Applied Soft Computing 43 (2016) 1–19

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

# A comparative study of local search within a surrogate-assisted multi-objective memetic algorithm framework for expensive problems

Pramudita Satria Palar[a,*], Takeshi Tsuchiya[a], Geoffrey Thomas Parks[b]

[a] University of Tokyo, Tokyo 113-8656, Japan
[b] University of Cambridge, Cambridge CB2 1PZ, United Kingdom

ARTICLE INFO

*Article history:*
Received 21 August 2015
Received in revised form 6 December 2015
Accepted 29 December 2015
Available online 16 February 2016

*Keywords:*
Memetic algorithm
Multi-objective
Local surrogate
Local search
Expensive problems

ABSTRACT

A comparative study of the impacts of various local search methodologies for the surrogate-assisted multi-objective memetic algorithm (MOMA) is presented in this paper. The base algorithm for the comparative study is the single surrogate-assisted MOMA (SS-MOMA) with the main aim being to solve expensive problems with a limited computational budget. In addition to the standard weighted sum (WS) method used in the original SS-MOMA, we studied the capabilities of other local search methods based on the achievement scalarizing function (ASF), Chebyshev function, and random mutation hill climber (RMHC) in various test problems. Several practical aspects, such as normalization and constraint handling, were also studied and implemented to deal with real-world problems. Results from the test problems showed that, in general, the SS-MOMA with ASF and Chebyshev functions was able to find higher-quality solutions that were more robust than those found with WS or RMHC; although on problems with more complicated Pareto sets SS-MOMA-WS appeared as the best. SS-MOMA-ASF in conjunction with the Chebyshev function was then tested on an airfoil-optimization problem and compared with SS-MOMA-WS and the non-dominated sorting based genetic algorithm-II (NSGA-II). The results from the airfoil problem clearly showed that SS-MOMA with an achievement-type function could find more diverse solutions than SS-MOMA-WS and NSGA-II. This suggested that for real-world applications, higher-quality solutions are more likely to be found when the surrogate-based memetic optimizer is equipped with ASF or a Chebyshev function than with other local search methods.

## 1. Introduction

Since they were first introduced, multi-objective evolutionary algorithm (MOEA) methodologies have been widely applied to many engineering and real-world problem cases. Many types of evolutionary algorithms (EAs) and other metaheuristic optimizers, such as the non-dominated sorting based genetic algorithm-II (NSGA-II) [1], multi-objective particle swarm optimization [2], multi-objective differential evolution [3], and MOEA based on decomposition (MOEA/D) [4], have been developed to tackle multi-objective optimization problems in which the goal is to find solutions that lie on the Pareto frontier. NSGA-II has also undergone subsequent developments to form NSGA-III [5] in order to tackle many-objective problems. Several other algorithms, such as the S-metric selection evolutionary multi-objective algorithm (SMS-EMOA) [6] and HypE [7], use the concept of hypervolume to guide the search to find the Pareto front.

Aside from their advantages, EA and metaheuristic optimizers are mainly characterized by their slow convergence to the Pareto front [8], which limits their potential applicability to expensive real-world problems. Due to EA's requirement of numerous function evaluations to reach the global optimum, combining the multi-objective optimizer with a surrogate-modeling methodology is now becoming popular. A surrogate model is an approximation of the true function and acts as a cheap replacement of the original exact function.

A combination of EA with a surrogate model could improve the search performance, especially if expensive function evaluation is involved. The combination of the two methodologies is mainly done through the framework of global surrogate modeling. The

* Corresponding author. Present address: Bandung Institute of Technology, Indonesia. Tel.: +62 81297058664.
*E-mail addresses:* pramsp@ftmd.itb.ac.id (P.S. Palar), tsuchiya@mail.ecc.u-tokyo.ac.jp (T. Tsuchiya), gtp10@cam.ac.uk (G.T. Parks).

http://dx.doi.org/10.1016/j.asoc.2015.12.039
1568-4946/© 2016 Elsevier B.V. All rights reserved.

framework uses all available samples and then searches the surrogate model to find an approximated Pareto front [8]. Various surrogate models exist; among those common in literature are radial basis functions (RBFs) [9], Kriging [10,11], polynomial regression [12], and support vector regression [13]. The methodology of surrogate models have been applied in cases such as airfoil [14–16], compressor blade [17], turbine blade [18], supersonic transport [19], and wing optimization [20–22]. However, the problem with the global surrogate methodology is that it suffers from the curse of dimensionality if the number of decision variables increases [9]. Another approach to increasing the optimizer's efficiency is to introduce local search [23–25] as a search mechanism, but this is not a panacea for expensive problems, because expensive evaluations typically have to be conducted during the process. The use of a scalarizing function is one popular means of guiding the local search, although several other local search approaches, such as the multi-objective Rosenbrock algorithm [26], the hill climber with sidestep [27], and the adaptive directional local search strategy [28], are also available. Gradient information could assist the local search, but not all problems readily provide this information. To cope with these problems, a local surrogate model implemented inside the local search module of the memetic multi-objective algorithm was developed [29].

The memetic algorithm [30] itself is an EA equipped with a local search module to perform local improvement of individual solutions [31]. EA optimization with local search based on a local surrogate was first introduced by Ong et al. [9] for a single-objective problem. Many local surrogate models are now built to assist the local search process of the memetic algorithm instead of building one global model. The main advantage of this local search-local surrogate framework is the combination of the power of an evolutionary operator and local search-local surrogate to explore and exploit the search space, respectively. The use of a local surrogate also greatly reduces the number of function calls needed during the local search procedure. The multi-objective version performs a local search on the offspring of the current generation and uses a weighted sum (WS) function to guide the local search implemented inside a trust region framework [29]. The methodology takes advantage of the exploration power of the evolutionary operator and the exploitation ability of local search using a local surrogate as the cornerstone. One prominent memetic algorithm that uses a local surrogate is the single surrogate-assisted multi-objective memetic algorithm (SS-MOMA), which also has a generalized surrogate version (GS-MOMA) [29]. The use of multiple surrogates [32] was also introduced to improve the approximation quality of the surrogate model. Several other algorithms that use the local surrogate concept have since appeared in the literature, with examples being the MOMA with an aggregate surrogate model (ASM-MOMA) [33] (which builds the surrogate model based on the distance to the currently non-dominated set) and the genetic diversity memetic algorithm (GDMA) [34]. A study of the effects of various optimizer (not scalarizing functions) on ASM-MOMA performance is given by Pilat and Neruda [35].

The method is open to the further improvement since there are still some challenges to be tackled before it can be used practically in real-world applications. From the algorithmic perspective, the WS function used in the original SS-MOMA has difficulty dealing with a non-convex Pareto front. This difficulty might be particularly problematic if the computational budget is low; thus, an alternative is needed for the local search method. A fundamental aspect of the local surrogate-assisted EA (or local search in general) is the type of scalarizing function employed within the local search module. The choice of a scalarizing function can have a profound effect on the performance of an EA with a local surrogate framework. However, despite its importance, few studies comparing different scalarizing functions within an EA framework exist. One such study is by Derbel et al. [36] who used a $(1+\lambda)$-EA optimizer and tested it on bi-objective NK-landscapes. As reported in this paper, we have sought to improve the existing methodologies, while keeping practical applications and continuous problems in mind, by investigating the effect of various local search methods on optimizer performance. In addition, we have studied the effect of randomized weights and various normalization schemes on the search capability of the optimizer. The issue of constraint handling is also investigated in this paper.

This paper begins with an introduction to the methodology of multi-objective genetic algorithms with surrogate-assisted local search in Section 2. Section 2 also explains the local search methodologies that we studied and compared in this paper. The comparative study on artificial test problems we undertook is explained in Section 3. After this study on artificial test problems to find an improved local search methodology, Section 4 presents the application of the improved method to airfoil multi-objective optimization. Finally, conclusions are drawn and future work discussed in Section 5.

## 2. Local surrogate assisted multi-objective memetic algorithm

SS-MOMA is an algorithm developed by Lim et al. [29] and falls in the class of multi-objective memetic algorithms. SS-MOMA uses NSGA-II as the building block with the local search based on a local surrogate methodology, instead of using a global surrogate model that has a tendency to get trapped in local optima. The local search is applied to improve the efficiency of the optimizer with the local surrogate model helping to provide an estimate of the direction in which improvement can be obtained. The local search works by first building a surrogate model using the individual in action and its nearest neighbors, and then searches this surrogate model using sequential quadratic programming (SQP) guided by a scalarizing function or other method that relies on a principle such as the non-domination principle.

The main loop of SS-MOMA is essentially the same as NSGA-II except in its local search module. The purpose of the local search module is to perform local improvement of solutions that pass this module. In contrast to single-objective optimization, the treatment of local search in the multi-objective case is not straightforward because the multi-objective problem has to be converted into a single-objective one. This conversion could take the form of a scalarizing function, which is a single-objective representation of the original multi-objective problem. The alternatives to a scalarizing function are methods that directly deal with the original multi-objective form and use operators that guide the search using the domination principle.

Since the focus of our study is to improve the methodologies by introducing various local search methods and considering real-world problems, SS-MOMA, rather than the generalized version (GS-MOMA) that uses multiple different types of surrogates, is employed for simplicity. Nonetheless, there is still much room for improvement of SS-MOMA. This is one of the main aims of this paper. Furthermore, several modifications are necessary to deal with practical, computationally expensive applications. In this paper we are not focusing on a comparison of surrogate models but on the study and improvement of the local search performance and practical application of SS-MOMA.

We studied the effects of different types of local search on search-convergence trends. The original SS-MOMA used a WS scalarizing function; here we also use an achievement scalarizing function (ASF), Chebyshev function, and random mutation hill climber (RMHC) and compare these with WS. Practical concerns, such as the normalization of the objective functions during local search and constraint handling, were also considered.

### 2.1. General framework and pseudocode

The main backbone of NSGA-II is the non-dominated sorting operator. This operator sorts both parent and offspring solutions based on the non-domination fronts and diversity of the current solutions. This procedure ensures that only the best solutions from the viewpoints of domination and diversity are passed on to the next generation. In the memetic algorithm framework, the offspring will usually undergo a local search module before the selection procedure. SS-MOMA applies this memetic algorithm concept and perform selection on the parents, offspring, and after-local-search solutions. For each local search, a surrogate model is built and the surrogate is searched using either a scalarizing function or domination-based local search. The optimal local solution from this surrogate is then evaluated using the exact function and the result is returned to the optimizer for further non-dominated sorting. If the trust region framework is employed, the single-objective optimization continues and is refined until a termination criterion is reached.

The pseudocode and main loop of SS-MOMA are given below:

**Algorithm 1.** SS-MOMA main loop.

```
Initialize parent population $P_c$;
Evaluate initial population;
Start the generation counter $o = 1$;
while computational budget is not exhausted do
    Perform evolutionary operator and generate offspring population $P_o$;
    Evaluate the solutions;
    /****Local Search Phase****/;
    for each individual $\mathbf{x_{ind}}$ in $P_o$ do
        Build the sampling plan by choosing $m$ nearest points to $\mathbf{x_{ind}}$ in the database;
        Build surrogate model $M$ for each objective function (Chebyshev, ASF,
        RMHC) or aggregate function (WS) using this sampling plan;
        Search the surrogate model to find the optimum point $\mathbf{x_{opt}}$ of the defined
        scalarizing function;
        Evaluate $\mathbf{x_{opt}}$ with the exact function;
        Enter $\mathbf{x_{opt}}$ into the after-local-search solutions $A_l$;
    end
    Perform non-dominated sorting on $P_c$, $P_o$ and $A_l$ to create the new parent
    population $P_{c+1}$;
    Increment the generation counter $o = o + 1$;
end
```

### 2.2. Surrogate model building phase

The surrogate model had first to be built before the local search could proceed. Here, we used a linear spline RBF and Kriging to approximate the response surface where the sampling points used were the individual to be improved and its neighbors. The upper and lower bounds used to build the response surface were exactly the maximum and minimum values of the sampling points' decision variables, respectively. We always normalized decision variables of the surrogate model to the range [0, 1] because disparately scaled decision variables could negatively affect the quality of the surrogate.

For a design variable vector $\mathbf{x} = \{x_1, x_2, \ldots, x_{n_d}\}^{\mathrm{T}}$, the approximation started by choosing samples $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}^{\mathrm{T}}$ and responses $\mathbf{y} = \{y^{(1)}, y^{(2)}, \ldots, y^{(n)}\}^{\mathrm{T}}$, where $n_d$ and $n$ are the decision variable dimension and sample size, respectively. Using this information about sample locations and responses, a specific surrogate model method can then be used to build the approximation model of the black-box function. The concept of an approximation using RBF and Kriging is explained in the subsection that follows, where that explanation refers to a book written by Forrester et al. [37].

#### 2.2.1. Radial basis function

A RBF is a real-valued function whose value depends on the distance from the origin. The expression for the RBF approximation $\hat{f}$ is:

$$\hat{f}(x) = \mathrm{w}^{\mathrm{T}}\boldsymbol{\psi} = \sum_{i=1}^{n_c} w_i \psi(||x - c^{(i)}||) \quad (1)$$

where $c^{(i)}$ denotes the $i$th basis function center out of $n_c$ and $\boldsymbol{\psi}$ is the $n_c$-vector of the basis function $\psi$ values that depends on the distances between the evaluated points and the centers. There are several types of RBF for the approximation including linear, cubic, thin plate spline, and Gaussian; however, here for simplicity only the linear type of RBF is used:

$$\psi(r) = r \quad (2)$$

The following equation has to be solved to obtain $\mathbf{w}$:

$$\Psi\mathbf{w} = \mathbf{y} \quad (3)$$

where $\Psi_{i,j} = \psi(||x^{(i)} - x^{(j)}||)$, $i, j = 1, \ldots, n$.

#### 2.2.2. Kriging

Kriging approximates a function to be modeled using a combination of the basis functions of:

$$\psi^i = \exp\left(-\sum_{j=1}^{n_d} \theta_j |x_j^{(i)} - x_j|^{p_j}\right) \quad (4)$$

The Kriging basis has a vector $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_{n_d}\}^{\mathrm{T}}$, meaning that the width of the basis function can vary. The exponent of the Kriging basis $\mathbf{p} = \{p_1, p_2, \ldots, p_{n_d}\}^{\mathrm{T}}$ is also tunable and can be varied for each dimension. These parameters are called the hyperparameters and can be tuned to minimize the approximation error. This flexibility in the hyperparameters allows Kriging to approximate very complex functions given an adequate number of samples.

The hyperparameters can be optimized by maximizing the log-likelihood function:

$$\ln(L) \approx -\frac{n}{2}\ln(\hat{\sigma}^2) - \frac{1}{2}\ln|\mathbf{\Psi}| \tag{5}$$

where $\mathbf{\Psi}$ is the correlation of the random variables, given by:

$$\mathrm{cor}[Y(x^{(i)}), Y(x^{(l)})] = \exp\left(-\sum_{j=1}^{n_d}\theta_j|x_j^{(i)} - x_j^{(l)}|^{p_j}\right) \tag{6}$$

The correlation matrix $\mathbf{\Psi}$ can then be calculated as:

$$\mathbf{\Psi} = \begin{pmatrix} \mathrm{cor}[Y(x^1), Y(x^1)] & \cdots & \mathrm{cor}[Y(x^1), Y(x^{(n)})] \\ \vdots & \ddots & \vdots \\ \mathrm{cor}[Y(x^{(n)}), Y(x^1)] & \cdots & \mathrm{cor}[Y(x^{(n)}), Y(x^{(n)})] \end{pmatrix} \tag{7}$$

where the maximum likelihood estimates for $\mu$ and $\sigma^2$ are:

$$\hat{\mu} = \frac{\mathbf{1}^{\mathrm{T}}\mathbf{\Psi}^{-1}\mathbf{y}}{\mathbf{1}^{\mathrm{T}}\mathbf{\Psi}^{-1}\mathbf{1}} \tag{8}$$

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\mu)^{\mathbf{T}}\mathbf{\Psi}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{n} \tag{9}$$

After the values of the hyperparameters are set, the Kriging predictor is:

$$\hat{y}(x) = \hat{\mu} + \psi^{\mathrm{T}}\Psi^{-1}(y - 1\hat{\mu}) \tag{10}$$

The log-likelihood function can be optimized using direct search methods, such as genetic algorithms, and fine-tuned with a local search methodology. A study on the optimization of these hyperparameters can be found in [38].

## 2.3. Local search method

The aim of the local search is to find the optimum of the surrogate model defined by the type of scalarizing function or local search methodology. The local search is performed on the surrogate, so the computational cost is not an issue. The choice of local search method is important because it can affect the convergence of the optimization and behavior of the evolved solutions. This avenue of research has yet to be explored in depth, especially in the context of expensive multi-objective optimization using a genetic algorithm. In this paper, several local search methods are studied with test functions chosen to reflect some possible characteristics of the Pareto front. We also limit our study by focusing on implementation aspects and comparison of various local search methods and not stressing the development of a silver bullet and power algorithm to solve expensive multi-objective optimization problems (recall the no-free-lunch theorem).

The local search methods compared in this paper are explained in the subsections that follow.

### 2.3.1. Weighted sum

The WS approach works by assigning a convex combination of the different objectives of interest. Let $\lambda_i$ be an individual weighting where $\lambda_i > 0$ and $\sum_{i=1}^{M}\lambda_i = 1$ with $M$ being the number of the objectives, then an expression for WS is:

$$\text{minimize:} \quad \sum_{i=1}^{M}\lambda_i f_i(x) \tag{11}$$

where $\lambda_i = (\lambda_1, \lambda_2, \ldots, \lambda_M)$ is a randomly generated weight vector.

### 2.3.2. Achievement scalarizing function

ASF uses a defined reference point $\bar{z}_i$ to improve the current solution. An expression for the ASF approach is:

$$\text{minimize:} \quad \max_{i=1}^{M} \ \lambda_i(f_i(x) - \bar{z}_i) \tag{12}$$

where $\lambda_i = (\lambda_1, \lambda_2, \ldots, \lambda_M)$ is a randomly generated weight vector and $\sum_{i=1}^{M}\lambda_i = 1$.

ASF improves an individual solution $\bar{z}$ by maximizing the ASF in a direction defined by the weights. Note that in the hybrid NSGA-II paper by Sindhya et al. [25] the weights were fixed and depended only on the estimated upper and lower bounds of the objective spaces, whereas in this paper the weights were varied to allow more exploration during the local search phase. An advantage of ASF is that an optimal solution of a given achievement function is a guaranteed Pareto optimum. Furthermore, the ASF approach does not require the problem to be convex [25,39].

### 2.3.3. Chebyshev function

The expression for Chebyshev-based local search is exactly the same as ASF. However, instead of using the individual solution as a reference point $\bar{z}$, Chebyshev-based local search uses the upper bounds for normalization (of the offspring or of all evaluated solutions) as the reference point. With this, the individual to be improved will move as far as possible from the defined upper bounds in a direction dictated by the weights. Because basically this function is also an achievement-type function, all properties of ASF local search hold for Chebyshev local search. One reason why the upper bounds are used as the reference point is to allow more diverse search, since the search will move towards the same point if the lower bounds are used. The concept of Chebyshev-based local search in this study was inspired by the use of the Chebyshev function in MOEA/D [4].

### 2.3.4. Random mutation hill climber

RMHC [40,41] is based on the non-domination principle and works by searching the neighborhood of a current solution $x_s$ by generating a mutant solution $x_{mut}$. The mutant solution is then compared with the current solution and accepted when it dominates it. This means that mutant solutions are continuously generated until one represents an improvement. This improved mutant is then accepted as the next step, and mutation then is performed again until the computational budget is exhausted. In this paper, a polynomial mutation operator [1] is used to generate the mutants.

RMHC is a very simple algorithm and it does not need any normalization of the objective spaces. It might be expensive to apply RMHC if a surrogate model is not involved; however, here, the computational cost of optimizing the surrogate is not the issue, since RMHC performs the search on the local surrogate model. The RMHC algorithm is formally defined in Algorithm 2.

**Algorithm 2.** RMHC loop.

```
Generate initial solution $x_s$; in the local search case, the initial solution is the solution
to be improved;
Evaluate initial solution;
while computational budget is not exhausted do
    Generate a mutant solution $x_{mut}$ by perturbing $x_s$ with polynomial mutation;
    Evaluate the mutant;
    if $x_{mut} \succ x_s$ then
        $x_s = x_{mut}$;
    else
        Do nothing;
    end
end
```

This algorithm is also applied to the constrained problems. To deal with constrained problems, the mutant is accepted under the following conditions:

- If $x_s$ is infeasible, and $x_{mut}$ is feasible.
- If both $x_s$ and $x_{mut}$ are infeasible, but $x_{mut}$ has a lower constraint violation value (constraint domination).
- If both $x_s$ and $x_{mut}$ are feasible, but $x_{mut}$ dominates $x_s$.

More sophisticated non-domination-based methods can be used instead of RMHC, but here RMHC is elaborate enough to search the surrogate given the surrogate model can be evaluated cheaply. The number of iterations for RMHC during each local search was set to 10,000.

Here, for sake of clarity, we refer to SS-MOMAs using ASF, WS, RMHC, and Chebyshev as SS-MOMA-ASF, SS-MOMA-WS, SS-MOMA-RMHC, and SS-MOMA-CHEB, respectively.

## 2.4. Normalization

Objective functions of different magnitudes are commonly encountered in real optimization problems. An example of this is to be found in the aerodynamic domain, where the lift and drag coefficients have disparately scaled values, with the former ideally being much greater than unity and the latter much less. If such a scaling difference is great, a simple WS scalarizing function favors the objective that changes in value most during optimization, resulting in a biased search direction. Thus, in this study, to negate this effect SS-MOMA always normalizes the objective functions when local search is performed.

Several normalization methods exist and different normalization methods can affect the search. Here, normalization is only used when scalarizing-function-based local searches such as ASF and WS are applied; RMHC does not need normalization. We studied two different normalization methods for local search in this paper. The first was based on the upper and lower bounds of the generated offspring solutions, while the second used the estimated upper and lower bounds of the objective space. We expected that, due to the effects of normalization, similar behavior would be observed for other types of scalarizing function. Therefore, we studied normalization effects using only ASF as the local search method.

The two normalization methods studied in this paper are detailed in the subsections that follow.

### 2.4.1. Normalization with upper and lower bounds of the offspring solutions

This normalization method is imposed using the estimated maximum and minimum objective function values of the intermediate population when the search on the local surrogate is performed by replacing the objective $f_i$ with [42]:

$$\bar{f}_i = \frac{f_i - f^*_{i_{\min}}}{f^*_{i_{\max}} - f^*_{i_{\min}}} \tag{13}$$

where $f^*_{i_{\max}}$ and $f^*_{i_{\min}}$ are the maximum and minimum values of $f_i$ among the offspring, respectively.

### 2.4.2. Normalization with the estimated upper and lower bounds of the objective spaces

The mathematical expression of this type of normalization is similar to Eq. (13) except that now $f^*_{i_{\max}}$ and $f^*_{i_{\min}}$ are replaced by the upper and lower bounds among all evaluated solutions denoted by $f^\dagger_{i_{\max}}$ and $f^\dagger_{i_{\min}}$, respectively. The expression for this type of normalization is as follows:

$$\bar{f}_i = \frac{f_i - f^\dagger_{i_{\min}}}{f^\dagger_{i_{\max}} - f^\dagger_{i_{\min}}} \tag{14}$$

## 2.5. Constraint handling

A constraint-handling method is necessary if constraints are present. Engineering problems typically involve constraints, so we equipped SS-MOMA with a constraint-handling method to deal with this kind of problem. A simple adaptive constraint-handling method that is just the sum of the violated constraints is used by SS-MOMA in this paper. The values of the constraints can have different magnitudes and this is why normalization is needed. Assuming that the maximum extent of constraint violation is not known in advance (the minimum is zero), the maximum violation value is updated on the fly. The following is the equation for the adaptive constraint violation technique for $k$ constraints:

$$c_{tot} = \sum_{i=1}^{k} \left| \frac{c_i}{c_{i_{\max}}} \right| \tag{15}$$

where $c_{tot}$, $c_i$, and $c_{i_{\max}}$ are the total constraint violation, the constraint violation for constraint $i$, and the maximum constraint violation value for constraint $i$ found so far, respectively. With this formulation, the maximum possible value for $c_{tot}$ is $k$, the number of constraints. This simple constraint-handling approach proved to be effective, without the need for a more complex constraint-handling mechanism, and is applicable to high-dimensional constraint spaces.

Constraint handling in the local search module is achieved by building a surrogate model of the constraint function and using it in combination with the constrained SQP algorithm (if a scalarizing function is used). The constraint-violation value itself cannot be used directly in a surrogate model because the zero values (for solutions that do not violate the constraint) will cause difficulties in the surrogate-building process. However, in some real cases the constraint function is analytical and can be used directly without any need to build a surrogate model. An example of this type of constraint is the thickness of an airfoil, which can be evaluated very straightforwardly.

**Table 1**
Details of the ZDT1, ZDT2, and ZDT3 problems.

| Problem | $h(f_1(x), g(x))$ |
|---|---|
| ZDT1 | $1-\sqrt{\frac{f_1(x)}{g(x)}}$ |
| ZDT2 | $1-\left(\frac{f_1(x)}{g(x)}\right)^2$ |
| ZDT3 | $1-\sqrt{\frac{f_1(x)}{g(x)}}\sin(19\pi f_1(x))$ |

## 3. Study on artificial problems

To study the capabilities of SS-MOMA with various local search methods, we applied the algorithm to several artificial unconstrained problems and compared the results with those of NSGA-II and SS-MOMA with WS. The characteristics of the test functions varied from convex/non-convex to continuous/discontinuous Pareto fronts. To simulate real-world problems, the number of decision variables was set to a moderate value and the computational budget was assumed to be limited (less than 2000 function evaluations). We did not perform the test on constrained problems since the capabilities of the local search methods are better tested on unconstrained problems. However, the constraint-handling method was applied to the real-world problem presented in Section 4.

### 3.1. Unconstrained test problems

We compared SS-MOMA with various types of local search and NSGA-II on the ZDT1, ZDT2, ZDT3 [43], CONV1, and CONV2 [44,27], UF1, and UF7 [45] test problems. The problems have convex (ZDT1, CONV1, CONV2, UF1), concave (ZDT2), linear (UF7), or discontinuous (ZDT3) Pareto fronts and served as good benchmarks for optimizer testing. CONV1 has a convex Pareto front but its scale is relatively smaller with respect to the entire objective space. Another characteristic of the CONV1 problem is that its Pareto front does not coincide with the bounds of objective space as in the ZDT and UF problems. The CONV2 problem, while exhibiting the same characteristics as CONV1, has three objectives to be optimized. The UF1 and UF7 problem were chosen due to their characteristic of complicated Pareto sets. For ZDT and CONV problem sets, linear RBF were used as the local surrogate model due to the simplicity of the function landscape. Kriging model was used only for the UF problems because the functions are highly non-linear and test with RBF showed a very low accuracy when approximating the function.

The ZDT1, ZDT2, and ZDT3 problems are of the form:

$$\text{Minimize} = \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)h(f_1(x), g(x)) \end{cases} \tag{16}$$

where

$$g(x) = 1 + \frac{9}{n_d - 1}\sum_{i=2}^{n_d} x_i \tag{17}$$

The formulation of $h(f_1(x), g(x))$ is different for each ZDT problem, as listed in Table 1.

The decision variables for ZDT1, ZDT2, and ZDT3 are:

$$0 \le x_i \le 1$$
$$1 \le i \le n_d$$

where $n_d$ is the number of decision variables. To simulate typical real-world problems, $n_d$ was set to 15 for the ZDT problems.

**Table 2**
SS-MOMA parameter values.

| | |
|---|---|
| Initial population size | 150 |
| Population size | 26 (ZDT and CONV), 50 (UF) |
| Maximum number of generations | 26 (ZDT and CONV), 7 (UF) |
| Crossover operator | Simulated binary crossover |
| Mutation operator | Polynomial mutation |
| Crossover probability $P_{\text{cro}}$ | 0.9 |
| Mutation probability $P_{\text{mut}}$ | $1/n_d$ |
| Number of neighbors | 150 |
| Surrogate type | Linear RBF (ZDT and CONV), Kriging (UF) |

The CONV1 problem was formulated as:

$$\text{Minimize} \begin{cases} f_1(x) = (x_1-1)^4 + \sum_{j=2}^{n_d}(x_j-1)^2 \\ f_2(x) = \sum_{j=1}^{n_d}(x_j+1)^2 \end{cases} \tag{18}$$

The CONV2 problem was formulated as:

$$\begin{aligned} &\text{Minimize} \quad f_i(x) = \sum_{\substack{j=1 \\ j\neq i}}^{n_d}(x_j - a_{i_j})^2 + (x_i - a_i)^4, \quad i = 1, 2, 3 \\ &a_1 = (1, \ldots, 1) \in \mathbb{R} \\ &a_2 = (-1, \ldots, -1) \in \mathbb{R} \\ &a_3 = (1, -1, 1, -1, \ldots) \in \mathbb{R} \end{aligned} \tag{19}$$

The decision variables for both CONV1 and CONV2 were:

$$-5 \le x_i \le 5$$

$n_d$ was set to 16 for CONV1 and 18 for CONV2.

UF1 problem was formulated as:

$$\text{Minimize} \begin{cases} f_1(x) = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2 \\ f_2(x) = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_1} y_j^2 \end{cases} \tag{20}$$

UF7 problem was formulated as:

$$\text{Minimize} \begin{cases} f_1(x) = \sqrt[5]{x_1} + \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2 \\ f_2(x) = 1 - \sqrt[5]{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2} y_j^2 \end{cases} \tag{21}$$

where

$$y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n_d}\right), \quad j = 2, \ldots, n_d \tag{22}$$

For UF1 and UF7 problem, $J_1 = \{j|j$ is odd and $2 \le j \le n_d\}$ and $J_1 = \{j|j$ is even and $2 \le j \le n_d\}$. The search space is $[0, 1] \times [-1, 1]^{n_d - 1}$ for both UF1 and UF7 problem, and $n_d$ is set to 8.

The parameters for the optimizer are listed in Table 2.

The maximum number of function evaluations and $n_d$ for ZDT and CONV problems were set to relatively high value, resembling real cases with relatively high dimensionality and computational budget, but still limited within genetic algorithm context. On the other side, the maximum number of function evaluations and $n_d$ for UF problems were set to lower value than ZDT and CONV problems to resemble problems with moderate dimensionality and more limited computational budget.

**Table 3**
Statistics and *p*-values of the final solutions on ZDT problems obtained by SS-MOMA-ASF with various normalization methods.

| Problem | Mean IGD | | Std. IGD | | Max. IGD | | Min. IGD | | *p*-Value |
|---|---|---|---|---|---|---|---|---|---|
| | off | obj | off | obj | off | obj | off | obj | |
| ZDT1 | 0.0747 | 0.0807 | 0.0738 | 0.0468 | 0.2568 | 0.1611 | 0.0164 | 0.0290 | 0.3075 |
| ZDT2 | 0.2088 | 0.3108 | 0.2505 | 0.3196 | 0.8173 | 0.7885 | 0.0122 | 0.0138 | 0.6232 |
| ZDT3 | 0.4094 | 0.4192 | 0.0849 | 0.1952 | 0.5721 | 0.8031 | 0.2946 | 0.2402 | 0.4727 |

For all tests on artificial problems the results were averaged from ten runs to account for the stochastic nature of the optimizers. Before comparing the performance of the various local searches on the test functions, we studied the effect of randomization of weights and normalization on the ASF-based searches. The purpose of this study was to seek the best practice in the implementation of ASF-based searches to guide the local surrogate-based optimizer. We only used an ASF-type scalarizing function for the normalization study because the Chebyshev-type function is similar, differing only in the reference point used.

## 3.2. Performance metric

The performance metric used was the Inverted Generational Distance (IGD) that measures how close the Pareto fronts to the current non-dominated solutions. The advantage of using IGD as a performance metric is that it provides a measure of both proximity and diversity in a single metric. The Pareto-optimal solutions of the ZDT and UF problems are already known, thus enabling this metric to be used. For the CONV problems, where the Pareto front is not exactly known a priori, a multi-objective optimizer with high number of population and generations were applied to find the non-dominated solutions that serve as the reference points and arranged to be as uniformly distributed as possible. For statistical test, we used Mann–Whitney *U* test to compare two independent sets of sampled data. The output from this test is the *p*-value which is the estimation of the probability of rejecting the null hypothesis of the study question when that hypothesis is true. Low *p*-value (usually defined with a significance level lower than 0.05) means that the difference between the two groups are statistically significant, while high *p*-value indicates otherwise. If $P$ and $P^*$ are the approximation to the Pareto set and the Pareto set, respectively, IGD metric is formulated as follows:

$$\mathrm{IGD}(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \tag{23}$$

where $d(v, P)$ is the minimum Euclidean distance between $v$ and points in $P$ and $|P^*|$ is the number of points in $P^*$. For ZDT and UF problems, the number of reference points for IGD calculations was set to 500. For CONV1 and CONV2 problems, the number of reference points was 200 and 500, respectively. We also plotted the set of solutions with best and worst IGD in all problems using the attainment surface method [46], which is defined as ‘a boundary in the objective space separating those points that are dominated by or equal to at least one data points, from those that no data point dominates or equals’ [47]. The use of attainment surface methodology allows one to visually compare the performance of the optimizer instead of putting all of the solutions in one plot. This is very useful if there are too many solutions to be plotted with many type of the optimizer to be compared.

## 3.3. Effect of normalization

We studied the effect of normalization on the optimization process and compared the two normalization schemes described in Section 2.4:

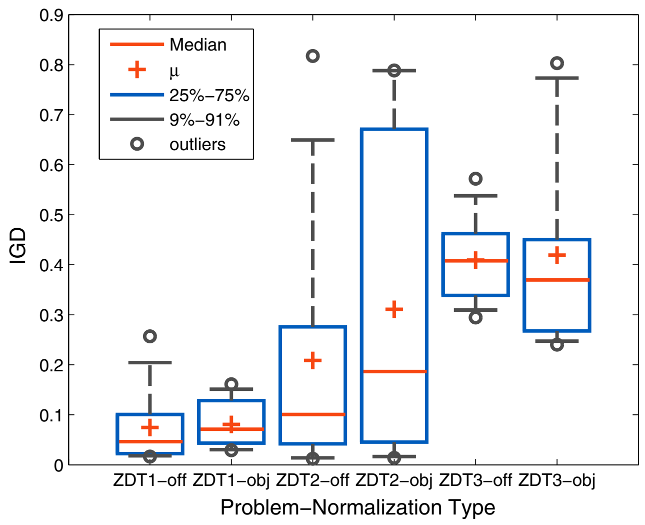


**Fig. 1.** Boxplot of final solutions obtained from SS-MOMA-ASF with various normalization method.

- Normalization using the upper and lower bounds of the offspring solutions (SS-MOMA-ASF-off).
- Normalization using the estimated upper and lower bounds of the objective spaces (SS-MOMA-ASF-obj).

To limit our study, we compared these normalization schemes only on ZDT1, ZDT2, and ZDT3 to see how they affected the search; the results are shown in Fig. 1, and Tables 3 and 4. Random weight generation was used as it was in the original version of SS-MOMA.

For ZDT1, the performance of SS-MOMA-ASF-off is comparable to SS-MOMA-ASF-obj. Statistical hypothesis test confirms that both type of normalization are almost comparable. However, the mean value of the SS-MOMA-ASF-off on ZDT1 problem is slightly lower than the SS-MOMA-ASF-obj although it has higher standard deviation which is reasonable because the changing values used for

**Table 4**
Function evaluations required to achieve average IGD on ZDT problems obtained by SS-MOMA-ASF with various normalization method.

| ZDT1 | IGD | | | |
|---|---|---|---|---|
| Algorithm | 0.5 | 0.1 | 0.05 | 0.025 |
| SS-MOMA-off | 228 | 280 | 384 | 1190 |
| SS-MOMA-obj | 228 | 280 | 514 | 1320 |
| ZDT2 | IGD | | | |
| Algorithm | 1 | 0.5 | 0.35 | 0.25 |
| SS-MOMA-off | 332 | 748 | 1112 | 1372 |
| SS-MOMA-obj | 332 | 826 | 1268 | – |
| ZDT3 | IGD | | | |
| Algorithm | 1 | 0.6 | 0.5 | 0.45 |
| SS-MOMA-off | 280 | 488 | 826 | 1138 |
| SS-MOMA-obj | 254 | 618 | 1008 | 1268 |

**Table 5**
Statistics and $p$-values of the final solutions on ZDT problems obtained by SS-MOMA-ASF with various weight randomization methods.

| Problem | Mean IGD | | Std. IGD | | Max. IGD | | Min. IGD | | $p$-Value |
|---|---|---|---|---|---|---|---|---|---|
| | R | F | R | F | R | F | R | F | |
| ZDT1 | 0.0747 | 0.1499 | 0.0738 | 0.1181 | 0.2568 | 0.4394 | 0.0164 | 0.0252 | 0.0640 |
| ZDT2 | 0.2088 | 0.9924 | 0.2505 | 0.2885 | 0.8173 | 1.4067 | 0.0122 | 0.5483 | 0.0329 |
| ZDT3 | 0.4094 | 0.4184 | 0.0849 | 0.1759 | 0.5721 | 0.6683 | 0.2946 | 0.1596 | 0.7913 |

normalization introduced another aspect of randomness into the search process. The average convergence speed of SS-MOMA-ASF-off is slightly faster than the SS-MOMA-ASF-obj as it is indicated in Table 4. To reach the IGD value of 0.05, SS-MOMA-ASF-off needs about 384 evaluations while it takes 514 evaluations for SS-MOMA-ASF-obj (1.33 times faster).

SS-MOMA-ASF-off had notably better performance on ZDT2. From the result, it can be seen that the SS-MOMA-ASF-obj had a higher probability of creating solutions which are clustered at the boundary of the objective spaces than SS-MOMA-ASF-off. Moreover the standard deviation of IGD on ZDT2 was much larger for SS-MOMA-ASF-obj than SS-MOMA-ASF-off. Although statistical hypothesis shows that both results cannot be easily distinguished, it is safe to conclude here that normalization using the objective values of the offspring yielded better optimizer performance than the estimated bounds of the entire objective space of the ZDT2 problem. The average convergence speed of SS-MOMA-ASF-off is slightly faster than the SS-MOMA-ASF-obj on ZDT2 problem, more notably on reaching the IGD value of 0.35 where SS-MOMA-ASF-off is 1.14 times faster than the SS-MOMA-ASF-obj. SS-MOMA-ASF-obj cannot reach the IGD value of 0.25 within the given computational budget while SS-MOMA-ASF-off needs 1372 function evaluations.

On ZDT3, it is not clear which normalization method was better. The standard deviation of SS-MOMA-ASF-obj is higher but it is able to reach lowest IGD value than the SS-MOMA-ASF-off although one run shows a very worse value. However, statistical hypothesis shows that the results are comparable, which means that it is difficult to judge which normalization type is better. To be noted is that the average convergence speed of SS-MOMA-ASF-off is faster than SS-MOMA-ASF-obj on ZDT3 problems as it is shown in Table 4.

Based on this small experiment, we chose the upper and lower bounds of the offspring as the normalization method due to its better performance on average compared with the other option. Normalization of this type gave more consistent diversity and greater exploration in the optimization process. Moreover, we anticipate that the inconsistency of the SS-MOMA-ASF-obj scheme will be more apparent if the objective range spanned by the Pareto front is much smaller than the range of the entire objective space; a good reason to choose the SS-MOMA-ASF-off scheme.

## 3.4. *Effect of randomized weights*

The introduction of randomized weights, hypothetically, should have been able to improve the performance of the optimizer, especially in terms of diversity. In the local search scheme of Sindhya et al., [25], the ASF was equipped with a fixed weight of one (only normalized), whereas, in this study, we tried to compare local search performance with fixed and randomized weights to see if the hypothesis was true.

As explained in the previous subsection, a normalization based on the upper and lower bounds of the offspring solutions was used. We used ZDT1, ZDT2, and ZDT3 to represent three types of Pareto front. The results are shown in Fig. 2 and Table 5, where it can be seen that, on average, SS-MOMA-ASF-R (randomized weights) performed better than SS-MOMA-ASF-F (fixed weights). Result from statistical hypothesis test that shows low $p$-values on ZDT1 and ZDT2 confirms the performance of the SS-MOMA-ASF-R.

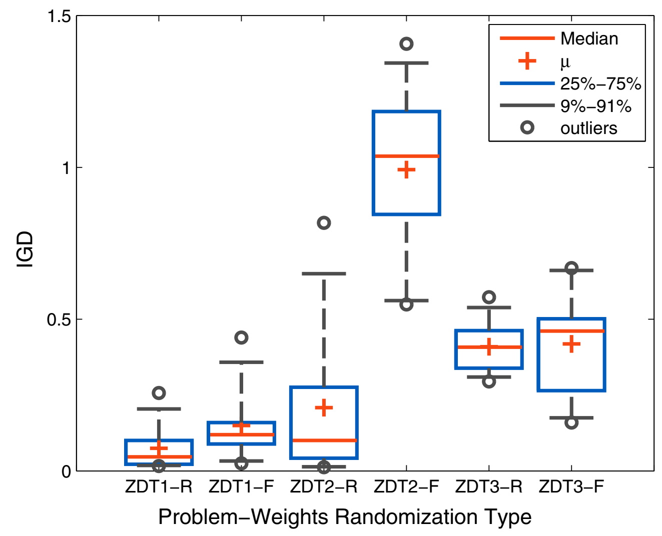


**Fig. 2.** Boxplot of final solutions obtained from SS-MOMA-ASF with various weight randomization method.

SS-MOMA-ASF-R performs slightly and significantly better than SS-MOMA-ASF-F on ZDT1 and ZDT2 problem, respectively. The average convergence speed results (see Table 6) show that SS-MOMA-ASF-R and SS-MOMA-ASF-F reached the IGD value of 0.05 within 384 and 956 functions evaluations on ZDT1 problem, respectively, which means that SS-MOMA-ASF-R is roughly 2.49 times faster. On ZDT2 problem, SS-MOMA-ASF-F is the worst performer while SS-MOMA-ASF-R successfully approached the Pareto front in many runs. An exception is on ZDT3 where although the IGD mean and standard deviation for SS-MOMA-ASF-R were lower than that for SS-MOMA-ASF-F, statistical hypothesis thesis shows that both algorithms are comparable. However, the average convergence speed of SS-MOMA-ASF-R is slightly faster than that for SS-MOMA-ASF-F. Based on the result in Table 6,

**Table 6**
Function evaluations required to achieve average IGD on ZDT problems obtained by SS-MOMA-ASF with various weight randomization method.

| ZDT1 | IGD | | | |
|---|---|---|---|---|
| Algorithm | 0.5 | 0.1 | 0.05 | 0.025 |
| SS-MOMA-R | 228 | 280 | 384 | 1190 |
| SS-MOMA-F | 280 | 384 | 956 | – |
| **ZDT2** | **IGD** | | | |
| Algorithm | 1 | 0.5 | 0.35 | 0.25 |
| SS-MOMA-R | 332 | 748 | 1112 | 1372 |
| SS-MOMA-F | 1528 | – | – | – |
| **ZDT3** | **IGD** | | | |
| Algorithm | 1 | 0.6 | 0.5 | 0.45 |
| SS-MOMA-R | 280 | 488 | 826 | 1138 |
| SS-MOMA-F | 280 | 540 | 956 | 1190 |

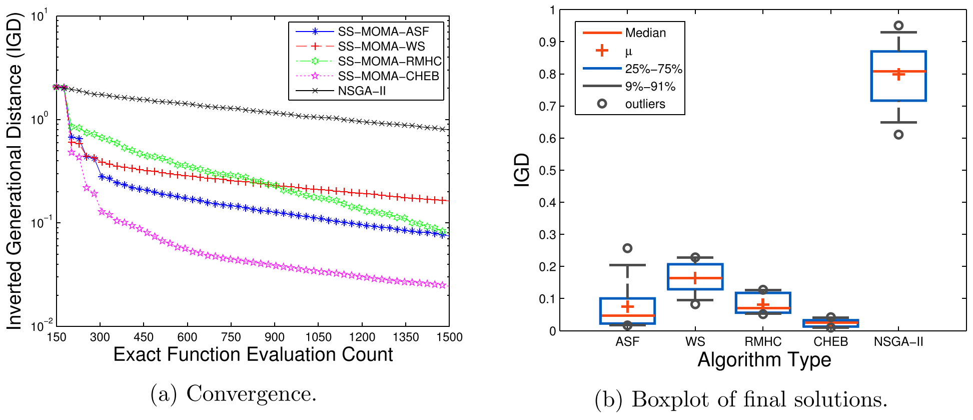


**Fig. 3.** Convergence and boxplots of performance indicators for various algorithms on ZDT1.

SS-MOMA-ASF-R is 1.15 and 1.04 times faster than SS-MOMA-ASF-F to reach IGD value of 0.5 and 0.45, respectively.

On ZDT2, SS-MOMA-ASF-F performance was adversely affected by a high frequency of solutions which were clustered on the edge of the Pareto front. This is due to the fixed weights scheme limiting the ability of the optimizer to perform more exploration of the search space. One way of addressing this problem is the diversity-enhancement method detailed by Sindhya et al. [25]. That method adds complexity to the optimizer and here the use of randomized weights offered a simpler method for enhancing diversity.

It is clear that the introduction of randomized weights in the ASF approach can greatly enhance the diversity of the evolved solutions. The use of randomized weights offered high variation in the search directions of the offspring solutions, thus facilitating exploration of a larger section of the Pareto front.

### 3.5. Results on artificial problems

This subsection presents and analyses the results obtained by applying SS-MOMA with various local search methods and NSGA-II to the selected test problems. Based on our study of normalization and weighting schemes, we used randomized weights and normalization using the upper and lower bounds among the offspring solutions for all local searches based on scalarizing functions for all test problems.

#### 3.5.1. ZDT1

The results for ZDT1 showed that SS-MOMA-CHEB outperforms the other approaches in regard to the IGD convergence to the Pareto front, as indicated by its superiority shown in Fig. 3 and Tables 7 and 8. Its very low standard deviation meant that SS-MOMA-CHEB was very consistent in discovering the ZDT1 Pareto front. The performance of SS-MOMA-ASF was less consistent (higher standard deviation) than SS-MOMA-CHEB but it could find higher quality non-dominated solutions than SS-MOMA-WS. The higher standard deviation can be attributed to the highly randomized nature of SS-MOMA-ASF due to random weight generation and changing reference points (solutions to be improved) for the local search.

On this particular problem, SS-MOMA-RMHC also performed well and had good IGD convergence properties, although it was still inferior to SS-MOMA-CHEB. Moreover, the depiction of the convergence of IGD shows that SS-MOMA-RMHC was slow during early and middle generations but was able to surpass SS-MOMA-WS at the end of the search process. This trend can be attributed to the search in SS-MOMA-RMHC which focuses on similar directions. In contrast, scalarizing-function-type searches have highly exploratory characteristics due to their random weight generation scheme. SS-MOMA-RMHC needed extra time to be able to explore and cover wider sections of the Pareto front. One of the characteristics of ZDT problems (objective space coincides with the Pareto front on $f_1 = 0$) also contributes to this behavior. We anticipate that SS-MOMA-RMHC would perform better on real-world problems that do not exhibit this characteristic. The performance of NSGA-II was, as expected, inferior to SS-MOMA with all types of local search. Result from statistical hypothesis test (see Table 9) confirmed these analysis, where one thing to be noted is that the performance of SS-MOMA-ASF and SS-MOMA-RMHC were comparable; both were superior and inferior to the SS-MOMA-WS and SS-MOMA-CHEB, respectively. SS-MOMA-CHEB also has very fast average convergence speed relative to the other type of local searches. It needs 586 function evaluations to reach IGD value of 0.1 while SS-MOMA-ASF and SS-MOMA-RMHC need 1340 and 1574 function evaluations, respectively, with SS-MOMA-WS cannot reach even IGD value of 0.1.

The non-dominated solutions from the independent runs with the best and worst IGD values for each algorithm are depicted in Fig. 4. Qualitatively speaking, it can be seen that the best solutions obtained by SS-MOMA-ASF and SS-MOMA-CHEB surpass those of the other local searches in quality.

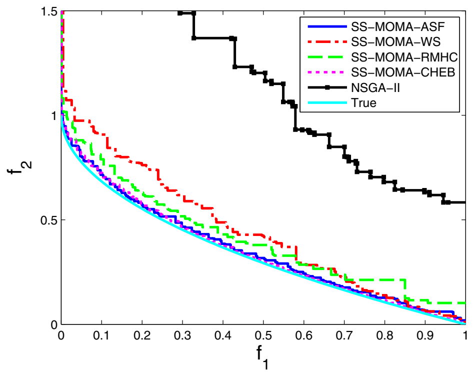
On ZDT1, SS-MOMA with all types of local search could converge to the Pareto front with sufficient accuracy without any problem in the diversity of the solutions, but it is clear that ASF- and Chebyshev-type local search was superior to SS-MOMA-RMHC, SS-MOMA-WS (which was the original local search for SS-MOMA), and NSGA-II.

**Table 7**
Statistics of the final solutions on ZDT1.

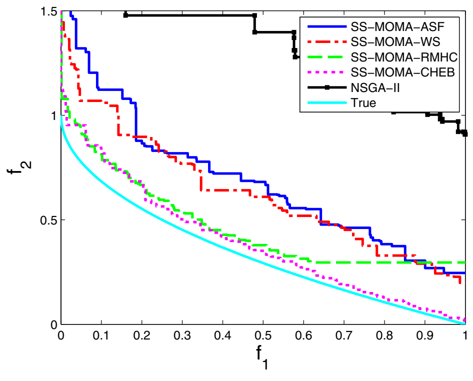
| Algorithm | Mean IGD | Std. IGD | Max. IGD | Min. IGD |
|---|---|---|---|---|
| SS-MOMA-ASF | 0.0747 | 0.0737 | 0.2568 | 0.0164 |
| SS-MOMA-WS | 0.1629 | 0.0488 | 0.2282 | 0.0823 |
| SS-MOMA-RMHC | 0.0812 | 0.0304 | 0.1274 | 0.0514 |
| SS-MOMA-CHEB | 0.0244 | 0.0113 | 0.0419 | 0.0095 |
| NSGA-II | 0.7977 | 0.1020 | 0.9506 | 0.6106 |

**Table 8**
Function evaluations required to achieve average IGD on ZDT1 problem.

| Algorithm | 0.5 | 0.1 | 0.05 | 0.025 |
|---|---|---|---|---|
| SS-MOMA-ASF | 430 | 1340 | – | – |
| SS-MOMA-WS | 430 | – | – | – |
| SS-MOMA-RMHC | 612 | 1574 | – | – |
| SS-MOMA-CHEB | 378 | 586 | 846 | 1652 |
| NSGA-II | – | – | – | – |

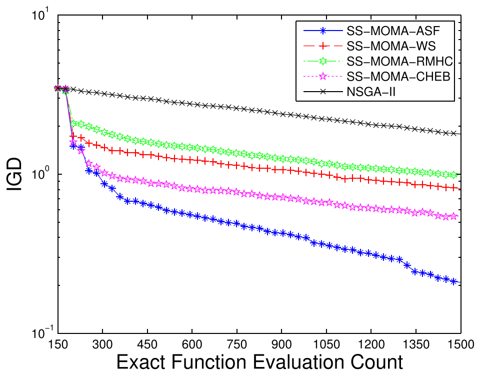


(a) Best solutions.

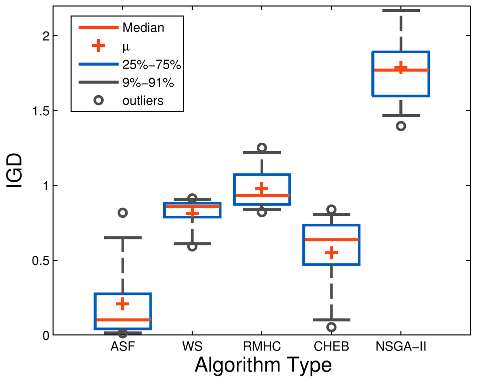


(b) Worst solutions.

**Fig. 4.** Attainment surfaces of the solutions with best and worst IGD values for various algorithms on ZDT1.



(a) Convergence.



(b) Boxplot of final solutions.

**Fig. 5.** Convergence and boxplots of performance indicators for various algorithms on ZDT2.

**Table 9**
Statistical hypothesis test (*p*-value) of IGD of the final solutions on ZDT1.

| Algorithm | ASF | WS | RMHC | CHEB | NSGA-II |
|---|---|---|---|---|---|
| ASF | – | 0.0057 | 0.1858 | 0.0312 | 0.0001 |
| WS | 0.0057 | – | 0.0010 | 0.0001 | 0.0001 |
| RMHC | 0.1858 | 0.0010 | – | 0.0001 | 0.0001 |
| CHEB | 0.0312 | 0.0001 | 0.0001 | – | 0.0001 |
| NSGA-II | 0.0001 | 0.0001 | 0.0001 | 0.0001 | – |

### 3.5.2. *ZDT2*

The results obtained from the experiments on the ZDT2 problem are shown in Fig. 5 and Tables 10 and 11. For ZDT2, SS-MOMA-ASF clearly outperformed the other types of local search and NSGA-II. Both SS-MOMA-WS and SS-MOMA-RMHC frequently found only a very small portion of the Pareto front, where the solutions were concentrated in one area as depicted in Fig. 6. SS-MOMA-RMHC suffered this problem worse than SS-MOMA-WS; the mean value of IGD shows that the diversity of the non-dominated solutions was extremely low. The failure of RMHC on ZDT2 was due to the landscape of the problem, which directed the search into a small portion of the Pareto front. RMHC search is unaffected by the convexity of the problem because progress only depends on whether the next step dominates the current solution or not. Therefore, for each improvement of an individual solution, there was no process like finding an optimal solution of a given subproblem. This is in contrast to local search using a scalarizing function that offers an optimal solution that depends on the type of scalarizing function and the weights for the given subproblem.

To see whether NSGA-II also suffered from the same problem, we tested it with the same population size and number of generations. The results clearly showed that it also failed to discover a large portion of the Pareto front. This means that pure evolutionary operators were the source of the relatively poor performance of all schemes, and the addition of a WS or RMHC local search could not help improve the performance of the optimizer for this particular problem. In contrast, the introduction of ASF- or Chebyshev-type local searches into the memetic algorithm scheme is able to remedy and enhance the diversity of the solutions. The better performance of SS-MOMA-ASF and SS-MOMA-CHEB can be attributed to the properties of achievement-type scalarizing

**Table 10**
Statistics of the final solutions on ZDT2.

| Algorithm | Mean IGD | Std. IGD | Max. IGD | Min. IGD |
|---|---|---|---|---|
| SS-MOMA-ASF | 0.2088 | 0.2504 | 0.8173 | 0.0122 |
| SS-MOMA-WS | 0.8105 | 0.1105 | 0.9131 | 0.5925 |
| SS-MOMA-RMHC | 0.9803 | 0.1422 | 1.2510 | 0.8215 |
| SS-MOMA-CHEB | 0.5486 | 0.2531 | 0.8388 | 0.0535 |
| NSGA-II | 1.7860 | 0.2499 | 2.3093 | 1.3961 |

**Table 11**
Function evaluations required to achieve average IGD on ZDT2 problem.

| Algorithm | 1.5 | 1 | 0.5 | 0.25 |
|---|---|---|---|---|
| SS-MOMA-ASF | 254 | 332 | 748 | 1372 |
| SS-MOMA-WS | 332 | 1086 | – | – |
| SS-MOMA-RMHC | 592 | 1476 | – | – |
| SS-MOMA-CHEB | 254 | 358 | – | – |
| NSGA-II | – | – | – | – |

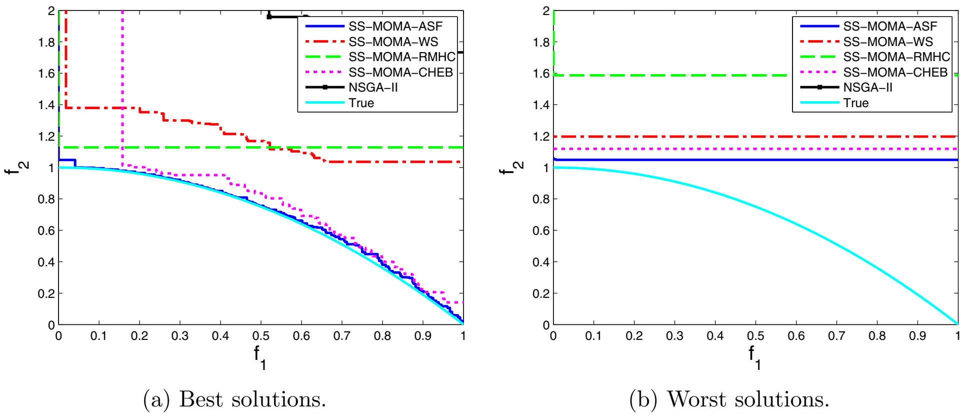


**Fig. 6.** Attainment surfaces of the solutions with best and worst IGD values for various algorithms on ZDT2.

functions that are independent of the convexity of the problem. On this problem, we observed that the memetic algorithm with WS had difficulty solving a non-convex problem. This was in contrast to the result in the original SS-MOMA paper [29] where the scheme with a WS was able to find the entire Pareto front. The failure of SS-MOMA-WS in our case might well be due to the low population size and maximum generation number values specified preventing NSGA-II with a WS scalarizing function from performing well on ZDT2. On this particular problem, independence to the convexity of the problem was important, and this was the reason why SS-MOMA-ASF and SS-MOMA-CHEB clearly performed best. It can be seen in Table 11 that SS-MOMA-ASF and SS-MOMA-CHEB have significantly faster convergence rate than the SS-MOMA-RMHC and SS-MOMA-WS. Statistical hypothesis test (see Table 12) shows that the results from all optimizers are statistically significant, which is indicated by low *p*-values.

Comparing SS-MOMA-ASF and SS-MOMA-CHEB, the former was able to find more diverse solutions than the latter, although the standard deviation was still very high due to the difficulty of the clustered solutions on the edge of the Pareto front. On some runs, SS-MOMA-CHEB was able to overcome this difficulty but many of the runs suffered. The difficulties of SS-MOMA-CHEB were due to its exploration being less dynamic than SS-MOMA-ASF's. Once the reference point was set at a point near the boundary of objective space, it became difficult for SS-MOMA-CHEB to escape. In contrast, SS-MOMA-ASF had a greater likelihood of escaping because of the dynamic nature of its reference point for local improvement.

Clearly, the superior method on ZDT2 was SS-MOMA-ASF. Fig. 6, which shows solutions from the independent runs with the best and worst GD values, confirms this.

### 3.5.3. *ZDT3*

The results show that this problem was more difficult than either ZDT1 or ZDT2 from the viewpoint of approaching the Pareto front. In fact, no local search method could get very close to the true Pareto front of ZDT3. Nevertheless, optimization with a local search was still far better than optimization without it, as indicated by the performance metrics presented in Tables 13 and 14 and Fig. 7. On average, SS-MOMA-ASF and SS-MOMA-CHEB outperformed SS-MOMA-WS in terms of IGD convergence. SS-MOMA-RMHC had the best proximity property to the Pareto front but its diversity was worse than those of other local searches (see Fig. 8). This again shows that domination-based local search is inferior from the diversity viewpoint to scalarizing-function-type local search on complex Pareto fronts, as was also shown by the results for ZDT2. Statistical hypothesis test depicted in Table 15 shows that the results from SS-MOMA-ASF, SS-MOMA-RMHC, and SS-MOMA-CHEB are indeed comparable.

Table 14 shows the superior convergence speed of SS-MOMA-ASF and SS-MOMA-CHEB compares to SS-MOMA-WS and SS-MOMA-RMHC. SS-MOMA-RMHC is slower at the early generation but its performance is roughly the same with SS-MOMA-ASF and SS-MOMA-CHEB at the end of the search.

### 3.5.4. *CONV1*

The CONV1 function has more of the characteristics of a real-world optimization problem because the boundary of its Pareto front does not coincide with the boundary of decision space. The

**Table 13**
Statistics of the final solutions on ZDT3.

| Algorithm | Mean IGD | Std. IGD | Max. IGD | Min. IGD |
|---|---|---|---|---|
| SS-MOMA-ASF | 0.4094 | 0.0848 | 0.5721 | 0.2946 |
| SS-MOMA-WS | 0.4933 | 0.1183 | 0.6706 | 0.2571 |
| SS-MOMA-RMHC | 0.4085 | 0.1173 | 0.5948 | 0.2403 |
| SS-MOMA-CHEB | 0.4086 | 0.1074 | 0.5989 | 0.2895 |
| NSGA-II | 0.8585 | 0.1020 | 1.0003 | 0.6408 |

**Table 14**
Function evaluations required to achieve average IGD on ZDT3 problem.

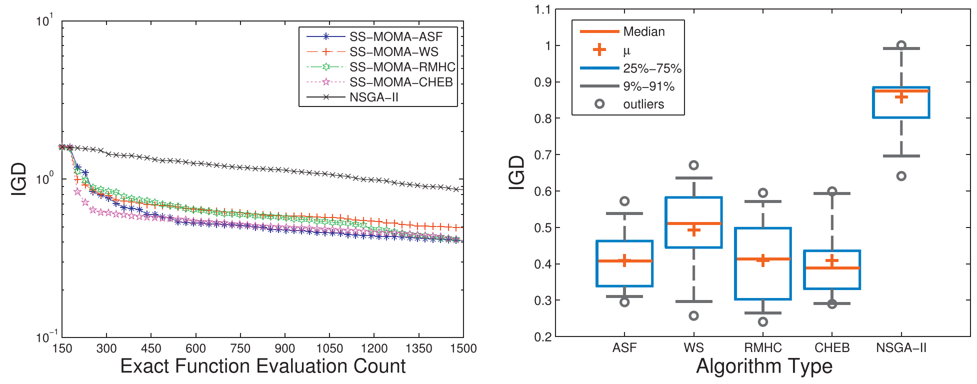
| Algorithm | 1 | 0.75 | 0.5 |
|---|---|---|---|
| SS-MOMA-ASF | 280 | 358 | 826 |
| SS-MOMA-WS | 228 | 358 | 1476 |
| SS-MOMA-RMHC | 254 | 436 | 1216 |
| SS-MOMA-CHEB | 228 | 254 | 904 |
| NSGA-II | – | – | – |

**Table 12**
Statistical hypothesis test (*p*-value) of IGD of the final solutions on ZDT2.

| Algorithm | ASF | WS | RMHC | CHEB | NSGA-II |
|---|---|---|---|---|---|
| ASF | – | 0.0005 | 0.0001 | 0.0172 | 0.0001 |
| WS | 0.0005 | – | 0.0113 | 0.0057 | 0.0001 |
| RMHC | 0.0001 | 0.0113 | – | 0.0002 | 0.0001 |
| CHEB | 0.0172 | 0.0057 | 0.0002 | – | 0.0001 |
| NSGA-II | 0.0001 | 0.0001 | 0.0001 | 0.0001 | – |

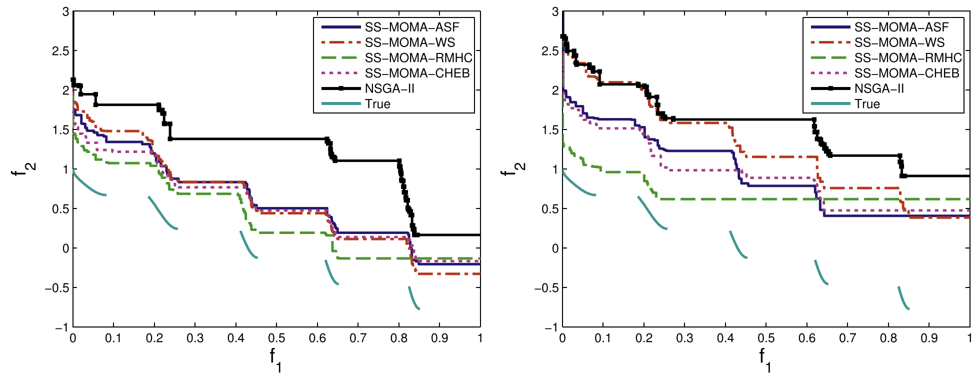
**Table 15**
Statistical hypothesis test (*p*-value) of IGD of the final solutions on ZDT3.

| Algorithm | ASF | WS | RMHC | CHEB | NSGA-II |
|---|---|---|---|---|---|
| ASF | – | 0.0451 | 1 | 0.7337 | 0.0001 |
| WS | 0.0451 | – | 0.1619 | 0.1619 | 0.0002 |
| RMHC | 1 | 0.1619 | – | 0.9097 | 0.0001 |
| CHEB | 0.7337 | 0.1619 | 0.9097 | – | 0.0001 |
| NSGA-II | 0.0001 | 0.0002 | 0.0001 | 0.0001 | – |



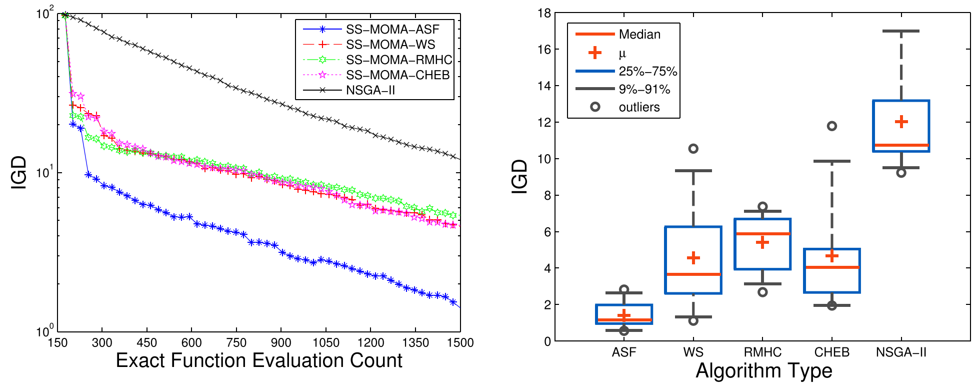
(a) Convergence. (b) Boxplot of final solutions.

**Fig. 7.** Convergence and boxplots of performance indicators for various algorithms on ZDT3.



(a) Best solutions. (b) Worst solutions.

**Fig. 8.** Attainment surfaces of the solutions with best and worst IGD values for various algorithms on ZDT3.



(a) Convergence. (b) Boxplot of final solutions.

**Fig. 9.** Convergence and boxplots of performance indicators for various algorithms on CONV1.

dimensionality of this problem was set to 16. This dimensionality was chosen after solving CONV1 using NSGA-II with large population sizes and high numbers of generations with various dimensions to see which problem had good Pareto front characteristics. To be able to use IGD for performance comparison, the "true" Pareto front was found first by applying NSGA-II with a population size of 100 for 400 generations. This "true" Pareto front was then used as the reference set for calculating the performance metrics.

From the results shown in Fig. 9, Tables 16 and 17, SS-MOMA-ASF was the best performer on CONV1, and was superior to the other local search methods in terms of IGD performance. All the

**Table 16**
Statistics of the final solutions on CONV1.

| Algorithm | Mean IGD | Std. IGD | Max. IGD | Min. IGD |
|---|---|---|---|---|
| SS-MOMA-ASF | 1.4036 | 0.7593 | 2.8212 | 0.5603 |
| SS-MOMA-WS | 4.5736 | 2.9881 | 10.5443 | 1.1103 |
| SS-MOMA-RMHC | 5.4011 | 1.5581 | 7.3664 | 2.6790 |
| SS-MOMA-CHEB | 4.662 | 2.9203 | 11.7830 | 1.9418 |
| NSGA-II | 12.0282 | 2.8041 | 18.4804 | 9.2192 |

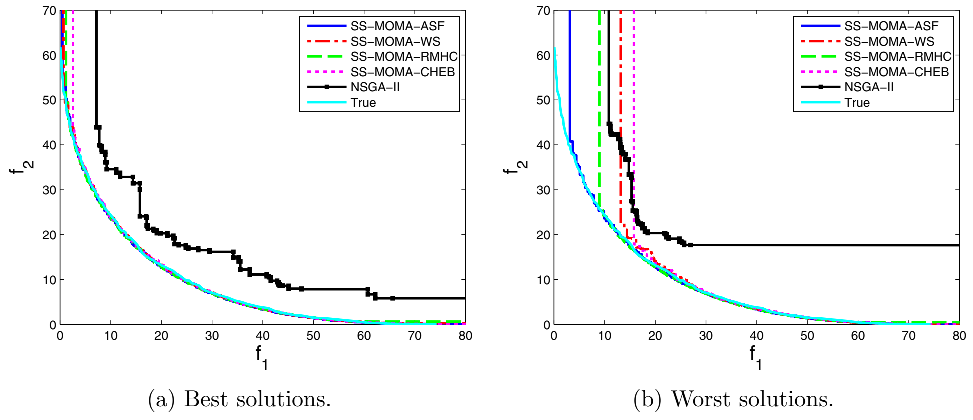


**Fig. 10.** Attainment surfaces of the solutions with best and worst IGD values for various algorithms on CONV1.

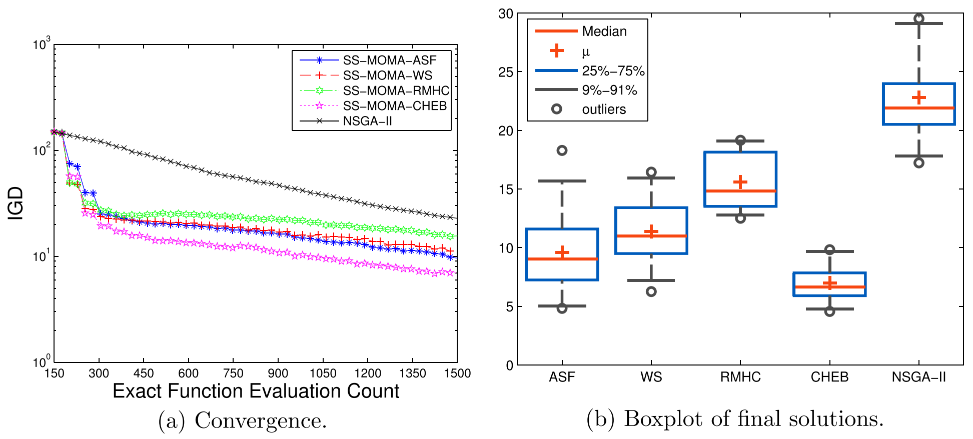


**Fig. 11.** Convergence and boxplots of performance indicators for various algorithms on CONV2.

**Table 17**
Function evaluations required to achieve average IGD on CONV1 problem.

| Algorithm | 20 | 10 | 5 | 2.5 |
|---|---|---|---|---|
| SS-MOMA-ASF | 254 | 280 | 644 | 1164 |
| SS-MOMA-WS | 332 | 774 | 1476 | – |
| SS-MOMA-RMHC | 280 | 826 | – | – |
| SS-MOMA-CHEB | 332 | 826 | 1424 | |
| NSGA-II | 1138 | – | – | – |

local search methods had approximately the same performance of converging to the Pareto front but with notable differences in the diversity performance (see Fig. 10). The hypothesis test (see Table 18) shows that SS-MOMA-WS, SS-MOMA-RMHC, and SS-MOMA-CHEB were all comparable while SS-MOMA-ASF clearly surpasses all of them. NSGA-II was again the worst performer, suggesting that, on this problem, applying a surrogate-based local search is always beneficial, regardless of the local search method used. The experiments on this function showed that, although the function was relatively easy to approximate, the choice of scalarizing function could greatly impact the evolved solutions, thus affecting their quality. In this case, SS-MOMA-ASF was able to explore a wide section of the Pareto front compared to the other local search methods, thanks to the combined effects of randomized weights and suitable normalization methods. As shown in Fig. 10, even the worst solutions obtained with SS-MOMA-ASF feature a diverse set of solutions compared to the other local searches. The average convergence speed of SS-MOMA-ASF is evidently faster than the others, where SS-MOMA-ASF only needs 644 functions evaluations compares to the SS-MOMA-WS and SS-MOMA-CHEB that needs 1476 and 1424 function evaluations, respectively, to reach the mean IGD value of 5.

### *3.5.5. CONV2*

The “true” Pareto front from the CONV2 problem was obtained in the same fashion as the CONV1 problem. The results of this problem are depicted in Fig. 11 and Tables 19 and 20. The best and worst solution sets from each algorithm are not depicted here due to the difficulty of visualizing three-dimensional solutions. Judging from the average performance at the end of the search, SS-MOMA-CHEB is the best performer. The results show

**Table 18**
Statistical hypothesis test ($p$-value) of IGD of the final solutions on CONV1.

| Algorithm | ASF | WS | RMHC | CHEB | NSGA-II |
|---|---|---|---|---|---|
| ASF | – | 0.0036 | 0.0002 | 0.0013 | 0.0001 |
| WS | 0.0036 | – | 0.3447 | 0.9097 | 0.0004 |
| RMHC | 0.0002 | 0.3447 | – | 0.1858 | 0.0001 |
| CHEB | 0.0013 | 0.9097 | 0.1858 | – | 0.0010 |
| NSGA-II | 0.0001 | 0.0004 | 0.0001 | 0.0010 | – |

**Table 19**
Statistics of the final solutions on CONV2.

| Algorithm | Mean IGD | Std. IGD | Max. IGD | Min. IGD |
|---|---|---|---|---|
| SS-MOMA-ASF | 9.5967 | 3.9002 | 18.2958 | 4.8200 |
| SS-MOMA-WS | 11.3803 | 3.1088 | 16.4395 | 6.2523 |
| SS-MOMA-RMHC | 15.6064 | 2.4943 | 19.1593 | 12.5025 |
| SS-MOMA-CHEB | 6.9935 | 1.7009 | 9.8276 | 4.5621 |
| NSGA-II | 22.7957 | 3.8862 | 29.5280 | 17.2294 |

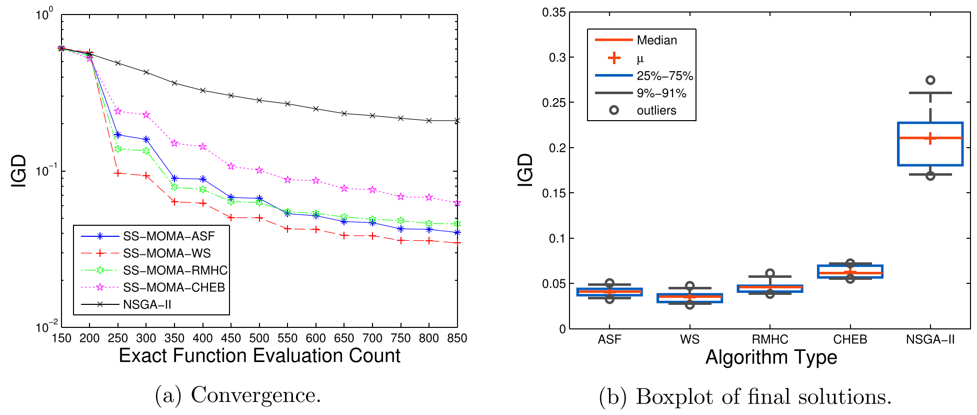


**Fig. 12.** Convergence and boxplots of performance indicators for various algorithms on UF1.

**Table 20**
Function evaluations required to achieve average IGD on CONV2 problem.

| Algorithm | 30 | 20 | 10 | 7 |
|---|---|---|---|---|
| SS-MOMA-ASF | 332 | 566 | 1502 | – |
| SS-MOMA-WS | 280 | 670 | – | – |
| SS-MOMA-RMHC | 332 | 1086 | – | – |
| SS-MOMA-CHEB | 280 | 332 | 1034 | 1450 |
| NSGA-II | 1268 | – | – | – |

**Table 21**
Statistical hypothesis test ($p$-value) of IGD of the final solutions on CONV2.

| Algorithm | ASF | WS | RMHC | CHEB | NSGA-II |
|---|---|---|---|---|---|
| ASF | – | 0.2122 | 0.0017 | 0.0640 | 0.0002 |
| WS | 0.2122 | – | 0.0072 | 0.0036 | 0.0001 |
| RMHC | 0.0017 | 0.0072 | – | 0.0001 | 0.0007 |
| CHEB | 0.0640 | 0.0036 | 0.0001 | – | 0.0001 |
| NSGA-II | 0.0002 | 0.0001 | 0.0007 | 0.0001 | – |

that SS-MOMA-CHEB could more consistently find more diverse solutions with good convergence property, as indicated by the lower mean and standard deviation of IGD. SS-MOMA-CHEB also has faster average convergence speed than the others. Statistical hypothesis test (see Table 21) shows that SS-MOMA-ASF is comparable to SS-MOMA-WS and differs from SS-MOMA-CHEB and SS-MOMA-RMHC. SS-MOMA-CHEB has a low $p$-value if compared with SS-MOMA-ASF, which means that it is reasonable to say that SS-MOMA-CHEB has better performance.

### 3.5.6. UF1

In contrast to the ZDT problems, UF1 has more complex Pareto set and create additional difficulties for the optimizer to discover the Pareto front. The results of this problem are depicted in Fig. 12 and Tables 22 and 23. For this problem, SS-MOMA-WS performs slightly better than the SS-MOMA-ASF while the performance of SS-MOMA-RMHC is comparable to SS-MOMA-ASF. SS-MOMA-CHEB has the worst performance after NSGA-II in this problem as it is indicated by its highest mean IGD value and slower convergence.

**Table 22**
Statistics of the final solutions on UF1.

| Algorithm | Mean IGD | Std. IGD | Max. IGD | Min. IGD |
|---|---|---|---|---|
| SS-MOMA-ASF | 0.0404 | 0.0052 | 0.0502 | 0.0325 |
| SS-MOMA-WS | 0.0346 | 0.0064 | 0.0471 | 0.0263 |
| SS-MOMA-RMHC | 0.0458 | 0.0066 | 0.0610 | 0.0382 |
| SS-MOMA-CHEB | 0.0625 | 0.0069 | 0.0721 | 0.0549 |
| NSGA-II | 0.2099 | 0.0330 | 0.2746 | 0.1687 |

**Table 23**
Function evaluations required to achieve average IGD on UF1 problem.

| Algorithm | 0.1 | 0.05 | 0.04 | 0.035 |
|---|---|---|---|---|
| SS-MOMA-ASF | 350 | 650 | – | – |
| SS-MOMA-WS | 250 | 550 | 650 | 850 |
| SS-MOMA-RMHC | 350 | 700 | – | – |
| SS-MOMA-CHEB | 550 | – | – | – |
| NSGA-II | – | – | – | – |

**Table 24**
Statistical hypothesis test ($p$-value) of IGD of the final solutions on UF1.

| Algorithm | ASF | WS | RMHC | CHEB | NSGA-II |
|---|---|---|---|---|---|
| ASF | – | 0.0625 | 0.0770 | 0.0001 | 0.0001 |
| WS | 0.0625 | – | 0.0027 | 0.0001 | 0.0001 |
| RMHC | 0.0770 | 0.0027 | – | 0.0004 | 0.0001 |
| CHEB | 0.0001 | 0.0001 | 0.0004 | – | 0.0001 |
| NSGA-II | 0.0001 | 0.0001 | 0.0001 | 0.0001 | – |

Statistical hypothesis test depicted in Table 24 shows that the results from all optimizers have low $p$-value when compared to each other. Fig. 13 shows that it is very difficult to cover the entire Pareto front even with the help of the surrogate-based local search. However, it is clear that standard NSGA-II cannot perform adequately without local search. All optimizer show good proximity characteristics so the problem is mainly on the diversity. Analysis of why achievement type scalarizing function perform worse than SS-MOMA-ASF is explained on the results on UF7 problem, because they share similar characteristics.

### 3.5.7. UF7

The results of this problem are depicted in Fig. 14 and Tables 25 and 26. Results on UF7 problems shows a similar conclusion with the results on UF1 problem. SS-MOMA-WS and SS-MOMA-CHEB were the best and worst local-surrogate based optimizer, respectively, while SS-MOMA-ASF and SS-MOMA-RMHC were comparable with high $p$-value (see Table 27). All optimizers

**Table 25**
Statistics of the final solutions on UF7.

| Algorithm | Mean IGD | Std. IGD | Max. IGD | Min. IGD |
|---|---|---|---|---|
| SS-MOMA-ASF | 0.0368 | 0.0067 | 0.0461 | 0.0271 |
| SS-MOMA-WS | 0.0298 | 0.0052 | 0.0381 | 0.0229 |
| SS-MOMA-RMHC | 0.0358 | 0.0076 | 0.0511 | 0.0270 |
| SS-MOMA-CHEB | 0.0656 | 0.0151 | 0.0924 | 0.0412 |
| NSGA-II | 0.2847 | 0.0204 | 0.3217 | 0.2667 |

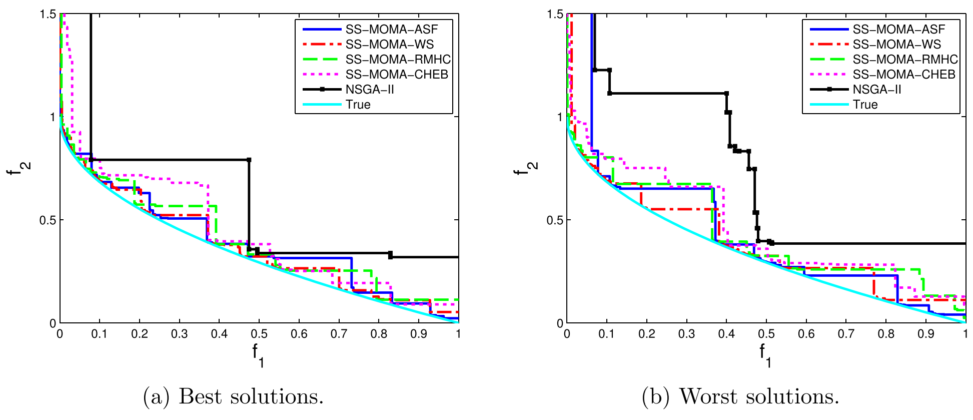


**Fig. 13.** Attainment surfaces of the solutions with best and worst IGD values for various algorithms on UF1.

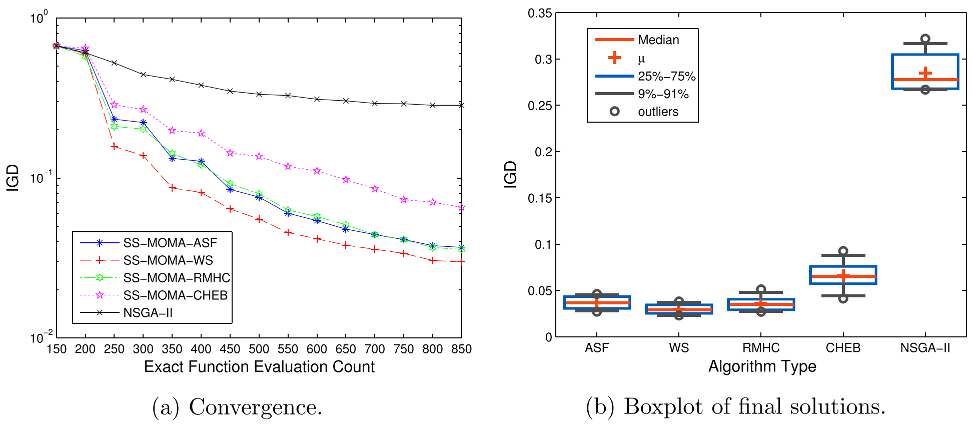


**Fig. 14.** Convergence and boxplots of performance indicators for various algorithms on UF7.

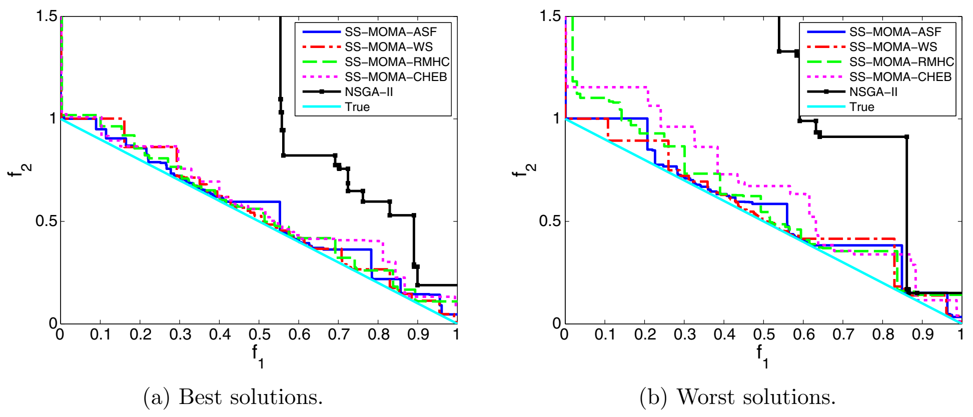


**Fig. 15.** Attainment surfaces of the solutions with best and worst IGD values for various algorithms on UF7.

produced solutions with good proximity, as it can be seen in Fig. 15 and it can be said that no local-search based optimizers encounters problem to discover the Pareto front, however, their performance differs to each other.

Analysis of a snapshot of the non-dominated solutions found by the optimizer reveals that the main problem of achievement type scalarizing function is that they frequently trapped in the flat sections on the edge of the objective spaces, coinciding with the Pareto

**Table 26**
Function evaluations required to achieve average IGD on UF7 problem.

| Algorithm | 0.1 | 0.05 | 0.04 | 0.03 |
|---|---|---|---|---|
| SS-MOMA-ASF | 450 | 650 | 800 | – |
| SS-MOMA-WS | 350 | 550 | 650 | 850 |
| SS-MOMA-RMHC | 450 | 700 | 800 | – |
| SS-MOMA-CHEB | 650 | – | – | – |
| NSGA-II | – | – | – | – |

**Table 27**
Statistical hypothesis test (*p*-value) of IGD of the final solutions on UF7.

| Algorithm | ASF | WS | RMHC | CHEB | NSGA-II |
|---|---|---|---|---|---|
| ASF | – | 0.0314 | 0.5457 | 0.0003 | 0.0001 |
| WS | 0.0314 | – | 0.1134 | 0.0001 | 0.0001 |
| RMHC | 0.5457 | 0.1134 | – | 0.0003 | 0.0001 |
| CHEB | 0.0003 | 0.0001 | 0.0003 | – | 0.0001 |
| NSGA-II | 0.0001 | 0.0001 | 0.0001 | 0.0001 | – |

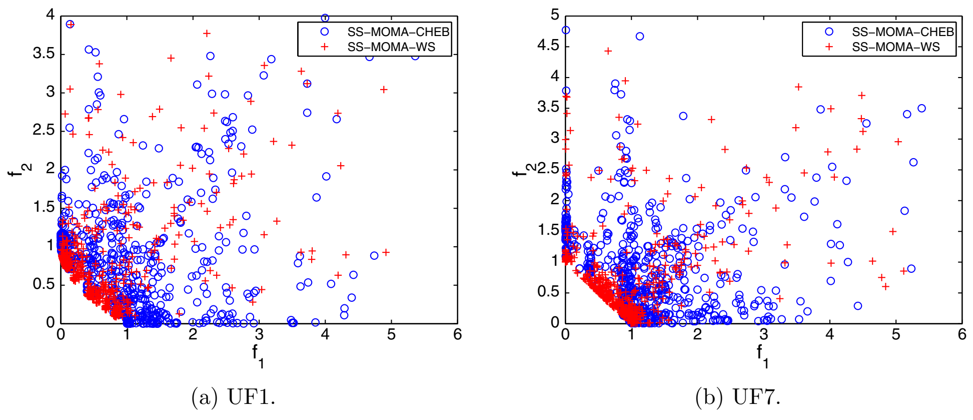


**Fig. 16.** Illustration of the difficulties encountered by SS-MOMA-CHEB in the problem with flat boundaries of the objective space on UF problems.

front. Fig. 16 illustrate this difficulty, where it can be seen that solutions generated by SS-MOMA-CHEB frequently appears on the boundaries of the objective spaces. On the other side, SS-MOMA-WS successfully found the Pareto front without encountering this kind of difficulty. This is because for a given achievement type scalarizing function which optimum point lies in this flat section, all solutions in the flat section are the optimum point and made the search progress becomes difficult if a solution got trapped in this flat section. It is true that ZDT problems has a flat section at $f_1 = 0$, but the fact that UF1 and UF7 problems have flat section in each objective made the achievement type function encountered serious difficulties to quickly converge to the Pareto front.

### *3.6. Discussion of artificial test problem findings*

The results on the artificial test problems showed that, in general, local search based on achievement scalarizing or Chebyshev functions performed better than WS and RMHC. SS-MOMA-ASF was the best performer on the ZDT2 and CONV1 problems but SS-MOMA-CHEB found the highest quality solutions on the ZDT1 and CONV2 problems. On ZDT3 problem, it is not clear which local search was the best, however, SS-MOMA-ASF and SS-MOMA-CHEB had faster average convergence speed compare to the others. SS-MOMA-WS was the best performer on UF1 and UF7 problems, but the lower performance of achievement type scalarizing function was mainly caused by the strong presence of the flat boundaries of the objective spaces. Based on these results, it can be said that search based on ASF- or Chebyshev-type functions was more robust than the original WS if to be applied in SS-MOMA with a limited evaluation budget.

The success of ASF/Chebyshev-based local search was mainly due to independence to the convexity of the problem and direct mapping to the Pareto optimum for a given reference point and weights. Local search methods based on a scalarizing function (ASF, Chebyshev, and WS) were also found to perform better than RMHC, even though the latter does not need normalization and random weights generation. Despite the additional requirements, such as the determination of weights and reference points, of scalarizing functions, they were still more robust than the method based on a non-domination operator. We still do not have enough evidence to say whether SS-MOMA-ASF is better than SS-MOMA-CHEB or vice versa; more test problems are needed. However, for real applications we are more confident about the capabilities of ASF/Chebyshev-based search rather than WS to guide the local search of SS-MOMA. It is important to test this hypothesis on real-world optimization problems. This motivates the study in the following subsection in which SS-MOMA-ASF in conjunction with the Chebyshev function is compared directly with SS-MOMA-WS on a real test problem.

## 4. Application to airfoil optimization

Due to the greater capability of achievement-type scalarizing functions to guide the local search for SS-MOMA, this combination was used for the airfoil case study to examine its utility in real engineering optimization problems. Such a test was also essential for determining the practicability of concepts such as normalization and constraint handling in real-world optimization problems.

On this problem, SS-MOMA-ASF was used as the base algorithm and a switch was made to Chebyshev-based local search if the ASF search got stuck. We chose SS-MOMA-ASF as the base algorithm because it had proved more robust in finding non-convex sections of Pareto fronts than SS-MOMA-CHEB. Since the shape of the Pareto front is unknown beforehand, it is better to use an optimizer with local search which is robust to any possible Pareto front shape. However, on this problem, there were some occasions when search using ASF could not further improve the solution (because the solution obtained by SQP was similar to the individual to be improved), so the local search method was automatically switched by simply changing the reference point if this occurred in a given subproblem.

The computational fluid dynamics (CFD) flow-solver used to solve the Euler equations was SU2 [48]. We did not use the Navier–Stokes equations because the present purpose was to demonstrate the capabilities of SS-MOMA-ASF; hence, a relatively quick evaluation was needed. It is not that expensive to solve for the Euler flow around an airfoil, especially if a high-performance computer is available. Nonetheless, it serves as a good benchmark problem for testing the capabilities of the optimizer. The Euler mesh was generated in an unstructured way using the distmesh algorithm [49] and used the airfoil coordinates to create the mesh; an example of the resulting mesh is shown in Fig. 17.

One also has to be careful when determining the neighbors of the solution to be improved. This is because the decision variables could be disparately scaled, and, if the ranges of the variables are not normalized, the surrogate building scheme could detect the wrong neighbors. Normalization could be done simply by transforming the decision variables to the range [0–1].

Not all generated airfoils returned aerodynamic coefficients because some had erroneous geometries or the CFD failed to converge. Even though we tried to minimize these possibilities, there was no way to guarantee that all generated solutions were error-free. If such cases did occur, the solution was automatically given a very large fitness value, thus eliminating its chance of survival. Moreover, these solutions were not included in the archive for building surrogate models, to ensure that the surrogate-building

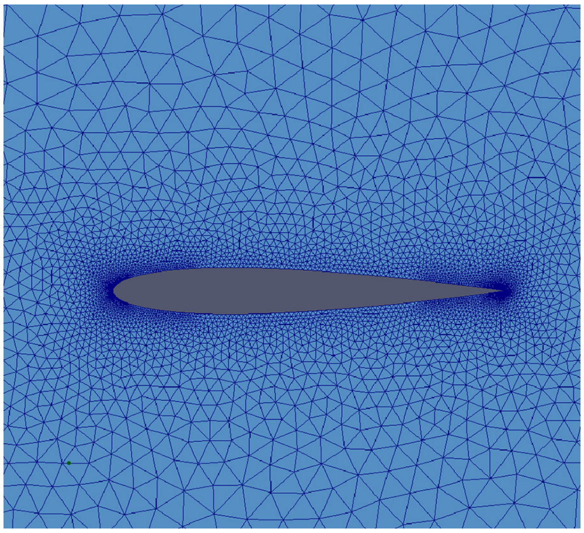


**Fig. 17.** An example of the mesh used with SU2.

process would not be affected by the presence of erroneous solutions.

This case study relates to the cases explained in Refs. [50,51]. The former deals with airfoil optimization while the latter was a redesign of an ONERA M6 wing. The population size was set to 26 in this problem. The first population was initialized with a Halton sequence [52] with 150 samples, with the optimization search was stopped when it reached maximum function evaluations of 700, resembling an optimization case with a limited budget. The crossover and mutation probabilities were set to 0.9 and $1/n_d$, where $n_d$ was the number of decision variables. For all optimizers (SS-MOMA-ASF with Chebyshev, SS-MOMA-WS, NSGA-II) the number of the independent runs was three.

The decision variables were 16-dimensional class-shape-transformation (CST) parameters [53] (8 for each surface) with zero value on the trailing edge ordinate and a thickness set to 0.001. The datum airfoil was RAE2822, with CST parameters found by minimizing the error between the true RAE2822 and the CST-made airfoil shape. The CST parameters were then varied to ±20% as the upper and lower bounds of the decision variables. For this problem, the objective functions were:

$$\begin{aligned}\text{minimize:}\quad & \frac{C_d}{C_l^2}, C_m^2 \\ \text{Subjectto:}\quad & t_{\max} \geq 11.5\%\end{aligned}$$

where $C_d$ is the drag coefficient, $C_l$ is the lift coefficient, $C_m$ is the moment coefficient, and $t_{\max}$ is the maximum thickness of the airfoil.

We wanted to observe optimizer performance when applied to a high-dimensionality problem, where it is very likely that a global surrogate model will fail. For the surrogate model, we used Kriging due to its robustness [54]. The training time for the Kriging hyperparameters can be considered negligible relative to the cost of function evaluation.

To enable the use of IGD as performance indicator, the reference non-dominated solutions were obtained using the solutions from all runs and optimizers. These non-dominated solutions were then used as the reference points for IGD calculation. We did not used statistical hypothesis test for this problem since the number of independent runs is limited. The results obtained by NSGA-II, SS-MOMA-ASF, and SS-MOMA-WS are shown in Table 28 and Fig. 18. In Fig. 18, we directly plotted the obtained solutions instead of the attainment surface since there are only relatively small number of the total non-dominated solutions. These indicate that SS-MOMA-ASF and SS-MOMA-WS clearly outperformed NSGA-II. The solutions obtained by NSGA-II, however, were not so poor in quality compared with those of the other algorithms, despite of the high-dimensionality of the problem. Nonetheless, the use of the local-surrogate and memetic algorithm framework clearly improved the quality of the solutions obtained. The superiority of SS-MOMA-ASF to the others can be clearly observed from the lower mean of IGD from the three runs and its faster convergence rate (see Fig. 19). An examination of all evolved Pareto fronts from the three runs in Fig. 18 clearly shows that the results from SS-MOMA-ASF had a larger diversity than those from SS-MOMA-WS.

**Table 28**
IGD results obtained by various algorithms on the transonic airfoil optimization problem.

| Algorithm | 1st run | 2nd run | 3rd run | Mean |
|---|---|---|---|---|
| SS-MOMA-ASF | 3.770e−04 | 1.162e−04 | 5.008e−04 | 3.318e−04 |
| SS-MOMA-WS | 5.244e−04 | 7.431e−04 | 7.144e−04 | 6.608e−04 |
| NSGA-II | 1.056e−03 | 8.957e−04 | 6.334e−04 | 8.6174e−04 |

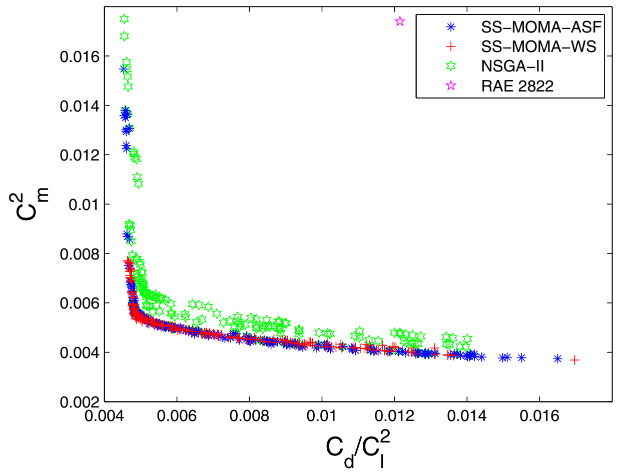


**Fig. 18.** Non-dominated solutions obtained by SS-MOMA-ASF, SS-MOMA-WS and NSGA-II in three independent runs on the transonic airfoil optimization problem.

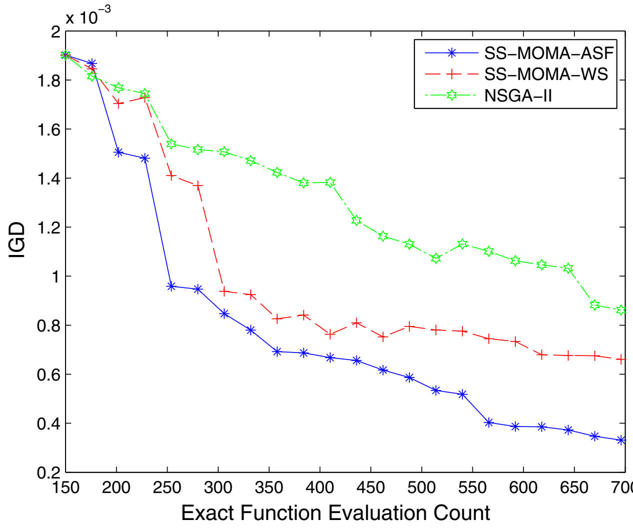


**Fig. 19.** Histories of IGD mean obtained by SS-MOMA-ASF, SS-MOMA-WS and NSGA-II on the transonic airfoil optimization problem.

## 5. Conclusions

This paper has examined improvements in the SS-MOMA methodology for dealing with computationally expensive problems by introducing various local search methodologies beyond the standard WS method. The first local search alternative was the ASF-based local search method, which works by optimizing an achievement-type function using the individual solution to be improved as the reference point. A slightly different version of ASF-based search studied in this paper was Chebyshev-based search, which uses the upper and lower bounds of the offspring solutions as a reference point. In addition to ASF and Chebyshev local search, a non-scalarizing function local search (a random mutation hill climber) was also studied and compared with the others. Improvements were also made by studying the effects of normalization, constraint handling, and diversity enhancement through random weights generation. The study shows that ASF- and Chebyshev-based search performed best overall; they outperformed the local searches based on WS and RMHC on 5 out of 7 test functions. SS-MOMA-CHEB outperformed SS-MOMA-ASF on the ZDT1 and CONV2 problems, while SS-MOMA-ASF found higher quality and more diverse solutions on the ZDT2 and CONV1 problems. On ZDT3 problem, SS-MOMA-ASF and SS-MOMA-CHEB had faster average convergence speed than the others. Surprisingly, SS-MOMA-WS performed as the best in UF1 and UF7, which problems are characterized by complex Pareto sets. Nonetheless, low performance of SS-MOMA-ASF and SS-MOMA-CHEB on UF set was mainly caused by the existence of flat section on the boundaries of both objectives. This suggest that there is no true silver bullet in the choice of the local search methods, although in average, achievement-type scalarizing function performs better than SS-MOMA-RMHC and SS-MOMA-WS.

A real-world aerodynamic optimization problem was tested using aerodynamic efficiency and moment coefficient as the objectives. The results showed that SS-MOMA-ASF used in conjunction with Chebyshev-based local search found higher quality solutions than NSGA-II and SS-MOMA-WS. This demonstrated that the achievement type scalarizing function offers higher likelihood and confidence in finding higher quality solutions within a limited computational budget.

There are a number of avenues to be explored in future work. Other algorithms can be used inside the surrogate-based memetic algorithm framework. For example, instead of using NSGA-II, algorithms such as MOEA/D, SPEA-2, SMS-EMOA, HypE, or NSGA-III could be combined with surrogate-based local search and their performance compared with the existing SS-MOMA framework. The implementation of adaptive weights variation schemes rather than random weights is another potential avenue for exploration. The use of other more advanced scalarizing functions or local search methodologies also needs to be investigated. Furthermore, a selection procedure that automatically chooses the most suitable local search method is also a potential future study. Such studies might all lead to improvements in the search capabilities of the surrogate-based memetic algorithm.

## References

[1] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[2] C.A. Coello Coello, M.S. Lechuga, MOPSO: a proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02, vol. 2, IEEE, 2002, pp. 1051–1056.
[3] T. Robič, B. Filipič, DEMO: differential evolution for multiobjective optimization, in: C.A. Coello Coello, A.H. Aguirre, E. Zitzler (Eds.), Evolutionary Multi-Criterion Optimization, Vol. 3410 of Lecture Notes in Computer Science, Springer, 2005, pp. 520–533.
[4] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[5] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach. Part I. Solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2014) 577–601.
[6] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: multiobjective selection based on dominated hypervolume, Eur. J. Oper. Res. 181 (3) (2007) 1653–1669.
[7] J. Bader, E. Zitzler, HypE: an algorithm for fast hypervolume-based many-objective optimization, Evol. Comput. 19 (1) (2011) 45–76.
[8] I. Voutchkov, A. Keane, Multi-objective optimization using surrogates, in: Y. Tenne, C.-K. Goh (Eds.), Computational Intelligence in Optimization, Vol. 7 of Adaptation, Learning, and Optimization, Springer, 2010, pp. 155–175.
[9] Y.S. Ong, P.B. Nair, A.J. Keane, Evolutionary optimization of computationally expensive problems via surrogate modeling, AIAA J. 41 (4) (2003) 687–696.
[10] N. Cressie, The origins of Kriging, Math. Geol. 22 (3) (1990) 239–252.
[11] T.W. Simpson, T.M. Mauery, J.J. Korte, F. Mistree, Kriging models for global approximation in simulation-based multidisciplinary design optimization, AIAA J. 39 (12) (2001) 2233–2241.
[12] F.H. Lesh, Multi-dimensional least-squares polynomial curve fitting, Commun. ACM 2 (9) (1959) 29–30.
[13] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (3) (2004) 199–222.
[14] S. Jeong, M. Murayama, K. Yamamoto, Efficient optimization design method using Kriging model, J. Aircraft 42 (2) (2005) 413–420.
[15] W. Yamazaki, M.P. Rumpfkeil, D.J. Mavriplis, Design optimization utilizing gradient/Hessian enhanced surrogate model, in: Proceedings of the 28th AIAA Applied Aerodynamics Conference, AIAA 2010-4363, 2010.
[16] J.-C. Jouhaud, P. Sagaut, M. Montagnac, J. Laurenceau, A surrogate-model based multidisciplinary shape optimization method with application to a 2D subsonic airfoil, Comput. Fluids 36 (3) (2007) 520–529.
[17] A. Samad, K.-Y. Kim, T. Goel, R.T. Haftka, W. Shyy, Multiple surrogate modeling for axial compressor blade shape optimization, J. Propul. Power 24 (2) (2008) 301–310.
[18] A.J. Keane, Comparison of several optimization strategies for robust turbine blade design, J. Propul. Power 25 (5) (2009) 1092–1099.
[19] K. Sato, T. Kumano, M. Yonezawa, H. Yamashita, S. Jeong, S. Obayashi, Low-boom and low-drag optimization of the twin engine version of silent supersonic business jet, J. Fluid Sci. Technol. 3 (2008) 576–585.
[20] A. Keane, Wing optimization using design of experiment, response surface, and data fusion methods, J. Aircraft 40 (4) (2003) 741–750.
[21] T. Kumano, S. Jeong, S. Obayashi, Y. Ito, K. Hatanaka, H. Morino, Multidisciplinary design optimization of wing shape for a small jet aircraft using Kriging model, in: Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA 2006-932, 2006.
[22] S. Hosder, L.T. Watson, B. Grossman, W.H. Mason, H. Kim, R.T. Haftka, S.E. Cox, Polynomial response surface approximations for the multidisciplinary design optimization of a high speed civil transport, Optim. Eng. 2 (4) (2001) 431–452.
[23] J.D. Knowles, D.W. Corne, Memetic algorithms for multiobjective optimization: issues, methods and prospects, in: W.E. Hart, J.E. Smith, N. Krasnogor (Eds.), Recent Advances in Memetic Algorithms, Vol. 16 of Studies in Fuzziness and Soft Computing, Springer, 2005, pp. 313–352.
[24] K. Sindhya, A. Sinha, K. Deb, K. Miettinen, Local search based evolutionary multi-objective optimization algorithm for constrained and unconstrained problems, in: IEEE Congress on Evolutionary Computation, 2009 (CEC'09), IEEE, 2009, pp. 2919–2926.
[25] K. Sindhya, K. Miettinen, K. Deb, A hybrid framework for evolutionary multi-objective optimization, IEEE Trans. Evol. Comput. 17 (4) (2013) 495–511.
[26] A. Caponio, F. Neri, Integrating cross-dominance adaptation in multi-objective memetic algorithms, in: C.-K. Goh, Y.-S. Ong, K.C. Tan (Eds.), Multi-Objective Memetic Algorithms, Vol. 171 of Studies in Computational Intelligence, Springer, 2009, pp. 325–351.
[27] A. Lara, G. Sanchez, C.A. Coello Coello, O. Schütze, HCS: a new local search strategy for memetic multi-objective evolutionary algorithms, IEEE Trans. Evol. Comput. 14 (1) (2010) 112–132.
[28] H. Kim, M.-S. Liou, Adaptive directional local search strategy for hybrid evolutionary multiobjective optimization, Appl. Soft Comput. 19 (2014) 290–311.
[29] D. Lim, Y. Jin, Y.-S. Ong, B. Sendhoff, Generalizing surrogate-assisted evolutionary computation, IEEE Trans. Evol. Comput. 14 (3) (2010) 329–355.
[30] P. Moscato, On evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms, Tech. Rep., Caltech Concurrent Computation Program, C3P Report 826, 1989.
[31] X. Chen, Y.-S. Ong, M.-H. Lim, K.C. Tan, A multi-facet survey on memetic computation, IEEE Trans. Evol. Comput. 15 (5) (2011) 591–607.
[32] Z. Zhou, Y.S. Ong, M.H. Lim, B.S. Lee, Memetic algorithm using multi-surrogates for computationally expensive optimization problems, Soft Comput. 11 (10) (2007) 957–971.
[33] M. Pilat, R. Neruda, ASM-MOMA: multiobjective memetic algorithm with aggregate surrogate model, in: IEEE Congress on Evolutionary Computation (CEC), 2011, IEEE, 2011, pp. 1202–1208.
[34] A. Massaro, E. Benini, Multi-objective optimization of helicopter airfoils using surrogate-assisted memetic algorithms, J. Aircraft 49 (2) (2012) 375–383.
[35] M. Pilat, R. Neruda, The effect of different local search algorithms on the performance of multi-objective optimizers, in: IEEE Congress on Evolutionary Computation (CEC), 2014, IEEE, 2014, pp. 2172–2179.
[36] B. Derbel, D. Brockhoff, A. Liefooghe, S. Verel, On the impact of scalarizing functions on evolutionary multiobjective optimization, in: T. Bartz-Beielstein, J. Branke, B. Filipič, J. Smith (Eds.), Parallel Problem Solving from Nature

(PPSN XIII), Vol. 8672 of Lecture Notes in Computer Science, Springer, 2014, pp. 548–558.

[37] A. Forrester, A. Sobester, A. Keane, Engineering Design via Surrogate Modelling: A Practical Guide, John Wiley & Sons, 2008.

[38] D.J. Toal, N. Bressloff, A. Keane, C. Holden, The development of a hybridized particle swarm for Kriging hyperparameter tuning, Eng. Optim. 43 (6) (2011) 675–699.

[39] A.P. Wierzbicki, On the completeness and constructiveness of parametric characterizations to vector optimization problems, OR Spektrum 8 (2) (1986) 73–87.

[40] E.H. Aarts, J.K. Lenstra, Local Search in Combinatorial Optimization, Princeton University Press, 2003.

[41] J.D. Knowles, Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization (Ph.D. thesis), University of Reading, 2002.

[42] A. Messac, A. Ismail-Yahaya, C.A. Mattson, The normalized normal constraint method for generating the Pareto frontier, Struct. Multidisc. Optim. 25 (2) (2003) 86–98.

[43] K. Deb, Multi-objective genetic algorithms: problem difficulties and construction of test problems, Evol. Comput. 7 (3) (1999) 205–230.

[44] M. Dellnitz, O. Schütze, T. Hestermeyer, Covering Pareto sets by multilevel subdivision techniques, J. Optim. Theory Appl. 124 (1) (2005) 113–136.

[45] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition, University of Essex, Colchester, UK and Nanyang Technological University, Singapore, Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Technical Report, 2008, pp. 1–30.

[46] J. Knowles, A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers, in: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications, 2005, ISDA'05, IEEE, 2005, pp. 552–557.

[47] C.M. Fonseca, P.J. Fleming, On the performance assessment and comparison of stochastic multiobjective optimizers, in: Parallel Problem Solving from Nature – PPSN IV, Springer, 1996, pp. 584–593.

[48] Aerospace Design Lab, SU2 The Open-Source CFD Code, 2014 http://su2.stanford.edu/.

[49] P.-O. Persson, G. Strang, A simple mesh generator in MATLAB, SIAM Rev. 46 (2) (2004) 329–345.

[50] T. Ray, H. Tsai, Swarm algorithm for single- and multiobjective airfoil design optimization, AIAA J. 42 (2) (2004) 366–373.

[51] A. Vicini, D. Quagliarella, Inverse and direct airfoil design using a multiobjective genetic algorithm, AIAA J. 35 (9) (1997) 1499–1505.

[52] J.H. Halton, Algorithm 247: radical-inverse quasi-random point sequence, Commun. ACM 7 (12) (1964) 701–702.

[53] B.M. Kulfan, Universal parametric geometry representation method, J. Aircraft 45 (1) (2008) 142–158.

[54] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, A study on metamodeling techniques, ensembles, and multi-surrogates in evolutionary computation, in: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO'07), ACM, 2007, pp. 1288–1295.