicators exist for assessing the performance of MOEAs, such as those summarized in [66], [67]. Here, the following three performance indicators are used.

1) *Generational Distance* (GD) [68], [69]: This measurement indicates the gap between the true Pareto front (PF^*) and the evolved Pareto front (PF). Mathemati-

[6] Since MF3 and MF4 have higher dimensionality, i.e., $d = 50$, greater initial database size is required. For these cases, G_{db} is set to 20.

TABLE XVI
MULTIOBJECTIVE BENCHMARK PROBLEMS (MF1–MF6). PARAMETRIC DOMAIN USED IS $[0, 1]^d$, WHERE d IS THE PROBLEM DIMENSIONALITY CONSIDERED IN THIS PAPER

Benchmark Function	Formulation	Characteristics
MF1 ($d = 30$)	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - \sqrt{f_1(\mathbf{x})/g(\mathbf{x})}]$ $g(\mathbf{x}) = 1 + 9(\sum_{i=2}^{d} x_i)/(d-1)$	Convex, 2-objective Pareto front
MF2 ($d = 30$)	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - f_1(\mathbf{x})/g(\mathbf{x})^2]$ $g(\mathbf{x}) = 1 + 9(\sum_{i=2}^{d} x_i)/(d-1)$	Nonconvex, 2-objective Pareto front
MF3 ($d = 50$)	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - \sqrt{f_1/g} - (f_1/g)sin(10\pi f_1)]$ $g(\mathbf{x}) = 1 + 9(\sum_{i=2}^{d} x_i)/(d-1)$	Convex, disconnected, 2-objective Pareto front
MF4 ($d = 50$)	$f_1(\mathbf{x}) = 1 - \exp(-4x_1)sin^6(6\pi x_1)$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2]$ $g(\mathbf{x}) = 1 + 9[\sum_{i=2}^{d} x_i/(d-1)]^{0.25}$	Nonconvex, 2-objective Pareto front
MF5 ($d = 20$)	$f_1(\mathbf{x}) = cos(\frac{\pi}{2}x_1)cos(\frac{\pi}{2}x_2)(1 + g(\mathbf{x}))$ $f_2(\mathbf{x}) = cos(\frac{\pi}{2}x_1)sin(\frac{\pi}{2}x_2)(1 + g(\mathbf{x}))$ $f_3(\mathbf{x}) = cos(\frac{\pi}{2}x_1)(1 + g(\mathbf{x}))$ $g(\mathbf{x}) = \sum_{i=3}^{d}(x_i - x_1)^2$	Nonconvex, 3-objective, Pareto front
MF6 ($d = 10$)	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - \sqrt{f_1(\mathbf{x})/g(\mathbf{x})}]$ $g(\mathbf{x}) = 1 + 10(d-1) + \sum_{i=2}^{d}(x_i^2 - 10\cos(4\pi x_i))$	Convex, 2-objective, multiple local Pareto front

cally, it can be formulated as

$$GD = \frac{1}{n_{PF}}\sqrt{\sum_{i=1}^{n_{PF}} {d_i}^2} \tag{21}$$

where n_{PF} is the number of members in PF, d_i is the Euclidean distance (in objective space) between member i of PF and its nearest member in PF^*. A low value of GD is more desirable since it reflects a good convergence to the true Pareto fronts.

2) *Maximum Spread* (*MS*) [70]: It is used to measure how well the true Pareto front (PF^*) is covered by the evolved Pareto front (PF). The MS measurement used in this paper is formulated as

$$MS = \sqrt{\frac{1}{r}\sum_{i=1}^{r}\left[\frac{\min(f_i^{\max}, F_i^{\max}) - \max(f_i^{\min}, F_i^{\min})}{F_i^{\max} - F_i^{\min}}\right]^2} \tag{22}$$

where $f_i^{\max}$ and $f_i^{\min}$ are the maximum and minimum of the ith objective in the evolved PF, respectively. $F_i^{\max}$ and $F_i^{\min}$ are the maximum and minimum of the ith objective in PF^*, respectively. Higher value of MS reflects a larger area of PF^* covered by PF, which is desirable.

3) *Hypervolume Ratio* (*HR*) [69]: This indicates the ratio between the hyperarea or hypervolume (H) [71] dominated by the evolved PF and PF^*, where HR is defined as

$$HR = \frac{H(PF)}{H(PF^*)}$$
$$H = volume\left(\bigcup_{i=1}^{n_{PF}} v_i\right). \tag{23}$$

Here, v_i denotes the hypercube constructed from member i of a particular Pareto front and the reference point. A HR value close to 1 indicates that the evolved Pareto front is quite close to the true Pareto front, in both convergence and spread of solutions.

1) *Experimental Results:* The obtained Pareto fronts of the benchmark problems for 20 independent runs are combined and depicted in Figs. 9–14. The respective performance metrics are then summarized in Figs. 15–20. From these results, all surrogate-assisted multiobjective EAs, i.e., SS-MOMAs and GS-MOMA, are shown to outperform the standard NSGA-II on MF1, MF2, MF5, and MF6. MF6 (ZDT4) is generally regarded as a challenging problem and hence commonly used

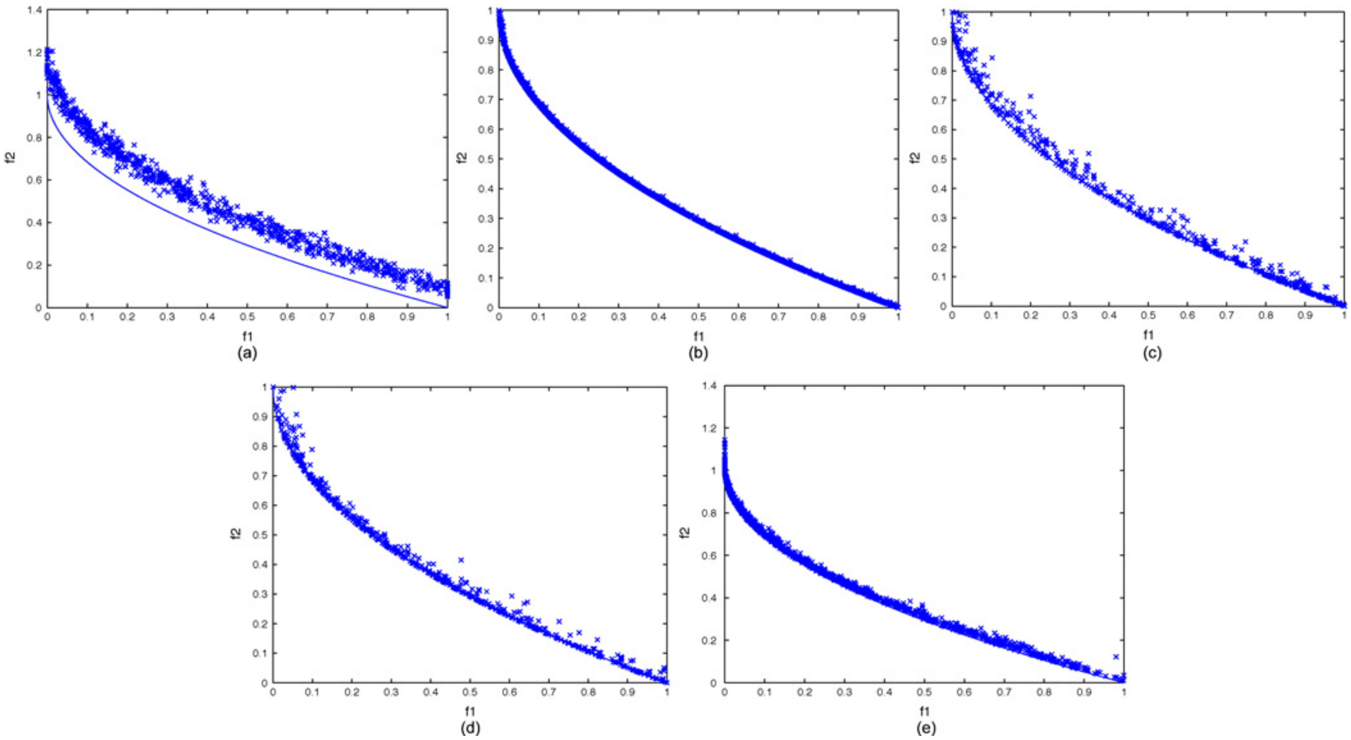


Fig. 9. Pareto front evolved for benchmark problem MF1 in (a) NSGA-II, (b) GS-MOMA, (c) SS-MOMA-I, (d) SS-MOMA-II, and (e) SS-MOMA-Perfect.

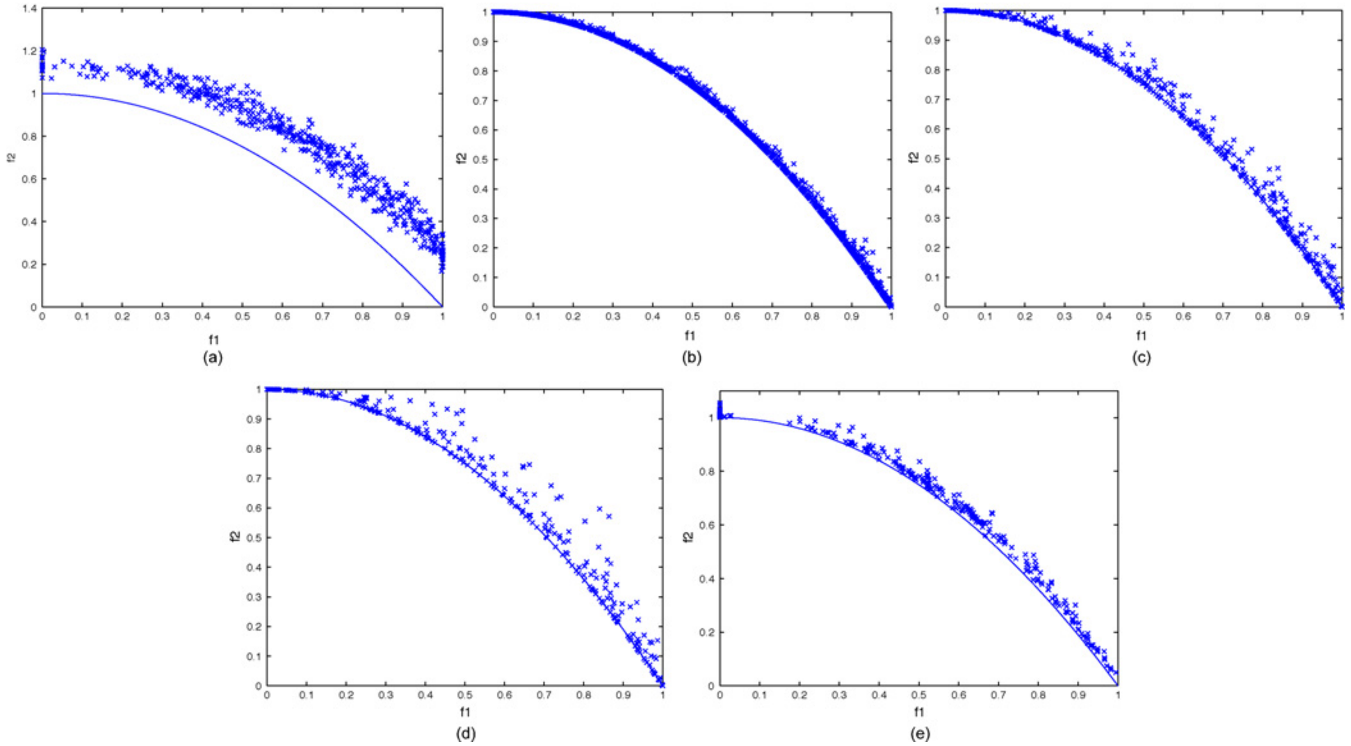


Fig. 10. Pareto front evolved for benchmark problem MF2 in (a) NSGA-II, (b) GS-MOMA, (c) SS-MOMA-I, (d) SS-MOMA-II, and (e) SS-MOMA-Perfect.

TABLE XVII

DEFINITION OF THE MULTIOBJECTIVE MAs (MOMAs) COMPARED

Algorithms	Definition
NSGA-II	No surrogate is used
GS-MOMA	Generalized surrogate MOMA with M_1: *Weighted-average* ensemble of GP, PR, and RBF M_2: PR
SS-MOMA-I	Single surrogate MOMA with M_1: Ensemble of GP, PR, and RBF
SS-MOMA-II	Single surrogate MOMA with M_1: PR
SS-MOMA-Perfect	Single surrogate MOMA with M_1: Perfect model

TABLE XVIII

SETTING OF EXPERIMENTS FOR NSGA-II, GS-MOMA, AND SS-MOMA

Parameters Setting	
Population size (N_{pop})	100
Crossover probability (P_{cross})	0.9
Mutation probability (P_{mut})	0.1
Maximum number of exact evaluations	MF1–MF2: 8000 MF3–MF4: 16 000 MF5: 30 000 MF6: 20 000
Evolutionary operators	Simulated binary crossover, polynomial mutation, binary tournament selection, elitism, nondomination rank, and crowded distance
Number of trust-region iteration (k_{term}) for SS-MOMA and GS-MOMA	2
Database building phase (G_{db}) for SS-MOMA and GS-MOMA (in number of generations)	MF1–MF2, MF5–MF6: 10 MF3–MF4: 20
Number of independent runs	20

TABLE XIX

RADIAL BASIS KERNELS

Linear splines	$\|\|\mathbf{x} - \mathbf{c}_i\|\|$
Thin plate splines	$\|\|\mathbf{x} - \mathbf{c}_i\|\|^k ln\|\|\mathbf{x} - \mathbf{c}_i\|\|$
Cubic splines	$\|\|\mathbf{x} - \mathbf{c}_i\|\|^3$
Gaussian	$exp - \frac{\|\|\mathbf{x}-\mathbf{c}_i\|\|^2}{\beta_i}$
Multiquadrics	$\sqrt{1 + \frac{\|\|\mathbf{x}-\mathbf{c}_i\|\|^2}{\beta_i}}$
Inverse multiquadrics	$(1 + \frac{\|\|\mathbf{x}-\mathbf{c}_i\|\|^2}{\beta_i})^{-\frac{1}{2}}$

by many in the literature. Here, we validate our results on ZDT4 against those obtained by Deb *et al.* [28]. While [28] reported to solve ZDT4 with from 21 781 to 22 730 exact function evaluations with an achieved spread measure[7] of 0.332 to 0.422, GS-MOMA requires only 20 000 exact evaluations at a competitive spread measure of 0.410 ± 0.046. On MF3 and MF4, some SS-MOMAs perform competitively or slightly poorer than NSGA-II [see Figs. 11(d) and 12(d)]. On the other hand, GS-MOMA searches more efficiently than all the SS-MOMA variants and NSGA-II on the six benchmark problems considered. Note that GS-MOMA also outperforms the SS-MOMA-Perfect on a majority of the MOO benchmarks with respect to all three performance metrics, thus suggesting the positive synergy of the ensemble and smoothing surrogate models in the GSM framework.

2) *Analyzing the Generalized Evolutionary Framework in Multiobjective Optimization:* To arrive at better understanding of the generalized framework in the context of multiobjective optimization, we analyze next the reliability and effectiveness of the ensemble (M_1) and smoothing (M_2) surrogate models in contributing to evolutionary search.

The N-RMSE, i.e., see (19), of fitness predictions based on GP, PR, RBF, or ensemble in GS-MOMA is summarized in Fig. 7. From the results, the ensemble model M_1 is shown to arrive at low N-RMSE on all the multiobjective test problems considered, which is consistent with observations obtained in the single-objective context. M_1 generates high-reliability predictions in comparison to the other single surrogate model counterparts, i.e., GP, PR or RBF.

Besides N-RMSE, the *solution archiving to replacement ratio*, labeled here as Γ, of the GS-MOMA search is also reported in Fig. 8. Γ indicates the degree of solution diversity (through archival of new nondominating solutions) against search convergence (through the process of Lamarckian learning replacement) in the GS-MOMA search. While Lamarckian learning helps to speedup convergence toward the desired Pareto front, the large Γ ratio observed on all benchmark problems implies frequent discovery of potential nondominating solutions when using both M_1 and M_2 with local refinements. This suggests *'bless of uncertainty'* may take the form of faster search convergence and better solution diversity in the context of multiobjective evolutionary search.

D. Computational Complexity of GSM Framework

In this subsection, we present an analytical study on the computational complexity of the GSM framework. The computational effort, referred here by T_{comp}, of GS-SOMA or GS-MOMA is formulated as follows:

$$T_{\text{comp}} = G_{\text{db}} N_{\text{pop}} \sum_{i=1}^{r} F_i + (G_{\max} - G_{\text{db}})$$
$$[N_{\text{pop}}(T_{\text{ens}} + T_{\text{PR}} + 2k_{\text{term}} \sum_{i=1}^{r} F_i + T_{\text{overhead}})] \quad (24)$$

where

G_{db}	number of standard SO/MOEA search generations configured for building the database of training data points at the initial search phase of the GSM framework;
$G_{\max}$	maximum number of search generations;
N_{pop}	population size;
r	number of objectives to optimize;
k_{term}	number of iterations made in the trust-region-regulated local searches;
F	original/exact function evaluation cost;
T_{ens}	time to build M_1, i.e., the ensemble model;
T_{PR}	time to build M_2, i.e., the polynomial regression model, which is not applicable if PR is already built when constructing M_1;

[7] The spread metric [72] considers the distance between two extreme ends of Pareto front as well as the uniformity of distribution for solutions between the two extremes. This metric may be used for measuring the diversity of converged Pareto fronts. Note that a lower spread metric is desirable.

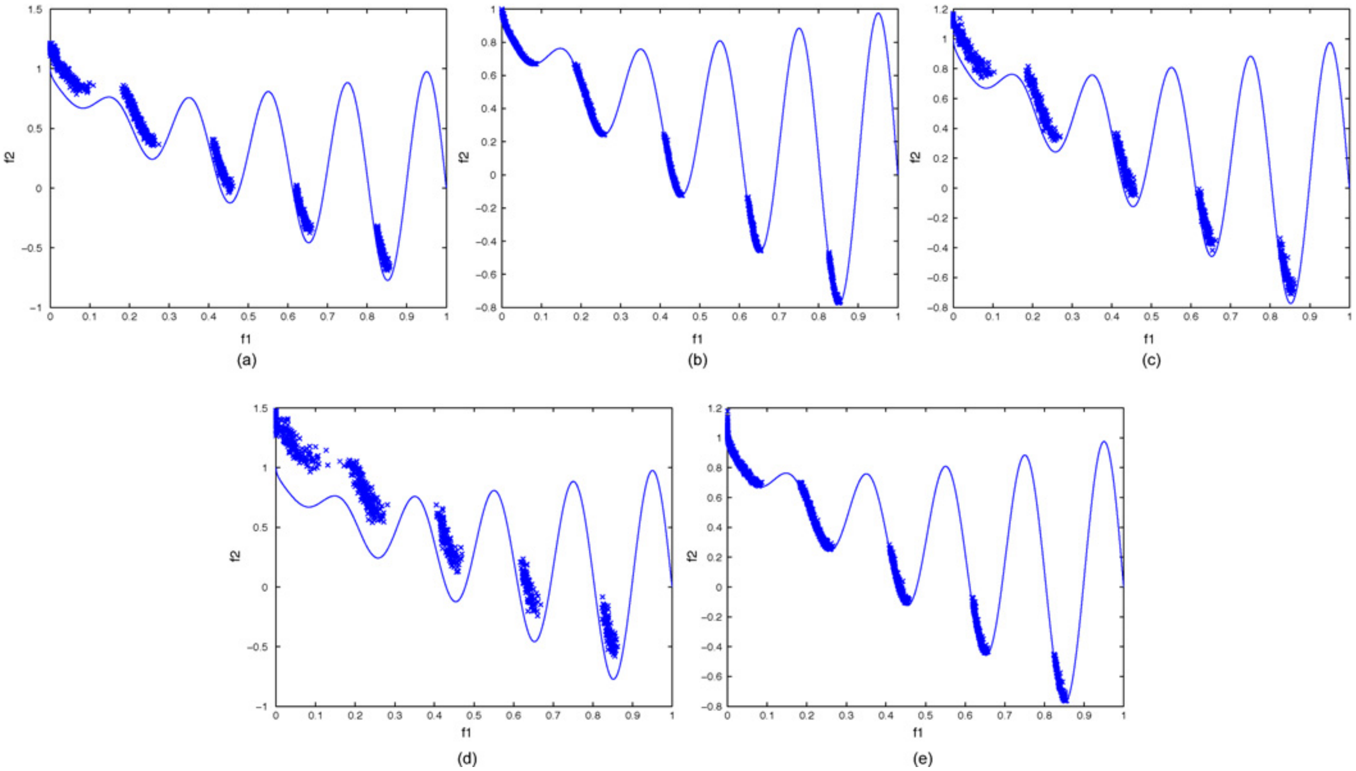


Fig. 11. Pareto front evolved for benchmark problem MF3 in (a) NSGA-II, (b) GS-MOMA, (c) SS-MOMA-I, (d) SS-MOMA-II, and (e) SS-MOMA-Perfect.

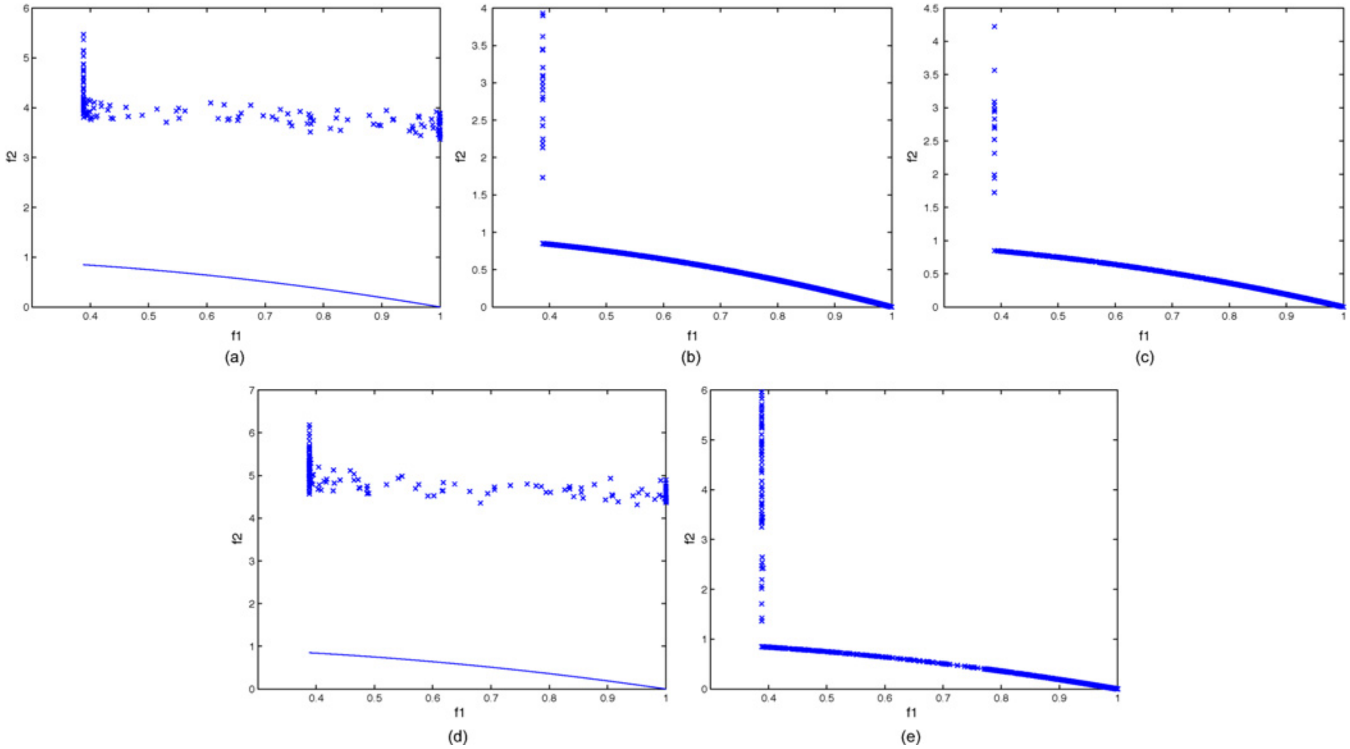


Fig. 12. Pareto front evolved for benchmark problem MF4 in (a) NSGA-II, (b) GS-MOMA, (c) SS-MOMA-I, (d) SS-MOMA-II, and (e) SS-MOMA-Perfect.

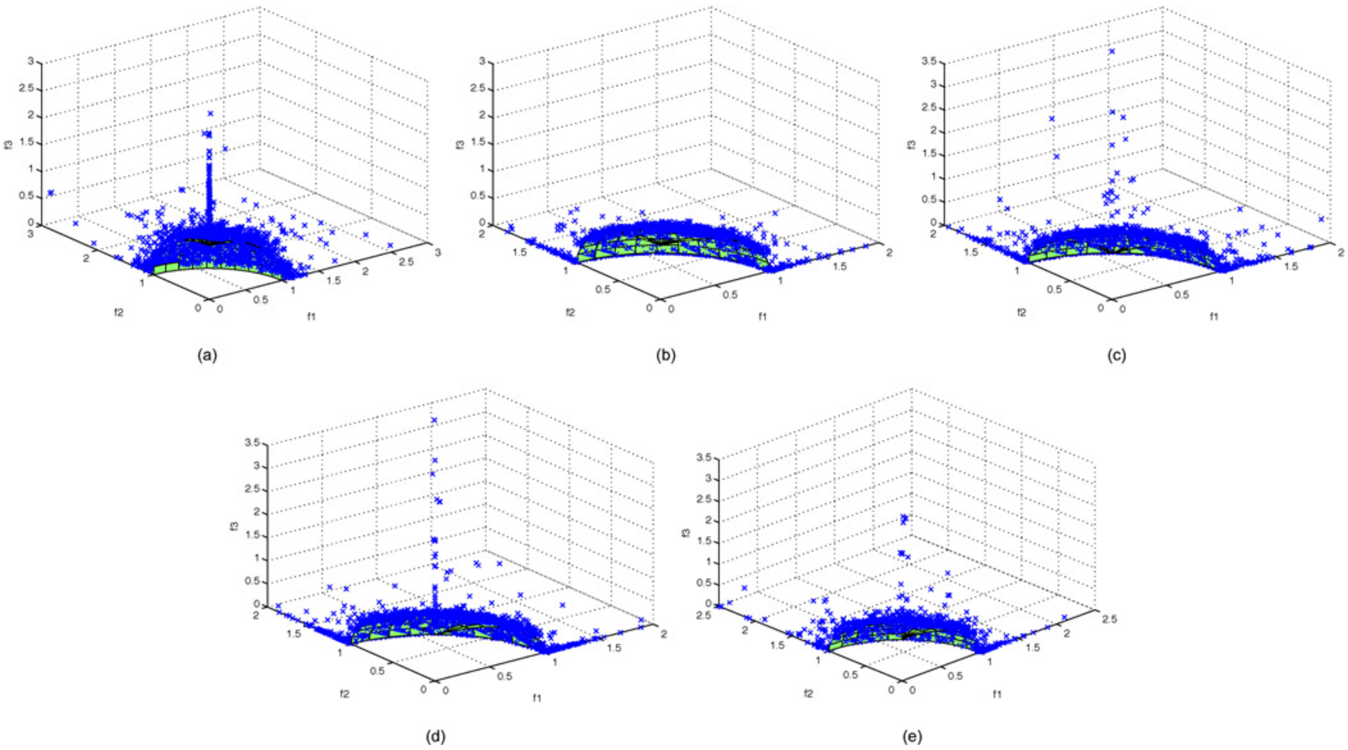


Fig. 13. Pareto front evolved for benchmark problem MF5 in (a) NSGA-II, (b) GS-MOMA, (c) SS-MOMA-I, (d) SS-MOMA-II, and (e) SS-MOMA-Perfect.

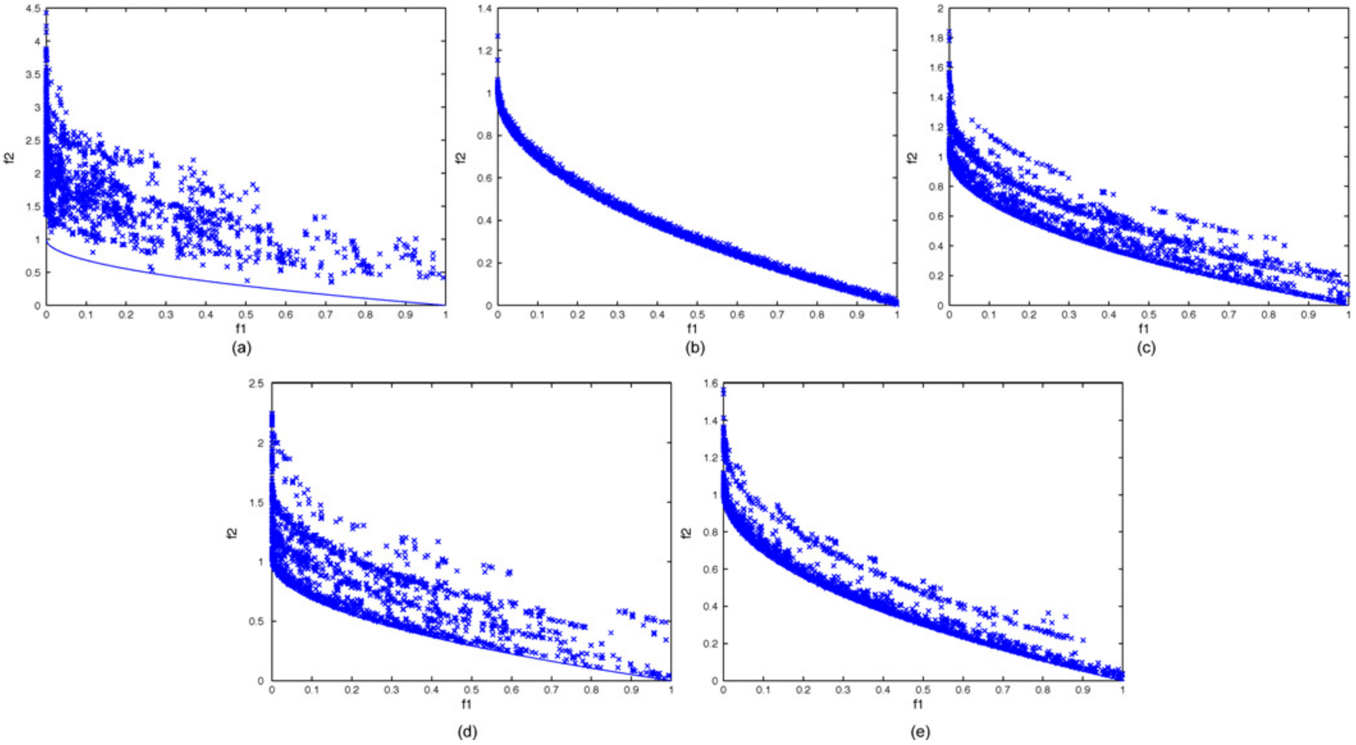
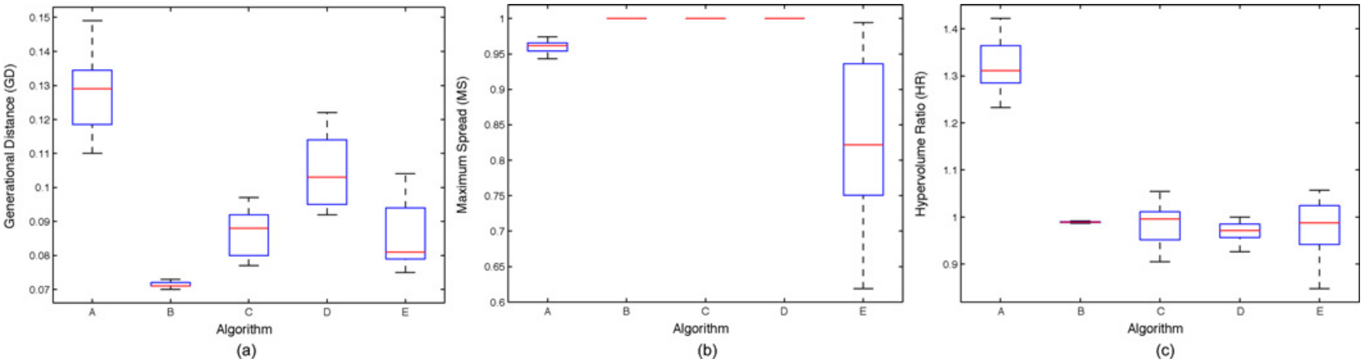
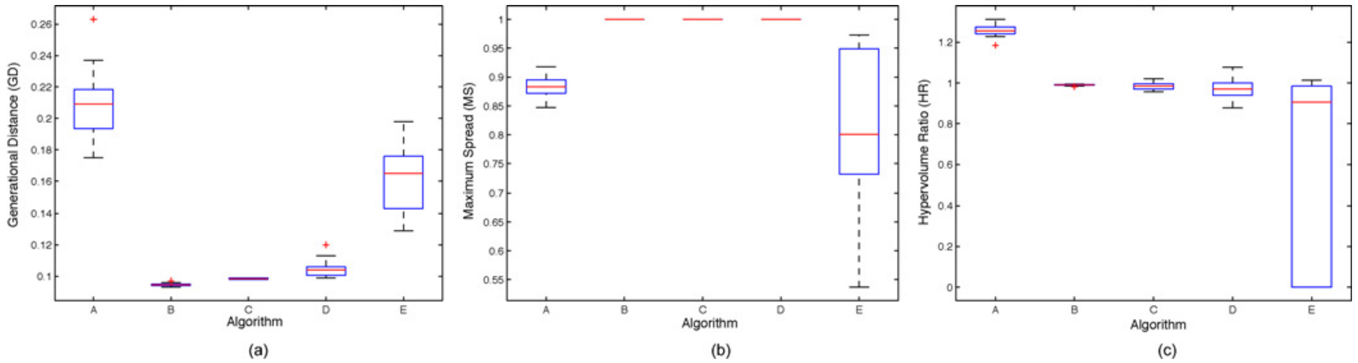


Fig. 14. Pareto front evolved for benchmark problem MF6 in (a) NSGA-II, (b) GS-MOMA, (c) SS-MOMA-I, (d) SS-MOMA-II, and (e) SS-MOMA-Perfect.



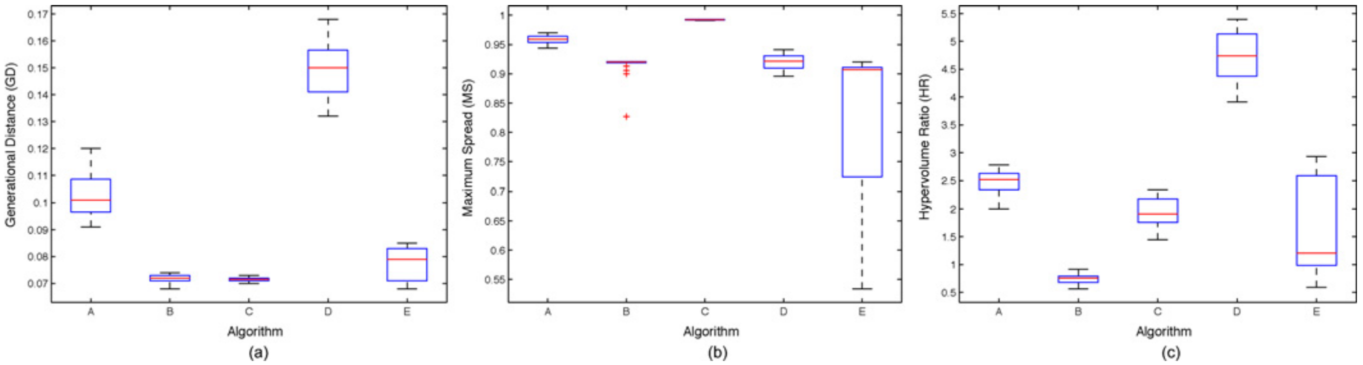
(a) (b) (c)

Fig. 15. Performance metrics for benchmark problem MF1. (a) Generational distance (GD). (b) Maximum spread (MS). (c) Hypervolume ratio (HR). (A:NSGA-II, B:GS-MOMA, C:SS-MOMA-I, D:SS-MOMA-II, E:SS-MOMA-Perfect.)



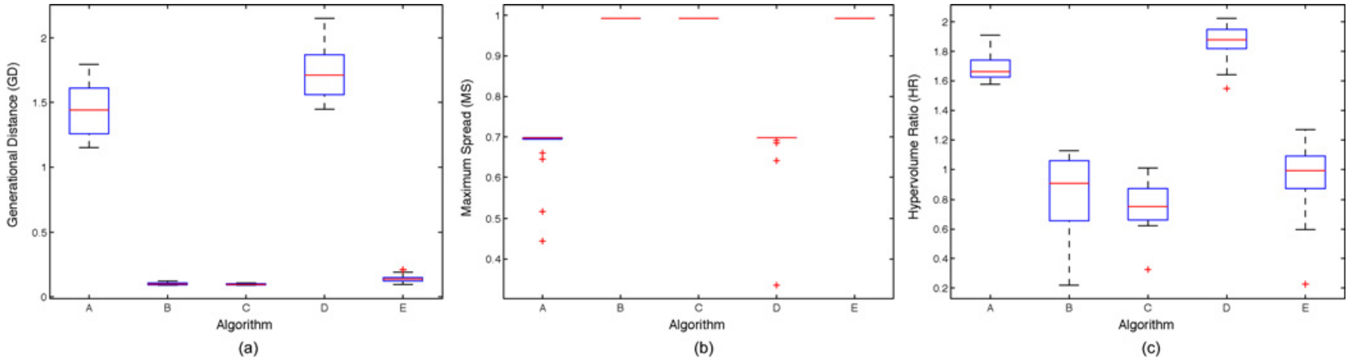
(a) (b) (c)

Fig. 16. Performance metrics for benchmark problem MF2. (a) Generational distance (GD). (b) Maximum spread (MS). (c) Hypervolume ratio (HR). (A:NSGA-II, B:GS-MOMA, C:SS-MOMA-I, D:SS-MOMA-II, E:SS-MOMA-Perfect.)



(a) (b) (c)

Fig. 17. Performance metrics for benchmark problem MF3. (a) Generational distance (GD). (b) Maximum spread (MS). (c) Hypervolume ratio (HR). (A:NSGA-II, B:GS-MOMA, C:SS-MOMA-I, D:SS-MOMA-II, E:SS-MOMA-Perfect.)



(a) (b) (c)

Fig. 18. Performance metrics for benchmark problem MF4. (a) Generational distance (GD). (b) Maximum spread (MS). (c) Hypervolume ratio (HR). (A:NSGA-II, B:GS-MOMA, C:SS-MOMA-I, D:SS-MOMA-II, E:SS-MOMA-Perfect.)

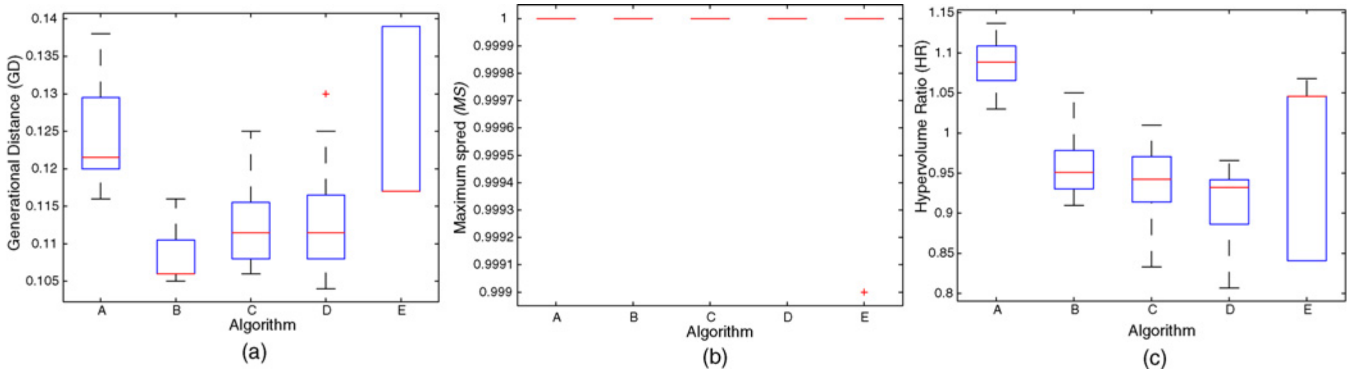


Fig. 19. Performance metrics for benchmark problem MF5. (a) Generational distance (GD). (b) Maximum spread (*MS*). (c) Hypervolume ratio (*HR*). (A:NSGA-II, B:GS-MOMA, C:SS-MOMA-I, D:SS-MOMA-II, E:SS-MOMA-Perfect.)

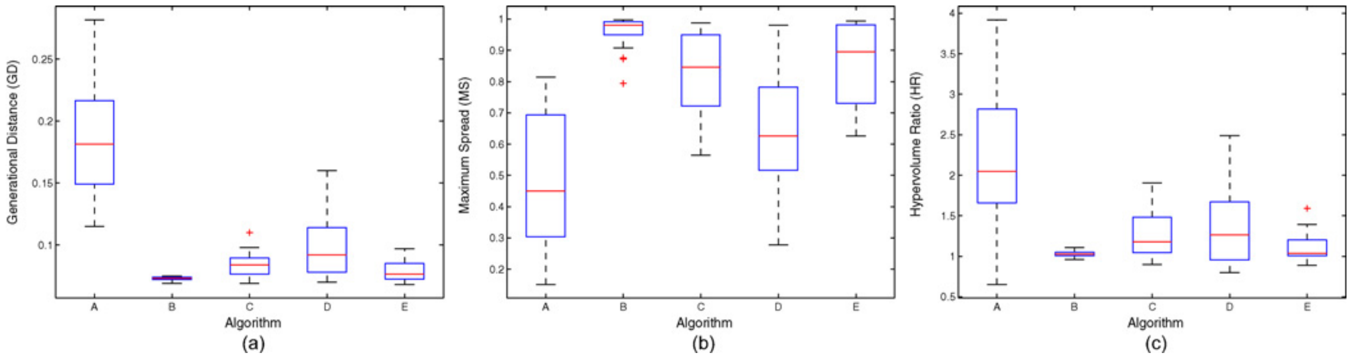


Fig. 20. Performance metrics for benchmark problem MF6. (a) Generational distance (GD). (b) Maximum spread (*MS*). (c) Hypervolume ratio (*HR*). (A:NSGA-II, B:GS-MOMA, C:SS-MOMA-I, D:SS-MOMA-II, E:SS-MOMA-Perfect.)

T_{overhead} other additional costs such as for fitness predictions and finding nearest points, which are often negligible.

On the other hand, the computational cost for SS-SOMA or SS-MOMA variants is

$$T_{\text{comp}} = G_{\text{db}} N_{\text{pop}} \textstyle\sum_{i=1}^{r} F_i + (G_{\max} - G_{\text{db}}) \\ [N_{\text{pop}}(T_m + k_{\text{term}} \textstyle\sum_{i=1}^{r} F_i + T_{\text{overhead}})] \quad (25)$$

where T_m is the time taken to build the particular surrogate model used.

Although there are several elements in (24) and (25), it is worth noting that when working with computationally expensive problems, the most significant part contributing to the total computational effort incurred is F. Hence, when F is significantly large, which is assumed to be fulfilled in any surrogate-assisted optimization framework, T_{ens}, T_{PR}, T_{overhead} and T_m are generally considered to be negligible, otherwise such frameworks should never be used.

V. Conclusion

With a plethora of approximation/surrogate modeling approaches available in the literature, the choice of technique to use greatly affects the performance of surrogate-assisted evolutionary searches. It is argued that every approximation technique introduces some unique characteristics suitable for modeling some classes of problems accurately but not for others. Given that *a priori* knowledge about the problem landscape is often scarce, the ability to tackle new problems in a reliable way is of significant value. This paper has investigated a generalized framework that unifies diverse surrogate models synergistically in the memetic evolutionary search. In contrast to existing studies, the studied memetic framework emphasizes not only on 1) mitigating the impact of *'curse of uncertainty'* robustly, but also 2) benefitting from the *'bless of uncertainty,'* through the use of ensemble and landscape smoothing surrogate models, respectively.

The core purpose of proposing any new search strategies, including the GSM framework, is to solve real-world optimization problems more robustly, effectively and/or efficiently. Hence, to facilitate possible systematic study and gain deeper understanding of the proposed methods for solving complex real-world problems plagued with computationally expensive functions, benchmark problems of diverse known properties have been employed. In this paper, we have presented extensive numerical studies on commonly used single/multiobjective optimization benchmark problems which have demonstrated the competitiveness of the generalized framework. Overall, the ensemble model is shown to be capable of attaining reliable, accurate surrogate models, while smoothing model speeds up evolutionary search performance by traversing through the multimodal landscape of complex problems.

Statistically, the generalized framework achieved significantly better performance on SOO/MOO when compared to SS-SOMA/MOMA and their underlying SO/MOEA.

Presently, the GSM framework is used for solving real-world problems plagued with computationally expensive functions, particularly in the field of aerodynamic and molecular structural designs. Based on our experiences with both benchmark and real-world problems that range from turbine blade [7], [20] to airfoil designs [8], [11], [22], [32], the observations obtained from the use of benchmark problems do not deviate significantly from those in the real-world problems we have experimented. Some of the observations and problems we have noted when dealing with real-world problems are listed as follows.

1) In contrast to benchmark problems, the time taken to collect adequate amount of database points when dealing with real-world problems can be relatively significant if unsupported by sufficient machines capability. A possible solution is to directly utilize an external database of previously evaluated design points, if available, instead of building the database from scratch in the initial G_{db} generations of evolutionary optimization. When existing database are unavailable, or the design points available are insufficient for building reliable surrogates, a smaller G_{db} can be used to obtain the initial design points necessary for the reliable surrogate building to facilitate time saving.
2) When parallel machines capability is available, multi-level parallelization can be leveraged through the GSM framework, namely, 1) generation level, i.e., individuals at the same generation are sent to multiple computing nodes for evaluation; 2) individual level, independent local searches utilizing M_1 and M_2 respectively, are executed in parallel. Hence, further acceleration can be expected.

ACKNOWLEDGMENT

D. Lim and Y.-S. Ong would like to thank the members of Nanyang Technological University, Singapore, for providing the computing resources.

APPENDIX A
APPROXIMATION/SURROGATE MODELING TECHNIQUES

Here, we provide a brief review on three different surrogate modeling techniques used in this paper, namely: Kriging/Gaussian process (GP), polynomial regression (PR), and radial basis function (RBF). Throughout this section, let $\mathcal{D} = \{\mathbf{x}_i, t_i\}$, $i = 1, \ldots, m$ denote the training dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ is an input design vector and $t_i \in \mathbb{R}$ is the corresponding target value.

A. Kriging/Gaussian Process (GP)

The GP surrogate model [55] assumes the presence of an unknown true modeling function $f(\mathbf{x})$ and an additive noise term v to account for anomalies in the observed data. Thus

$$t = f(\mathbf{x}) + v. \tag{26}$$

The standard analysis requires the specification of prior probabilities on the modeling function and the noise model. From a stochastic process viewpoint, the collection $\mathbf{t} = \{t_1, t_2, \ldots, t_m\}$ is called a Gaussian process if every subset of $\mathbf{t}$ has a joint Gaussian distribution. More specifically

$$P(\mathbf{t}|\mathbf{C}, \{\mathbf{x}_m\}) = \frac{1}{Z}\exp\left(-\frac{1}{2}(\mathbf{t}-\boldsymbol{\mu})^T\mathbf{C}^{-1}(\mathbf{t}-\boldsymbol{\mu})\right) \tag{27}$$

where $\mathbf{C}$ is a covariance matrix parameterized in terms of hyperparameters $\boldsymbol{\theta}$, i.e., $\mathbf{C}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j; \theta)$ and $\boldsymbol{\mu}$ is the process mean. The Gaussian process is characterized by this covariance structure since it incorporates prior beliefs both about the true underlying function as well as the noise model. In the present study, we use the following exponential covariance model:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp -(\mathbf{x}_i - \mathbf{x}_j)^T\Theta(\mathbf{x}_i - \mathbf{x}_j) + \theta_{d+1} \tag{28}$$

where $\Theta = \text{diag}\{\theta_1, \theta_2, \ldots, \theta_d\} \in \mathbb{R}^{d\times d}$ is a diagonal matrix of undetermined hyperparameters, and $\theta_{d+1} \in \mathbb{R}$ is an additional hyperparameter arising from the assumption that noise in the dataset is Gaussian (and output dependent). We shall henceforth use the symbol $\boldsymbol{\theta}$ to denote the vector of undetermined hyperparameters, i.e., $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_{d+1}\}$. In practice, the undetermined hyperparameters are tuned to the data using the evidence maximization framework. Once the hyperparameters have been estimated from the data, predictions can be readily made for a new testing point.

B. Polynomial Regression (PR)

In PR metamodeling technique [56], we define an exponent vector ε containing positive integers $(\pi_1, \pi_2, \ldots, \pi_d)$ and define $\mathbf{x}_i^{\varepsilon}$ as an exponent input vector $(x_{i_1}{}^{\pi_1}, x_{i_2}{}^{\pi_2}, \ldots, x_{i_d}{}^{\pi_d})$.

Given a set of exponent vectors $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_o$ and the set of data $(\mathbf{x}_i, t_i)$, where $i = 1, 2, \ldots, m$, the polynomial model of $(o-1)$th order has the form

$$\hat{t}_i = C_1\mathbf{x}_i^{\varepsilon_1} + C_2\mathbf{x}_i^{\varepsilon_2} + \cdots + C_m\mathbf{x}_i^{\varepsilon_o} \tag{29}$$

where $C_1, C_2, \ldots, C_o$ are the coefficient vectors to be estimated, and $C_j = (c_{j_1}, c_{j_2}, \ldots, c_{j_d})$, $j = 1, 2, \ldots, o$.

The least square method is then used to estimate the coefficients of the polynomial model. By definition, the least square error E to be minimized is

$$E = \sum_{i=1}^{m}[t_i - \hat{t}_i]^2. \tag{30}$$

It may be easily shown that $t_i = f(\mathbf{x}_i)$, and by multiplying both sides of (29) with $\mathbf{x}_i^{\varepsilon_j}$ and taking the sum of m pairs of input-output data, we arrive at

$$C_1\sum_i \mathbf{x}_i^{\varepsilon_1+\varepsilon_j} + \cdots + C_o\sum_i \mathbf{x}_i^{\varepsilon_o+\varepsilon_j} = \sum_i t_i\mathbf{x}_i^{\varepsilon_j}. \tag{31}$$

For $j = 1, 2, \ldots, o$, the polynomial model for the training dataset can be represented in the matrix notation as follows:

$$A\gamma^T = \boldsymbol{b}^T \tag{32}$$

where

$$A = \begin{bmatrix} \sum_i \mathbf{x}_i^{\varepsilon_1+\varepsilon_1} & \dots & \sum_i \mathbf{x}_i^{\varepsilon_1+\varepsilon_o} \\ \vdots & & \vdots \\ \sum_i \mathbf{x}_i^{\varepsilon_o+\varepsilon_1} & \dots & \sum_i \mathbf{x}_i^{\varepsilon_o+\varepsilon_o} \end{bmatrix} \tag{33}$$

$$\boldsymbol{b} = (\sum t_i \mathbf{x}_i^{\varepsilon_1}, \dots, \sum t_i \mathbf{x}_i^{\varepsilon_o}) \tag{34}$$

$$\gamma = (C_1, C_2, \dots, C_o). \tag{35}$$

Then the coefficient matrix of the polynomial is

$$\gamma = (A^{-1}\boldsymbol{b}^T)^T. \tag{36}$$

Let $B_i = (\mathbf{x}_i^{\varepsilon_1}, \dots, \mathbf{x}_i^{\varepsilon_o})$, the following equations may be derived:

$$A = \sum_i B_i^T B_i$$

$$\boldsymbol{b} = \sum_i t_i B_i$$

$$\hat{t}_i = \gamma . B_i^T .$$

The predicted output for a new input pattern is then given by $\hat{t}_i = \gamma . B_i^T$.

C. Radial Basis Function

The surrogate models of RBF used in this paper are interpolating radial basis function networks of the form

$$\hat{t} = \hat{f}(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(||\mathbf{x} - \mathbf{x}_i||) \tag{37}$$

where $K(||\mathbf{x} - \mathbf{x}_i||) : \mathbb{R}^d \rightarrow \mathbb{R}$ is a RBF and $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_m\} \in \mathbb{R}^m$ denotes the vector of weights. Hence, the number of hidden nodes in the RBF here is as many as the number of training points.

Typical choices for the kernel include linear splines, cubic splines, multiquadrics, thin-plate splines, and Gaussian functions [57]. Recent studies in [73], [74], indicate that the linear, cubic, and thin plate spline RBFs have better theoretical properties than the multiquadric and Gaussian RBFs. Hence, in this paper, we opt to use linear spline kernel function. The structure of some commonly used radial basis kernels and their parameterization are shown in Table XIX Given a suitable kernel, the weight vector can be computed by solving the linear algebraic system of equations $\mathbf{K}\boldsymbol{\alpha} = \mathbf{t}$, where $\mathbf{t} = \{t_1, t_2, \dots, t_m\} \in \mathbb{R}^m$ denotes the vector of outputs and $\mathbf{K} \in \mathbb{R}^{m \times m}$ denotes the Gram matrix formed using the training inputs (i.e., the ijth element of $\mathbf{K}$ is computed as $K(||\mathbf{x}_i - \mathbf{x}_j||)$).

Appendix B
Single-Objective Benchmark Functions

Single-objective benchmark functions used in this paper are presented in this section. The shifted and/or rotated functions are taken from [62] and [63]. Note that due to the long description for F7–F10, reader is referred directly to [63] for those functions. From F4–F6, the following nomenclature applies:

$\mathbf{o} = [o_1, o_2, \dots, o_d]$: the shifted global optimum
$\mathbf{M}$: linear transformation matrix, obtained from [63].

F1: Ackley

$$F(\mathbf{x}) = 20 + e - 20\exp\left(-0.2\sqrt{\tfrac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\tfrac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) \tag{38}$$
$$-32.768 \leq x_i \leq 32.768, \qquad i = 1, 2, \dots, d.$$

Global optimum $x_i^* = 0.0$ for $i = 1, \dots, d,\ F(\mathbf{x}^*) = 0.0$.

F2: Griewank

$$F(\mathbf{x}) = 1 + \textstyle\sum_{i=1}^{d} x_i^2/4000 - \prod_{i=1}^{d}\cos(x_i/\sqrt{i}) \tag{39}$$
$$-600 \leq x_i \leq 600, \qquad i = 1, 2, \dots, d.$$

Global optimum $x_i^* = 0.0$ for $i = 1, \dots, d,\ F(\mathbf{x}^*) = 0.0$.

F3: Rosenbrock

$$F(\mathbf{x}) = \textstyle\sum_{i=1}^{d-1}(100 \times (x_{i+1} - x_i^2)^2 + (1 - x_i)^2) \tag{40}$$
$$-2.048 \leq x_i \leq 2.048, \quad i = 1, 2, \dots, d.$$

Global optimum $x_i^* = 1.0$ for $i = 1, \dots, d,\ F(\mathbf{x}^*) = 0.0$.

F4: Shifted Rotated Rastrigin

$$F(\mathbf{x}) = \textstyle\sum_{i=1}^{d}(z_i^2 - 10cos(2\pi z_i) + 10) - 330 \tag{41}$$
$$\mathbf{z} = (\mathbf{x} - \mathbf{o}) * \mathbf{M},$$
$$-5 \leq x_i \leq 5, \qquad i = 1, 2, \dots, d.$$

Global optimum $\mathbf{x}^* = \mathbf{o},\ F(\mathbf{x}^*) = f_{bias} = -330$.

F5: Shifted Rotated Weierstrass

$$F(\mathbf{x}) = \textstyle\sum_{i=1}^{d}(\sum_{k=0}^{k_{\max}}[a^k cos(2\pi b^k(z_i + 0.5))]) - d\sum_{k=0}^{k_{\max}}[a^k cos(2\pi b^k .0.5)] + 90 \tag{42}$$
$$\mathbf{z} = (\mathbf{x} - \mathbf{o}) * \mathbf{M},$$
$$-0.5 \leq x_i \leq 0.5, \qquad i = 1, 2, \dots, d.$$

Global optimum $\mathbf{x}^* = \mathbf{o},\ F(\mathbf{x}^*) = f_{bias} = 90$. $a = 0.5$, $b = 3$, $k_{\max}$=20.

F6: Shifted Expanded Griewank Plus Rosenbrock

$$F(\mathbf{x}) = F_2(F_3(z_1, z_2)) + F_2(F_3(z_2, z_3)) + \dots + F_2(F_3(z_{d-1}, z_d)) + F_2(F_3(z_d, z_1)) - 130 \tag{43}$$
$$\mathbf{z} = \mathbf{x} - \mathbf{o} + \mathbf{1},$$
$$-3 \leq x_i \leq 1, \qquad i = 1, 2, \dots, d.$$

Global optimum $\mathbf{x}^* = \mathbf{o},\ F(\mathbf{x}^*) = f_{bias} = -130$.

F7: Hybrid Composition Function [63, F15].

F8: Rotated Hybrid Composition Function of F7 [63, F16].

F9: Rotated Hybrid Composition Function with Narrow Basin Global Optimum [63, F19].

F10: Noncontinuous Rotated Hybrid Composition Function [63, F23].

References

[1] C. G. Johnson and J. J. R. Caldalda, "Introduction: Genetic algorithms in visual art and music," *Leonardo*, vol. 35, no. 2, pp. 175–184, Apr. 2002.

[2] C. Aranha and H. Iba, "The memetic tree-based genetic algorithm and its application to portfolio," *Memetic Comput.*, vol. 1, no. 2, pp. 139–151, Jun. 2009.

[3] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008.

[4] S.M.K. Hasan, R. Sarker, D. Essam, and D. Cornforth, "Memetic algorithms for solving job-shop scheduling problems," *Memetic Comput.*, vol. 1, no. 1, pp. 69–83, Mar. 2009.

[5] E. W. Lameijer, T. Baeck, J. N. Kok, and A. P. Ijzerman, "Evolutionary algorithms in drug design," *Nat. Comput.*, vol. 4, no. 3, pp. 177–243, Sep. 2005.

[6] M. Olhofer, T. Arima, T. Sonoda, and B. Sendhoff, "Optimization of a stator blade used in a transonic compressor cascade with evolution strategies," *Adaptive Computing in Design and Manufacture* (ACDM). Berlin, Germany: Springer-Verlag, 2000, pp. 45–54.

[7] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness function," *IEEE Trans. Evol. Comput.*, vol. 6, no. 5, pp. 481–494, Oct. 2002.

[8] Y. S. Ong, P. B. Nair, and A. J. Keane, "Evolutionary optimization of computationally expensive problems via surrogate modeling," *Am. Inst. Aeronaut. Astronaut. J.*, vol. 41, no. 4, pp. 687–696, 2003.

[9] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Comput.*, vol. 9, no. 1, pp. 3–12, Jan. 2005.

[10] A. Ratle, "Kriging as a surrogate fitness landscape in evolutionary optimization," *Artif. Intell. Eng. Design Manufact.*, vol. 15, no. 1, pp. 37–49, May 2001.

[11] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum, "Combining global and local surrogate models to accelerate evolutionary optimization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 1, pp. 66–76, Jan. 2007.

[12] R. Smith, B. Dike, and S. Stegmann, "Fitness inheritance in genetic algorithms," in *Proc. ACM Symp. Appl. Comput.*, 1995, pp. 345–350.

[13] J. H. Chen, D. E. Goldberg, S. Y. Ho, and K. Sastry, "Fitness inheritance in multiobjective optimization," in *Proc. Genet. Evol. Comput. Conf.*, 2002, pp. 319–326.

[14] Y. Jin, M. Olhofer, and B. Sendhoff, "On evolutionary optimization with approximate fitness functions," in *Proc. Genet. Evol. Comput. Conf.*, 2000, pp. 786–792.

[15] M. Emmerich, A. Giotis, M. Oezdenir, T. Baeck, and K. Giannakoglou, "Metamodel-assisted evolution strategies," in *Proc. Parallel Problem Solving from Nature*, vol. 2439. 2002, pp. 371–380.

[16] H. Ulmer, F. Streichert, and A. Zell, "Evolution strategies assisted by Gaussian processes with improved preselection criterion," in *Proc. IEEE Congr. Evol. Comput.*, 2003, pp. 692–699.

[17] H. S. Kim and S. B. Cho, "An efficient genetic algorithms with less fitness evaluation by clustering," in *Proc. Congr. Evol. Comput.*, 2001, pp. 887–894.

[18] Y. Jin and B. Sendhoff, "Reducing fitness evaluations using clustering techniques and neural networks ensembles," in *Proc. Genet. Evol. Comput. Conf.*, vol. 3102. 2004, pp. 688–699.

[19] J. Branke and C. Schmidt, "Faster convergence by means of fitness estimation," *Soft Comput.*, vol. 9, no. 1, pp. 13–20, Jan. 2005.

[20] L. Gräning, Y. Jin, and B. Sendhoff. "Individual-based management of meta-models for evolutionary optimization with applications to 3-D blade optimization," *Evolutionary Computation in Dynamic and Uncertain Environments*, S. Yang, Y.-S. Ong, and Y. Jin, Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 225–250.

[21] P. K. S. Nain and K. Deb, "Computationally effective search and optimization procedure using coarse-to-fine approximation," in *Proc. Congr. Evol. Comput.*, 2003, pp. 2081–2088.

[22] Y. S. Ong, P. B. Nair, and K. Y. Lum, "Max–min surrogate-assisted evolutionary algorithm for robust aerodynamic design," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 392–404, Aug. 2006.

[23] Y. S. Ong, P. B. Nair, and K. Y. Lum, "Evolutionary algorithm with hermite radial basis function interpolations for computationally expensive adjoint solvers," *Comput. Optim. Applicat.*, vol. 39, no. 1, pp. 91–119, Jan. 2008.

[24] K. C. Giannakoglou, D. I. Papadimitriou, and I. C. Kampolis, "Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels," *Comput. Methods Appl. Mech. Eng.*, vol. 195, pp. 6312–6329, Sep. 2006.

[25] M. D. Schmidt and H. Lipson, "Coevolution of fitness predictors," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 736–749, Dec. 2008.

[26] J. Knowles, "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 50–66, Feb. 2006.

[27] D. Jones, M. Schonlau, and W. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optim.*, vol. 13, no. 4, pp. 455–492, Dec. 1998.

[28] K. Deb and P. K. S. Nain, "An evolutionary multiobjective metamodeling procedure using artificial neural networks," *Evolutionary Computation in Dynamic and Uncertain Environments*, S. Yang, Y. S. Ong, and Y. Jin, Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 297–322.

[29] M. Emmerich, K. Giannakoglou, and B. Naujoks, "Single and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 421–439, Aug. 2006.

[30] D. Chafekar, L. Shi, K. Rasheed, and J. Xuan, "Multiobjective GA optimization using reduced models," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 2, pp. 261–265, May 2005.

[31] I. Voutchkov and A. J. Keane. "Multiobjective optimization using surrogates," in *Proc. 7th Int. Conf. Adaptive Comput. Design Manufact.*, Baarn, The Netherlands: M.C. Escher Company, 2006, pp. 167–175.

[32] Y. S. Ong, P. B. Nair, A. J. Keane, and K. W. Wong, "Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems," *Knowledge Incorporation in Evolutionary Computation*, Y. Jin, Ed. Berlin, Germany: Springer-Verlag, 2004, pp. 307–331.

[33] D. Lim, Y. S. Ong, Y. Jin, and B. Sendhoff, "A study on metamodeling techniques, ensembles, and multisurrogates in evolutionary computation," in *Proc. Genet. Evol. Comput. Conf.*, London, U.K.: ACM, 2007, pp. 1288–1295.

[34] A. Samad and K.-Y. Kim, "Multiple surrogate modeling for axial compressor blade shape optimization," *J. Propulsion Power*, vol. 24, no. 2, pp. 302–310, Mar. 2008.

[35] L. E. Zerpa, N. V. Queipo, S. Pintos, and J.-L. Salager, "An optimization methodology of alkaline-surfactant-polymer flooding processes using field scale numerical simulation and multiple surrogates," *J. Petroleum Sci. Eng.*, vol. 47, no. 3–4, pp. 197–208, Jun. 2005.

[36] D. Marjavaara, S. Lundström, and W. Shyy, "Hydraulic turbine diffuser shape optimization by multiple surrogate model approximations of pareto fronts," *ASME J. Fluids Eng.*, vol. 129, no. 9, pp. 1228–1240, 2007.

[37] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker, "Surrogate-based analysis and optimization," *Progr. Aerospace Sci.*, vol. 37, pp. 59–118, 2001.

[38] E. Sanchez, S. Pintos, and N. V. Queipo, "Toward an optimal ensemble of kernel-based approximations with engineering applications," in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2006, pp. 2152–2158.

[39] A. Samad, K.-D. Lee, K.-Y. Kim, and R. T. Haftka, "Application of multiple-surrogate model to optimization of a dimpled channel," in *Proc. 7th World Congr. Struct. Multidisc. Optim.*, 2007, pp. 2276–2282.

[40] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo, "Ensemble of surrogates," *Struct. Multidisc. Optim.*, vol. 33, no. 3, pp. 199–216, Mar. 2007.

[41] E. Acar and M. Rais-Rohani, "Ensemble of metamodels with optimized weight factors," *Struct. Multidisc. Optim.*, vol. 37, no. 3, pp. 279–294, 2009.

[42] K.-H. Liang, X. Yao, and C. Newton, "Combining landscape approximation and local search in global optimization," in *Proc. 1999 Congr. Evol. Comput.*, Piscataway, NJ, vol. 2. 1999, pp. 1514–1520.

[43] K.-H. Liang, X. Yao, and C. Newton, "Evolutionary search of approximated *n*-dimensional landscape," *Int. J. Knowl.-Based Intell. Eng. Syst.*, vol. 4, no. 3, pp. 172–183, 2000.

[44] Y. S. Ong, Z. Zhou, and D. Lim, "Curse and blessing of uncertainty in evolutionary algorithm using approximation," in *Proc. Congr. Evol. Comput.*, Vancouver, BC, 2006, pp. 2928–2935.

[45] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments: A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.

[46] R. Meuth, M. H. Lim, Y. S. Ong, and D. C. Wunsch, II, "A proposition on memes and meta-memes in computing for higher-order learning," *Memetic Comput.*, vol. 1, no. 2, pp. 85–100, Jun. 2009.

[47] C. K. Goh, E. J. Teoh, and K. C. Tan, "Hybrid multiobjective evolutionary design for artificial neural networks," *IEEE Trans. Neural Netw.*, vol. 19, no. 9, pp. 1531–1548, Sep. 2008.

[48] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 107–125, Feb. 2008.

[49] D. S. Liu, K.C. Tan, C. K. Goh, and W. K. Ho, "A multiobjective memetic algorithm based on particle swarm optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 42–50, Feb. 2007.

[50] Y. S. Ong, M.H. Lim, N. Zhu, and K. W. Wong, "Classification of adaptive memetic algorithms: A comparative study," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 1, pp. 141–152, Feb. 2006.

[51] Y. Liu, X. Yao, and T. Higuchi, "Evolutionary ensembles with negative correlation learning," *IEEE Trans. Evol. Comput.*, vol. 4, no. 4, pp. 380–387, Nov. 2000.

[52] L. S. Kwan, "Diffusion equation and global optimization," PhD dissertation, Dept. Automat. Comput.-Aided Eng., Chin. Univ. Hong Kong, China, 2004.

[53] J. F. Rodriguez, J. E. Renaud, and L. T. Watson, "Convergence of Trust-region augmented lagrangian methods using variable fidelity approximation data," Virginia Polytech. Inst. State Univ. Blacksburg, Tech. Rep. TR-97-14, 1997.

[54] C. T. Lawrence and A. L. Tits, "A computationally efficient feasible sequential quadratic programming algorithm," *Soc. Ind. Appl. Math.*, vol. 11, no. 4, pp. 1092–1118, 2001.

[55] D. J. C. Mackay, "Introduction to Gaussian processes," *Neural Netw. Mach. Learn.*, vol. 168, pp. 133–165, 1998.

[56] F. H. Lesh, "Multidimensional least-square polynomial curve fitting," *Commun. ACM*, vol. 2, no. 9, pp. 29–30, 1959.

[57] C. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1995.

[58] H. Ishibuchi and T. Murata, "Multiobjective genetic local search algorithm," in *Proc. IEEE Int. Conf. Evol. Comput.*, 1996, pp. 119–124.

[59] A. Jaszkiewicz, "Genetic local search for multiple objective combinatorial optimization," Inst. Comput. Sci., Poznan Univ. of Tech., Tech. Rep. RA-014/98, 1998.

[60] J. Knowles and D. Corne, "Memetic algorithms for multiobjective optimization: Issues, methods and prospects," *Recent Advances in Memetic Algorithms*, W. E. Hart, N. Krasnogor, and J. E. Smith, Ed. Berlin, Germany: Springer-Verlag, 2005, pp. 313–352.

[61] N. Alexandrov, J. E. Dennis, R. M. Lewis, and V. Torczon, "A trust-region framework for managing the use of approximation models in optimization," *J. Struct. Optim.*, vol. 15, no. 1, pp. 16–23, 1998.

[62] J. G. Digalakis and K. G. Margaritis, "On benchmarking functions for genetic algorithms," *Int. J. Comput. Math.*, vol. 77, no. 4, pp. 481–506, 2001.

[63] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Nanyang Tech. Univ., Singapore, Tech. Rep. and IIT Kanpur, India, KanGAL Rep. 2005005, May 2005.

[64] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[65] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, "A fast elitist nondominated sorting genetic algorithm for multiobjective optimization: NSGA-II," in *Proc. Parallel Problem Solving from Nature VI Conf.*, vol. 1917. 2000, pp. 849–858.

[66] E. Zitzler, L. Thiele, M. Laumanns, C. M. Foneseca, and V. Grunert da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[67] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multiobjective Problems*. New York: Kluwer Academic, 2002, pp. 243–272.

[68] D. A. Van Veldhuizen and G. B. Lamont, "Evolutionary computation and convergence to a Pareto front," *Late Breaking Papers at the Genetic Programming*, 1998, pp. 221–228.

[69] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: Classifications, analysis, and new innovations," Ph.D. dissertation, Graduate School Eng., Air Force Inst. Technol., Dayton, OH, 1999.

[70] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: Methods and applications," Ph.D. dissertation, Swiss Federal Inst. Technol. (ETH), Zurich, Switzerland, TIK-Schriftenreihe Nr. 30, dissertation ETH No. 13398. Aachen, Germany: Shaker Verlag, 1999.

[71] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms: A comparative case study," in *Proc. Fifth Int. Conf. Parallel Problem Solving from Nature (PPSN-V)*. Berlin, Germany: Springer-Verlag, 1998, pp. 292–301.

[72] K. Deb, "Salient issues of multi-objective evolutionary algorithms," in *Multiobjective Optimization Using Evolutionary Algorithms*, 1st ed. Chichester, U.K.: Wiley, 2001, pp. 315–446.

[73] R. G. Regis and C. A. Shoemaker, "Local function approximation in evolutionary algorithms for the optimization of costly functions," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 490–505, Oct. 2004.

[74] H.-M. Gutmann, "On the semi-norm of radial basis function interpolants," Dept. Applied Math. Theor. Phy., Univ. Cambridge, U.K., Tech. Rep. DAMTP 2000/NA04, 2000.

Dudy Lim received the B.Eng. (Hons.) degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 2004. He is currently pursuing the Ph.D. degree on the topic of evolutionary optimization for computationally expensive problems.

Previously, he worked as a Grid Engineer in the Grid Operation and Training Center, Parallel and Distributed Computing Center, School of Computer Engineering, Nanyang Technological University. Currently, he is with Center for Computational Intelligence, School of Computer Engineering, Nanyang Technological University. His research interests include evolutionary computation, neural networks, and grid computing.

Yaochu Jin (M'98–SM'02) received the B.Sc., M.Sc., and Ph.D. degrees, all in automatic control, from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996, respectively, and the second Ph.D. degree in electrical and computer engineering from Ruhr-Universität Bochum, Bochum, Germany, in 2001.

Currently, he is the Principal Scientist with Honda Research Institute Europe, Offenbach, Germany, and also is the Scientific Coordinator at CoR-Lab Graduate School, Bielefeld University, Bielefeld, Germany. Previously, he was an Associate Professor in Electrical Engineering Department, Zhejiang University, a Visiting Researcher and was with the scientific staff at Institut für Neuroinformatik, Ruhr-Universität Bochum, a Postdoctoral Associate in the Industrial Engineering Department, Rutgers, Piscataway, NJ, and a Scientist at Honda Research Institute Europe. His research interests include computational approaches to understanding evolution, learning and development in biology, and evolutionary developmental and learning approaches to complex systems design.

Dr. Jin is the co-editor of *Evolutionary Computation in Dynamic and Uncertain Environments* (Berlin, Germany: Springer, 2007), and the editor of *Multiobjective Machine Learning* (Berlin, Germany: Springer, 2006) and *Knowledge Incorporation in Evolutionary Computation* (Berlin, Germany: Springer, 2005). He is the author of *Advanced Fuzzy Systems Design and Applications* (Heidelberg, Germany: Springer, 2003) and published about 100 journal and conference papers. Currently, he is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART C: APPLICATIONS AND REVIEWS, and the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE. He is a member of Evolutionary Computation Technical Committee and a Member of Emergent Technologies Technical Committee of the IEEE Computational Intelligence Society. He has been an Invited Keynote Speaker on three international conferences and workshops.

Yew-Soon Ong received the B.Eng. and M.Eng. degrees, both in electrical and electronics engineering, from Nanyang Technology University, Singapore, in 1998 and 1999, respectively, and the Ph.D. degree from University of Southampton, Southampton, U.K., in 2002.

Currently, he is an Associate Professor in the Division of Information Systems, School of Computer Engineering, Nanyang Technological University, Singapore, as well as the Director of the Center for Computational Intelligence (C2I), Nanyang Technological University, Singapore. His research interests include in computational intelligence spanning: memetic algorithms, evolutionary design, and grid computing.

Dr. Ong is the co-technical editor-in-chief of *Memetic Computing* journal, an Associate Editor of IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS - PART B, INTERNATIONAL JOURNAL OF SYSTEM SCIENCE AND SOFT COMPUTING JOURNAL. Also, he is the Chair of the Task Force on Memetic Algorithms in the IEEE Computational Intelligence Society Emergent Technology Technical Committee.

Bernhard Sendhoff (M'99–SM'05) studied physics at the Ruhr-Universität Bochum, Germany, and the University of Sussex, U.K. He received the Doctorate degree in physics from Ruhr-Universität Bochum, in 1998.

From 1998 to 1999, he was a Research Assistant at Institute for Neuroinformatics, Zurich, Switzerland. From 1999 to 2002, he was with Honda Research Institute Europe GmbH, Offenbach, Germany in several positions, lastly as Deputy Division Manager. Currently, he is the Chief Technology Officer at Honda Research Institute Europe GmbH in addition to being the Head of the Evolutionary and Learning Technology Group. Also, he is a Honorary Professor at the School of Computer Science, University of Birmingham, U.K., and a Professor at the Technical University Darmstadt, Darmstadt, Germany. He is the author and co-author of over 120 research papers in journals and refereed conferences. His research interests include topics from systems biology and computational intelligence such as evolutionary system design and structure optimization of adaptive systems.

Dr. Sendhoff is a member of the Association for Computing Machinery, the European Neural Network Society and the Deutsche Physikalische Gesellschaft. He is a member of several advisory boards and scientific committees.