Reference-Lines-Steered Memetic Multi-Objective Evolutionary Algorithm with Adaptive Termination Criterion

Riddhiman Saikia · Deepak Sharma

Abstract Multi-objective evolutionary algorithms (MOEAs) have been the choice for generating a set of Pareto-optimal (PO) solutions in one run. However, these algorithms sometimes suffer slow and poor convergence toward the PO front. One of the remedies to improve their convergence is to couple global search of MOEAs with local search. However, such coupling brings other implementation challenges, such as what, when, and how many solutions can be chosen for local search with MOEAs? In this paper, these challenges are addressed by developing a local search module that can choose solutions for local search using a set of reference lines. The heuristic strategies are also developed with the module for determining the frequency of executing local search and for terminating MOEA adaptively using a statistical performance indicator. The proposed algorithm, which is referred to as RM^{2}OEA, is tested on 2-objective ZDT and 3-objective DTLZ test problems. Results demonstrate faster and improved convergence of RM^{2}OEA over a benchmark MOEA from the literature.

1 Introduction

A multi-objective optimization problem (MOOP) can be written as

$$\text{Minimize } F_{\text{MOOP}}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x}))^T, \\ \text{subject to } \mathbf{x} \in S \subset R^n, \tag{1}$$

Department of Mechanical Engineering, Indian Institute of Technology, Guwahati, Assam, India
Tel.: +91-361-2582661
E-mail: {saiki174103012, dsharma}@iitg.ac.in

where $F_{\mathrm{MOOP}}(\mathbf{x})$ is the vector of conflicting objectives $(f_j(\mathbf{x}), \forall j = 1, \ldots, M)$, $\mathbf{x}$ is the vector of decision variables, and S is the feasible search space. Such MOOP can be solved efficiently by using multi-objective evolutionary algorithms (MOEAs) because these algorithms can generate a set of Pareto-optimal (PO) solutions in one run. Therefore, MOEAs have been successfully used for theoretical [8,29] and engineering optimization problems [1,19,23,24].

Despite of showing success in solving MOOP, MOEAs still suffer slow and poor convergence for some problems [12,26]. It generally occurs when many solutions of MOEA become non-dominated. In this case, the environmental selection of MOEA cannot differentiate such solutions, thereby reducing the selection pressure [18]. Moreover, the variation operators such as crossover and mutation operators are also becoming ineffective. One of the remedies to address this challenge is to develop the hybrid or memetic MOEAs [12, 15,25]. These memetic algorithms are developed by coupling global search of MOEA with the local search techniques by converting a MOOP into a single-objective optimization problem (SOOP).

In the literature, many attempts have been made to implement local search with global search of MOEAs. First of all, a MOOP is converted into a SOOP using the weighted-sum method [10,11], the $\epsilon-$constraint method [15,25], or the scalarizing functions [7,26]. After this conversion, a set of solutions for performing local search is chosen. In this case, either an appropriate solution [26] or a set of solutions is selected. The set either consists of the current non-dominated solutions [13,17], or offspring solutions after crossover and mutation [2]. The selection of the appropriate non-dominated solutions is also made heuristic when any solution can be selected with some probability [11,12]. Moreover, the local search is executed at the beginning on the initial population, during the generations, after finishing the fixed number of generations, or combination of two or more such strategies [11, 12]. These memetic algorithms mostly get terminated after a fixed number of generations [10] or function evaluations [25,15]. A little focus has been made to terminate these algorithms adaptively such as in [6,7].

From the literature, it is found that when local search is coupled with MOEAs, a set of implementation challenges emerges that need to be addressed for effective working of memetic MOEAs [11,26]. The first challenge is the criterion for choosing a solution or a set of solutions for local search. Since MOEAs work on a population, the effectiveness of memetic MOEAs depends on this challenge. The second challenge is the number of solutions for local search since limited local searches on few solutions may not be effective or too much local searches on many solutions can become computationally expensive. The third challenge is when to execute local search? Since MOEAs are generation or iteration-based algorithms, it is important to know the right time for executing local search. Otherwise, many local search computations can be wasted without improving the convergence. The fourth

challenge is the termination condition for memetic MOEAs. A fixed number of generations or function evaluations may not be needed since a memetic MOEA might have converged earlier or more number of generations may be needed for better convergence. Addressing all these challenges ensures a balance between local and global searches for better convergence of any memetic algorithm.

This papers focuses on the above challenges by developing a local search module that can be coupled with the existing MOEAs. The following are the contributions of the proposed local search module.

- Selection of solutions for local search using a set of reference lines: In this procedure, a single solution closest to each reference line is chosen for local search. The number of local search solutions is calculated using Das and Denis method [4].

- A heuristic strategy for the frequency of executing local search using the statistical performance indicator called modified inverse generalized distance (IGD^+) indicator.

- A heuristic strategy for an adaptive termination criterion for the proposed MOEA using IGD^+ indicator.

- Comparative analysis of the proposed MOEA over a set of 2- and 3-objective test problems with the existing MOEA.

The remaining paper is organized in five sections. Section 2 presents the relevant literature survey of hybrid or memetic MOEAs and their strategy for balancing local and global searches. Section 3 presents the proposed reference-lines-steered memetic MOEA with implementation details of the local search module. The proposed MOEA is tested and its results are presented in Section 4. The paper is concluded in Section 5 with note on the future work.

2 Related Literature Survey

Multi-objective genetic local search algorithm (MOGLS) [10] is one of the earliest implementations for solving a combinatorial optimization problem. In MOGLS, the weighted-sum method is used to convert a MOOP into a SOOP. All new solutions after crossover and mutation are chosen, and the local search is applied to them after assigning weights. MOGLS is then modified to perform local search on a set of solutions by using tournament selection operator [12]. The solutions for local search are selected using a probability for reducing the number of computations. The algorithm is further extended to S-MOGLS [11] in which the Pareto-ranking for MOEA and the weighted-sum method for local search are used. The selection of solutions

for local search is done with a probability using binary tournament selection operator. Another variant called cellular MOGLS is also developed [17] in which local search is applied to all non-dominated solutions in every generation. In another attempt, local search is coupled with the recombination operators [13] in which all offspring solutions are chosen for local search. A Pareto-local search (PLS) is developed [14] for combinatorial MOOPs, which is coupled with MOEA/D [28]. Three population sets are stored, and PLS is applied to one set so that other sets can be updated. During local search, the solutions are perturbed, and a SOOP is formulated using the weight-sum method.

Apart from the weighted-sum method, the $\epsilon-$constraint method is also used for converting a MOOP into a SOOP [15,25]. The non-dominated solutions in every generation are selected for performing local search using the sequential quadratic programming (SQP) method. The local search solutions are mixed with the current solutions, and the best solutions are chosen. The hybrid MOEAs get terminated using a fixed number of function evaluations. Local search is also implemented by determining all promising non-dominated directions and coupled with MOEA [2]. In another attempt, a local search strategy called hill-climbing sidestep is proposed [16] for MOEAs. The direction for local search is found using the geometry of directional cones. The sidestep is performed when the new solution is closer to the (local) PO solution. The local search is executed on every solution of the offspring population. A hybrid framework for MOEAs is proposed [26] in which a local search solution in every generation is selected from the clusters that are made after projecting the solutions on a hyperplane. A scalarizing function is used for performing local search using the SQP method. In another attempt, local search via perturbation of variables of a solution with respect to its neighborhood is coupled with MOEA [3]. The Gaussian mutation operator is used for perturbing local search solutions. The farthest-candidate solution method is used to include better local search solution in a population. A memetic version of MOEA/D [28] is also developed by coupling a heuristic sequential quadratic approximation method. For each sub-problem, local search using Tchebycheff function is executed in every generation.

Nadir point estimation is another area in which local search is used [6,7]. Since the Nadir point needs to be estimated, only the worst points in each objective from the current population or the extreme solutions of the current non-dominated solutions are found, and local search is executed using the augmented achievement scalarizing function. The SOOP is solved using the SQP method.

The local search has also been used with MOEAs for solving engineering optimization problems. For example, the multi-objective elastic structural topology optimization problem is solved using hybrid or customized MOEA [21–24]. The current non-dominated solutions are clustered, and the best

representative solution from each cluster is chosen for local search in every generation. A problem-specific local search is executed, and the solutions after local search replace its parent solution in the current population. Another example is the optimal soil cutting by bulldozer and its blade [1] in which local search is executed on the obtained non-dominated solutions of MOEA after termination.

From the above literature survey, it can be seen that the studies focused on choosing an appropriate set of local search solutions. The number of local search solutions was decided either heuristically or deterministically. The frequency of executing local search was determined either in every generation or using some functions. The termination of hybrid MOEAs was mostly on a number of generations or functional evaluations. The challenges mentioned in Section 1 have been handled uniquely by these studies in order to make a balance between local and global searches. In the following section, the proposed algorithm is described, and details are given for addressing the challenges of implementing local search with MOEA.

3 Proposed Memetic Multi-Objective Evolutionary Algorithm

The proposed memetic MOEA is developed using a commonly used framework of generational MOEAs, as presented in Algo. 1. It can be seen that a set of input parameters needs to be set at the beginning in Step 1. Thereafter, a random population P_0 of size N is generated in Step 2. The objective functions and constraints are evaluated in Step 3, and the fitness is assigned to each solution. In the standard loop of the generations, a mating pool (P_{tmp}) is created in Step 5 by using a selection operator to select good and above average solutions. The solutions in the mating are then used for creating an offspring population (Q_t) in Step 6 using variation operators, such as crossover and mutation. Both the parent and offspring populations are combined in Step 7. The purpose of combining and evaluating the populations is to select the N best solutions, which is done in Step 8. In this step, the fitness is given to each solution, and the best N solutions are selected for the next generation population (P_{t+1}). At last, the generation counter is increased by one. The loop over generation gets terminated when the counter reaches the maximum number of generations ($T_{\max}$).

3.1 The Local Search Module

The local search module is proposed to address challenges mentioned in Section 1 so that a balance can be made between local and global searches for better convergence. In order to implement local search, a MOOP needs to be converted into a SOOP. The details are given in the following section.

Algorithm 1 Commonly used framework of MOEAs.

1: Set $t = 0$, $T_{\max}$: maximum number of generations, N: population size and other parameters such as probabilities for variation operators, etc.
2: Generate initial population (P_t) of size N.
3: Evaluate and assign fitness to P_t.
4: **while** $t \leq T_{\max}$ **do**
5:     P_{tmp} : Selection of solutions from P_t for creating a mating pool.
6:     Q_t : Creation of offspring population from P_{tmp} using variation operators such as crossover and mutation.
7:     Evaluate Q_t and assign fitness to the combined population ($R_t = P_t \cup Q_t$).
8:     P_{t+1} : Environmental selection for selecting the N best solutions from R_t.
9:     $t + +$;
10: **end while**

3.1.1 Formulation for Local Search

The $\epsilon-$constraint method [20] is used for converting a MOOP to a SOOP, which is given as

$$
\begin{aligned}
\text{Minimize} \quad & f_M(\mathbf{x}), \\
\text{subject to} \quad & g_1(\mathbf{x}) = \epsilon_1 - f_1(\mathbf{x}) \geq 0, \\
& g_2(\mathbf{x}) = \epsilon_2 - f_2(\mathbf{x}) \geq 0, \\
& \vdots \\
& g_{M-1}(\mathbf{x}) = \epsilon_{M-1} - f_{M-1}(\mathbf{x}) \geq 0, \\
& \mathbf{x} \in S \subset R^n.
\end{aligned}
\tag{2}
$$

Here, the parameter, ϵ_j ($\forall j \in \{1, \ldots, M-1\}$), is the upper bound on their respective objective value, f_j. It can be seen that an unconstrained MOOP is converted into a constrained SOOP. In order to solve equation (2), the method of multiplier (MOM) [20] is used that converts the constraint SOOP into an unconstrained optimization problem. Using MOM, the resulting optimization problem is given as

$$
\text{Minimize } P(\mathbf{x}, \sigma^t, \tau^t) = f_M(\mathbf{x}) + R \times \sum_{j=1}^{J} \left\{ \left(\left\langle g_j(\mathbf{x}) + \sigma_j^{(t)} \right\rangle \right)^2 - \left(\sigma_j^{(t)} \right)^2 \right\}
$$
$$
+ R \times \sum_{k=1}^{K} \left\{ \left(h_k(\mathbf{x}) + \tau_k^{(t)} \right)^2 - \left(\tau_k^{(t)} \right)^2 \right\}.
\tag{3}
$$

Here, $P(\mathbf{x}, \sigma^t, \tau^t)$ is the penalty function, R is the penalty parameter, g_j's ($\forall j = 1, \ldots, M-1$) are inequality constraints, h_k's ($\forall k = 1, \ldots, K$) are equality constraints, and t represents iteration counter. The $\sigma_j^{(t)}$ and $\tau_k^{(t)}$ penalty parameters are calculated as

$$
\begin{aligned}
\sigma_j^{(t)} &= \left\langle g_j(\mathbf{x}) + \sigma_j^{(t-1)} \right\rangle, \\
\tau_k^{(t)} &= h_k(\mathbf{x}) + \tau_k^{(t-1)},
\end{aligned}
\tag{4}
$$

where $\langle a \rangle$ is the bracket operator, which is equal to a when $a < 0$. Otherwise, it is equal to zero. The initial values of $\sigma_j^{(0)}$ and $\tau_k^{(0)}$ are kept zero for all constraints.

The unconstrained problem given in equation (3) is solved using the steepest descent method [20] in which a unidirectional search is performed along the unit vector of the steepest descent direction. The gradient of the penalty function is calculated using the central difference method. The MOM method gets terminated after (T_{MOM}) iterations or the difference between the new solution and the current solution is less than or equal to a user-defined parameter (ϵ_{MOM}). In each sequence of MOM, the steepest descent method gets terminated after (T_{SDM}) iterations.

3.1.2 Number and Choice for Local Search Solutions

The number and choice of solutions for local search are determined using Das and Dennis method [4]. The number of solutions for local search (N_{LS}) is found by generating the structured reference points on a unit hyperplane, which is given as

$$N_{LS} = \binom{p + M - 1}{M - 1},\tag{5}$$

where M is the number of objectives, and p is the number of equal divisions on each objective axis.

The same reference points are used for choosing local search solutions through the reference lines. These lines are drawn from the origin and the reference points as shown in Fig. 1.
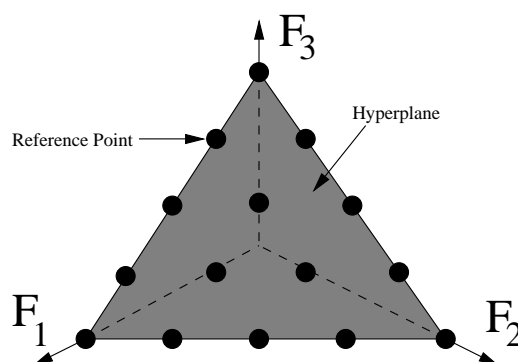
Fig. 1 A set of 15 reference points is generated using $p = 4$ for $M = 3$-objective case. The reference lines are drawn from the origin and the reference points.

In order to select solutions for local search from the current population, all solutions are first normalized and then, clustered. First, each objective value $f_j(\mathbf{x})$ of solution ($\mathbf{x}$) is translated to $\hat{f}_j(\mathbf{x}) = f_j(\mathbf{x}) - z_j^*$, where z_j^* is the $j-$th component of the ideal vector (z^*) found from the current population.

This translation shifts all solutions in the first quadrant of the objective space. Thereafter, the extreme points are found by minimizing the Achievement Scalarizing Function ($ASF(\mathbf{x}, w_j)$) given in equation (6).

$$\text{Minimize } \left\{ ASF(\mathbf{x}, w_j) = \max_{k=1}^{M} \left(\frac{\hat{f}_k(\mathbf{x})}{w_{j,k}} \right) \right\}, \qquad (6)$$

where $w_{j,k}(\forall k \in \{1, \ldots, M\})$ is a search vector for $j-$th objective. It is defined as $w_{j,k} = 1$ for $j = k$, or $w_{j,k} = 10^{-6}$, otherwise. The obtained extreme points are then used to construct a plane to find its intercept on each objective axis, that is, z^e. Using the ideal point and the intercepts, all solutions of the current population are normalized using equation (7).

$$\hat{F}_j(\mathbf{x}) = \frac{\hat{f}_j(\mathbf{x})}{z_j^e(\mathbf{x})}, \qquad (7)$$

where $\hat{F}_j(\mathbf{x})$ is the $j-$th normalized objective value of solution ($\mathbf{x}$).

After normalization, each solution is then clustered with a reference line that is closest to it. It is done by finding the Euclidean distance ($d_1(\mathbf{x}, \mathbf{w})$) between the solution ($\mathbf{x}$) and each reference line ($\mathbf{w}$), which is given as

$$d_1(\mathbf{x}, \mathbf{w}) = ||(\mathbf{x} - \mathbf{w}^T\mathbf{x}\mathbf{w}/||\mathbf{w}||^2)||. \qquad (8)$$

The solution (say $\mathbf{s}$), which has the least d_1 value to a reference line (say $\mathbf{r}$), is included into the cluster of the reference line ($\mathbf{r}$). Similarly, every solution is clustered with one of the reference lines. If any cluster for a reference line is empty, the nearest solution having the least PBI fitness [28] from the clusters of neighboring lines is selected to fill this cluster. The PBI fitness is calculated as

$$F_{PBI}(\mathbf{x}) = d_2 + \theta d_1. \qquad (9)$$

The distance d_2 for a solution ($\mathbf{x}$) along a reference line ($\mathbf{w}$) is calculated as

$$d_2(\mathbf{x}, \mathbf{w}) = \frac{\mathbf{x}^T\mathbf{w}}{||\mathbf{w}||^2}. \qquad (10)$$

Once the clusters are made, a solution having the least PBI distance from each cluster is selected for local search. In this way, the number of chosen set of local search solutions is the same as the number of the reference lines. Since local search is steered by a set of the reference lines, the proposed memetic MOEA is referred to as Reference-Lines-Steered Memetic MOEA (RM^{2}OEA).

3.1.3 Frequency of Executing Local Search

Apart from the number and choice of local search solutions, the frequency of executing local search is an important challenge to address for making a

balance between global and local searches. This challenge can be addressed by using the statistical indicator. There exist many indicators such as R2, hypervolume, generalized distance, inverse generalized distance, etc. Among them, hypervolume indicator is used in various studies, which does not need any reference PO solutions. However, it is computationally expensive as compared to other indicators. In this paper, a modified inverse generalized distance (IGD^+) indicator [9] is used because its performance is found similar to hypervolume indicator but with less computational requirement. It is calculated as

$$IGD^+ = \frac{1}{|P^*|} \sum_{j=1}^{|P^*|} \min d^+(\mathbf{x}_i, \mathbf{a}_j), \tag{11}$$

where P is the set of the obtained non-dominated solutions from MOEA and $\mathbf{x}_i$ is one of its solutions, P^* is the set of the uniformly distributed PO solutions and $\mathbf{a}_j$ is one of the PO solutions, $|P^*|$ is the cardinality of P^*, and $d^+(\mathbf{x}_i, \mathbf{a}_j)$ is the Euclidean distance calculated in the normalized objective space. It is calculated as

$$d^+(\mathbf{x}_i, \mathbf{a}_j) = \sqrt{\sum_{k=1}^{M} (\max\{\mathbf{x}_i^{(k)} - \mathbf{a}_j^{(k)}, 0\})^2}, \tag{12}$$

where $\mathbf{x}_i^{(k)}$ and $\mathbf{a}_j^{(k)}$ represent $k-$th objective value of solutions $\mathbf{x}_i \in P$ and $\mathbf{a}_j \in P^*$. A smaller value of IGD^+ represents better convergence and distribution of P with respect to P^*.

Using IGD^+, the local search is executed after satisfying the condition given in equation (13).

$$IGD_{max}^+ - IGD_{min}^+ - 2 \times IGD_{avg}^+ < 0.05. \tag{13}$$

Here, IGD_{max}^+, IGD_{min}^+, and IGD_{avg}^+ represent maximum, minimum, and average values of IGD^+ from the past 20 generations of RM²OEA. From simulations, it is found that once this condition gets satisfied in any generation, it remains the same for subsequent generations. In order to control many local searches, the next local search is allowed after $(0.1 \times T_{max})$ of generations.

3.2 Selection and Variation Operators

In Step 5 of Algo. 1, a random selection operator is used, which picks two solutions randomly for performing crossover. Once a pair of solutions is considered, they are not selected again and other pair of solutions is picked. In this way, a random selection is used for all solutions of the parent population P_t.

Simulated binary crossover (SBX) and polynomial mutation operators [5] are used as variation operators. SBX is performed with a probability of p_c, which is kept high. Mutation is performed with a probability of p_m, which is kept low.

3.3 Environmental Selection

Environmental selection of $\theta-$DEA [27] is adapted because it is found to be efficient for a large class of MOOPs. Also, it uses reference lines for selecting solutions from the combined population. The $\theta-$dominance principle is used for sorting the solutions. First, all solutions of the combined population are divided into clusters as defined in Section 3.1.2. Thereafter, the solutions from the same cluster are compared using the PBI distance given in equation (9). The solutions of the same cluster are then sorted using the $\theta-$dominance principle. In order to select solutions for the next generation population (P_{t+1}), the best rank solution from each cluster is copied.

3.4 Adaptive Termination of RM^{2}OEA

The proposed RM^{2}OEA gets terminated using IGD$^+$ indicator defined in Section 3.1.3. The IGD$^+$ values are stored from the past few generations and an adaptive termination condition is developed, which is given as

$$\left(\text{IGD}^+_{\max} < \epsilon_{\max}\right) \ \& \ \left(\text{IGD}^+_{\max} - \text{IGD}^+_{\min}\right) < \epsilon. \tag{14}$$

Here, IGD$^+_{\max}$ and IGD$^+_{\min}$ are the maximum and minimum values of IGD$^+$ indicator in the past 20 generations. The terms ϵ and $\epsilon_{\max}$ are used for terminating RM^{2}OEA. The values of these terms are generated using fluctuations in IGD$^+$ values. It means that an increase in IGD$^+$ value for successive generations considers a fluctuation. After experimenting with different sets of values, the following values are used, which are given in equation (15).

$$\begin{aligned}
\epsilon_{\max} &= \begin{cases} 5 \times 10^{-4} & \text{if number of fluctuations} = 0 \\ 1 \times 10^{-3} & 0 < \text{number of fluctuations} \leq 7 \\ 5 \times 10^{-3} & \text{number of fluctuations} > 7 \end{cases} \\
\epsilon &= \begin{cases} 5 \times 10^{-5} & \text{if number of fluctuations} = 0 \\ 1 \times 10^{-4} & 0 < \text{number of fluctuations} \leq 7 \\ 5 \times 10^{-4} & \text{number of fluctuations} > 7 \end{cases}
\end{aligned} \tag{15}$$

3.5 Scope of Local Searches

The local search module can be applied at various steps of RM^{2}OEA presented in Algo. 1. Three variants of RM^{2}OEA are presented in this paper.

For example, when local search module is executed before Step 5 (that is, before selection), it is referred to as RM²OEA-OP. In this implementation, the local search solutions replace their corresponding parent solutions from the population. The second variant is RM²OEA-OF in which local search module is executed before Step 7 (that is, after Q_t is generated). In this implementation, the local search solutions replace their corresponding offspring solutions from the population. The third variant is RM²OEA-ENV in which local search module is executed before Step 9 (that is, after environmental selection). In this implementation, the local search solutions replace their corresponding solutions from the next generation population (P_{t+1}).

4 Results and Discussion

Three variants of RM²OEA is now tested for its performance assessment. The details are as follows.

4.1 Test Problems

The performance of three variants is tested on 2-objective ZDT1, ZDT2, ZDT3, and ZDT6 problems [29]. ZDT1 is a 30−variable problem, which has the convex PO front. ZDT2 is also a 30−variable problem but the PO front is concave. ZDT3 is also a 30−variable problem, which has several disconnected PO fronts. ZDT6 is a 10−variable problem with a concave PO front but the density of solutions along with the PO front changes.

RM²OEA is also tested on 3-objective problems, that is, DTLZ1, DTLZ2, DTLZ3, and DTLZ4 problems [8]. DTLZ1 is a multi-modal MOOP, which has the linear PO front. DTLZ2 is a simple MOOP, which has the concave PO front. DTLZ3 is a multi-modal MOOP, which has the concave PO front. DTLZ4 is a biased MOOP, which has the concave PO front.

4.2 Algorithm for Comparison

It can be observed from the discussion in Section 3.3 that the environmental selection of $\theta-$DEA is used. Therefore, three variants of RM²OEA are tested with $\theta-$DEA. Moreover, $\theta-$DEA has already shown its out-performance over other benchmark MOEAs [27], the comparison is thus made with this algorithm only for simplicity and clarity. It is important to note that the variants of RM²OEA are developed using various local search implementations in the literature. RM²OEA-OF is motivated by MOGLS and its variants [10,11] in which local search is implemented on the offspring population. MOGLS implemented local search on newly created solutions after crossover and mutation. The weighted-sum method was used for converting

the multi-objective problem into the single-objective optimization problem. However, RM^{2}OEA-OF chooses a few solutions as described earlier from the offspring population, and local search is implemented using the $\epsilon-$constraint method.

RM^{2}OEA-ENV is motivated by the studies [15, 25] in which local search is applied to the solutions after the environmental selection. In these studies, all non-dominated solutions after environmental selection were selected for performing local search using the $\epsilon-$constraint method. The solutions after local search were then combined with the current population. However, RM^{2}OEA-ENV chooses few non-dominated solutions using the approach described earlier. The solutions after local search then replace their corresponding solutions in the current population.

It is noted that local search is executed after every $(0.1 \times T_{\max})$ generations for RM^{2}OEA-OP, RM^{2}OEA-OF, and RM^{2}OEA-ENV. The adaptive version of any variant signifies execution of local search as per the details given Section 3.1.3.

4.3 Performance Indicator

The statistical analysis of RM^{2}OEA is performed using IGD$^+$ indicator, which is described in Section 3.1.3. A smaller value of this indicator with respect to the PO front (P^*) signifies better convergence and good diversity among the obtained non-dominated solutions (P).

It is important to note that a set of the PO solutions is needed to calculate IGD$^+$ value. Since the above problems are mathematical MOOPs, their PO fronts are known. For example, the PO front of ZDT1 is defined by $f_2 = 1 - \sqrt{f_1}$ and $f_1 \in [0, 1]$. ZDT2 has the PO front defined by $f_2 = 1 - f_1^2$ and $f_1 \in [0, 1]$. ZDT3 has disconnected PO fronts, which are defined by the ranges in f_1 such as $f_1 = [0, 0.0830015349] \cup [0.1822287280, 0.2577623634] \cup [0.4093136748, 0.4538821041] \cup [0.61839667944, 0.6525117038] \cup [0.8233317983, 0.8518328654]$, and $f_2 = 1 - \sqrt{f_1} - f_1 \sin(10\pi f_1)$. ZDT6 has the same PO front as defined for ZDT2, however the range of f_1 is defined as $f_1 \in [0.2807753191, 1]$. For DTLZ1 problem, the PO front is defined by a linear plane in the first quadrant and having the intercept on each objective axis at 0.5. It is defined by $\sum_{i=1}^{M} f_i = 0.5, \forall f_i \geq 0$. DTLZ2 to DTLZ4 problems have the PO front defined by $\sum_{i=1}^{M} f_i^2 = 1, \forall f_i \geq 0$. It is a spherical surface defined in the first quadrant.

Since a set of the reference points is generated as described in Section 3.1.2, the reference lines drawn from the origin and the reference points are used to find their points of the intersection with the given PO front. These points of intersection become the PO solutions to a given problem. It can be observed that when a set of non-dominated solutions generated by MOEA is evolved closer to these generated PO solutions, a smaller IGD$^+$ value will

Table 1 Different input parameters for running RM^{2}OEA and $\theta-$DEA are presented. Here, N is the population size, $T_{\max}$ is the maximum number of generations, η_c is a parameter for SBX crossover operator, η_m is a parameter for polynomial mutation operator, p_c is the probability of crossover, and p_m is the probability of mutation.

MOOPs	N	$T_{\max}$	η_c	η_m	p_c	p_m
ZDT1	100	150	15	20	0.9	0.033
ZDT2	100	200	15	20	0.9	0.033
ZDT3	100	1000	15	20	0.9	0.033
ZDT6	100	1000	15	20	0.9	1
DTLZ1	92	400	30	20	1	0.1428571
DTLZ2	92	250	30	20	1	0.083
DTLZ3	92	1000	30	20	1	0.083
DTLZ4	92	600	30	20	1	0.083

be observed. It is noted that MOEAs are run for 30 times with different initial populations for performance assessment.

4.4 Experimental Settings

In order to run RM^{2}OEA and $\theta-$DEA, some input parameters need to be fixed. For example, Table 1 presents different input parameters for running the MOEAs. It is noted that RM^{2}OEA gets terminated either by the adaptive termination condition discussed in Section 3.4 or by the maximum number of allowed generations (T_{max}), whichever is satisfied early. The input parameters for running the local search using MOM are given in Table 2. The MOM method gets terminated either by T_{MOM} or ϵ_{MOM}, whichever is satisfied early. The only parameters left for $\theta-$DEA are $\theta = 5$ and the neighborhood size is $T_{NB} = 20$. It uses PBI function for calculating the fitness. It is also noted that all variants of RM^{2}OEA and $\theta-$DEA are run with ($p = N - 1$) equal divisions for 2-objective MOOP and $p = 12$ equal divisions for 3-objective MOOP for generating the reference points. In the case of 3-objective problems, the number of reference points becomes 91.

Table 2 The details of input parameters for running local search using the MOM method with the steepest descent method. The maximum number of iterations for local search using the MOM method is denoted by T_{MOM}, and in each iteration, the number of sequences for running the steepest descent method is denoted by K_{SDM}.

Parameter	Value
Maximum iterations (T_{MOM})	5
Maximum sequences (K_{SDM})	10
Penalty (R)	100
Error for termination (ϵ_{MOM})	0.001

4.5 Comparison based on IGD$^+$ Values

Tables 3 to 6 presents the performance assessment comparison of three vari-
ants of RM^{2}OEA with $\theta-$DEA on 2-objective MOOPs. The IGD$^+$ values
are shown for different values of local search solutions (N_{LS}). Meaning, a
smaller value of N_{LS} suggests less number of local search solutions, and
vice-versa. Table 3 shows that RM^{2}OEA-OP and RM^{2}OEA-OF variants are
found to be better than $\theta-$DEA for any number of N_{LS} for ZDT1 problem.
Except for a few IGD$^+$ values, RM^{2}OEA-ENV variant is also found to be
better. An interesting observation can be seen at the median IGD$^+$ values
of all variants RM^{2}OEA that the local search module shows similar perfor-
mance irrespective of different N_{LS} values. From Table 3, it can also be seen
that RM^{2}OEA-OF is found to be the best among other variants. Therefore,
RM^{2}OEA-OF is made adaptive using the condition described in Section
3.1.3. The last column of the same table suggests that adaptive RM^{2}OEA-
OF shows better performance than $\theta-$DEA but unable to generate the same
quality of solutions as RM^{2}OEA-OF.

For ZDT2 problem, Table 4 shows that RM^{2}OEA-OP and RM^{2}OEA-
OF variants are found to be better than $\theta-$DEA for any number of N_{LS}.
RM^{2}OEA-ENV is not found competitive as other variants. Among them,
RM^{2}OEA-OF is found to be the best. However, its adaptive version in the
last column of the same table is not as good as RM^{2}OEA-OF, but better
than $\theta-$DEA. From the table, it can be seen that N_{LS} does not show much
impact on the performance of three variants of RM^{2}OEA.

For ZDT3 problem, Table 5 presents a comparison among MOEAs. Since
ZDT3 has disconnected PO fronts, not a single variant is found to be out-
performing $\theta-$DEA. With $N_{\text{LS}} = 72$, RM^{2}OEA-OF is found to be better
than $\theta-$DEA for all three statistical values of IGD$^+$. The adaptive version
of RM^{2}OEA-OF seems to be not helping RM^{2}OEA. It can also be seen from
the table that RM^{2}OEA-OP is unable to generate better IGD$^+$ values than
$\theta-$DEA.

For ZDT6 problem, Table 6 shows a better performance of RM^{2}OEA-OP,
RM^{2}OEA-OF and adaptive RM^{2}OEA-OF variants over $\theta-$DEA. Except for
$N_{\text{LS}} = 36$, RM^{2}OEA-ENV is unable to perform better than $\theta-$DEA. Among
all the variants, RM^{2}OEA-OF is found to better than others.

Tables 7 to 10 presents the performance comparison of all variants of
RM^{2}OEA with $\theta-$DEA for 3-objective DTLZ problems. For DTLZ1 prob-
lem, RM^{2}OEA-ENV is found to be the best among all variants. Except for
the worst IGD$^+$ values for all N_{LS}, RM^{2}OEA-OP, RM^{2}OEA-OF and its
adaptive version are found to be better than $\theta-$DEA. Regarding N_{LS}, the
performance of all variants of RM^{2}OEA seems to be similar.

For DTLZ2 problem, Table 8 shows that RM^{2}OEA-OF and its adaptive
version are found to be better than $\theta-$DEA. For this problem, adaptive
RM^{2}OEA-OF is found to be the best. RM^{2}OEA-OP and RM^{2}OEA-ENV

are unable to generate good IGD$^+$ values than $\theta-$DEA. The performance of RM^{2}OEA variants seems to be quite similar for different N_{LS} values.

For DTLZ3 problem, Table 9 shows better performance of all variants over $\theta-$DEA. Among the variants, RM^{2}OEA-OF is found to be the best. For this example as well, different N_{LS} values show the similar performance of variants of RM^{2}OEA.

For DTLZ4 problem, Table 10 show that only RM^{2}OEA-OF shows the best performance for different values of N_{LS} over $\theta-$DEA. RM^{2}OEA-ENV is better than $\theta-$DEA only for $N_{\mathrm{LS}} = 91$. Other variants are unable to perform better than $\theta-$DEA. Since this problem is biased, this could be a probable reason for not so good performance of other variants.

From Tables 3 to 10, it can be seen that RM^{2}OEA-OF is the best variant and shows better IGD$^+$ values against $\theta-$DEA in most for the problems. The adaptive version of RM^{2}OEA-OF does not seem to be as effective as RM^{2}OEA-OF, except for DTLZ2. Moreover, it can also be observed that N_{LS} seems to have not much impact on the performance of any variants of RM^{2}OEA. It can be due to the selection of good solutions for local search that helps global search of MOEA for better convergence.

4.6 Obtained PO Solutions

From Section 4.5, it can be seen that RM^{2}OEA-OF is found to be the best among three variants. In this section, the solutions corresponding to the median value of IGD$^+$ for both $\theta-$DEA and RM^{2}OEA-OF are shown. Since the performance of all variants is similar for different values of N_{LS}, the obtained PO solutions for the least N_{LS} are shown, that are, $N_{\mathrm{LS}} = 20$ for 2-objective problems and $N_{\mathrm{LS}} = 15$ for 3-objective problems.

Fig. 2 shows the obtained PO solutions of ZDT1 problem. The PO front of this problem is non-convex. It can be seen that $\theta-$DEA is unable to reach the corner along the f_2-axis as compared to RM^{2}OEA-OF. This observation is similar to the outcome of IGD$^+$ comparison. Fig. 3 shows the obtained PO solutions for ZDT2 problem. The PO front of this problem is convex. For this problem as well, RM^{2}OEA-OF generates solutions throughout the range of both objectives, which is not visible with $\theta-$DEA. Fig. 4 shows the obtained PO solutions for ZDT3 problem. This problem has disconnected PO front. The figures shows a better distribution of solutions of RM^{2}OEA-OF over $\theta-$DEA. Fig. 5 shows distribution of solutions of both MOEAs for ZDT6 problem. It is noted that this problem has convex PO front. It can be seen from the figure that both MOEAs show similar distribution of solutions.

Fig. 6 shows obtained PO solution for DTLZ1 problem which has linear PO front. This problem is difficult to solve because it has many local PO fronts before reaching to the true PO front. It can be seen from the

figure that both MOEAs show similar distribution of solutions for DTLZ1 problems. It can be observed that RM^{2}OEA-OF shows a better performance based on IGD$^+$ values, however, the difference among these values is small. Fig. 7 shows the obtained PO solutions of both MOEAs for convex DTLZ2 problem. For this problem as well, the distribution of solutions is found similar for both MOEAs. Fig. 8 shows the obtained PO solutions for convex DTLZ3 problem. However, this problem is difficult to solve since there exists many local PO fronts. The distribution of solutions is again found similar for both MOEAs. Fig. 9 shows the obtained PO solutions for convex DTLZ4 problem. The figure shows similar distribution of solutions of RM^{2}OEA-OF and θ−DEA.

4.7 Convergence Details

From Sections 4.5 and 4.6, it can be observed that RM^{2}OEA was effective and generated better PO solutions and IGD$^+$ values. In this section, the convergence of RM^{2}OEA-OF is compared with θ−DEA. Fig. 10 shows the convergence plots for ZDT problems. For ZDT1 problem, RM^{2}OEA-OF with different N_{LS} values shows quicker convergence than θ−DEA. In fact, the convergence plots of RM^{2}OEA-OF get flatten at around 60 generations and are converged after 100 generations as compared to the fixed number of generations, that is 150, for θ−DEA. It seems that RM^{2}OEA-OF with $N_{\text{LS}} = 80$ has the quickest convergence. For ZDT2 problem, RM^{2}OEA-OF with different N_{LS} values again shows quicker convergence. The quickest convergence is observed with $N_{\text{LS}} = 80$ local search solutions, which has converged in 86 generations as compared to the fixed number of generations of θ−DEA, that is, 200. For ZDT3 problem again, a quicker convergence of RM^{2}OEA-OF with different N_{LS} values can be seen, which has converged in less than 200 generations against the maximum number of generations allotted, that is, 1000. For ZDT6 problem, RM^{2}OEA-OF has converged in less than 400 generations. A sharp improvement can be seen after 220 generations. For the above 2-objective ZDT problems, RM^{2}OEA-OF with $N_{\text{LS}} = 80$ local search solutions is found to be the best for the quickest convergence.

Fig. 11 shows the convergence plots for 3-objective DTLZ problems. For DTLZ1 problem, local search seems to be effective after 150 generations. Therefore, RM^{2}OEA-OF with $N_{\text{LS}} = 36, 91$ values has converged in less than 350 generations as compared to the fixed generations for θ−DEA, that is, 400. For DTLZ2 problem, RM^{2}OEA-OF and θ−DEA show similar convergence. For DTLZ3 problem, RM^{2}OEA-OF with different N_{LS} values shows quicker convergence than θ−DEA. The quickest convergence is observed with $N_{\text{LS}} = 91$, which has converged in 658 generations. For DTLZ4 problem, RM^{2}OEA-OF and θ−DEA show a similar convergence. However,

RM^{2}OEA-OF gets terminated in less than 370 generations because of the adaptive termination condition described in Section 3.4.

Table 11 presents generations required, functional evaluations, and computational time of RM^{2}OEA and $\theta-$DEA for the convergence. For ZDT1 problem, hybridizing RM^{2}OEA-OF with the given set of local search solutions (N_{LS}) has reduced the number of generations. RM^{2}OEA-OF with $N_{LS} = 80$ requires the least number of generations that saves more than 50% against $\theta-$DEA. However, since several local searches are involved with RM^{2}OEA-OF, it requires more number of function evaluations and computational time than $\theta-$DEA. For ZDT2 problem, RM^{2}OEA-OF with $N_{LS} = 80$ requires the least number of generations that saves around 57% of generations than $\theta-$DEA. Although the number of function evaluations required by RM^{2}OEA-OF is more, computational time required by it is relatively less than $\theta-$DEA. For ZDT3 problem, RM^{2}OEA-OF with different N_{LS} values requires quite fewer generations than $\theta-$DEA that saves the computational time. However, more number of function evaluations is needed by RM^{2}OEA-OF. For ZDT6 problem, RM^{2}OEA-OF is needed quite a less number of generations. Therefore, RM^{2}OEA-OF needs fewer function evaluations for convergence, and computational time is also reduced. For DTLZ1 problem, almost a saving of 50 generations can be seen for RM^{2}OEA-OF with $N_{LS} = 36, 91$ values against $\theta-$DEA. The number of function evaluations and computation time is more than $\theta-$DEA. For DTLZ2 problem, RM^{2}OEA-OF and $\theta-$DEA need all allotted generations. The same observation can be seen in Fig. 10 in which both MOEAs have a similar trend. In this case, RM^{2}OEA-OF needs more number of function evaluations and computational time than $\theta-$DEA. For DTLZ3 problem, except for $N_{LS} = 51$ case, RM^{2}OEA-OF requires less number of generations than $\theta-$DEA that saves its computational time. However, the number of function evaluations of RM^{2}OEA-OF is more than $\theta-$DEA. For DTLZ4 problem, a good number of generations is saved by RM^{2}OEA-OF against $\theta-$DEA that reduces computational time. However, RM^{2}OEA-OF needs more function evaluations than $\theta-$DEA.

From the above discussion, it can be seen that a larger number of local search solutions always helps RM^{2}OEA-OF in quicker convergence in most of the problems. Since the number is large, it requires many function evaluations. However, fewer number of local search solutions is equally good in helping RM^{2}OEA-OF for better convergence, which further reduces functional evaluations and computational time. Although a clear and distinct trend cannot be seen between the convergence and N_{LS}, this study brings out the fact that fewer number of local search solutions can be useful for better and quicker convergence unless a good and effective set of solutions is selected for local search.

5 Conclusion

The local search module has been proposed, which was steered by the reference lines. It was coupled with a commonly used MOEA framework, and three variants of RM^{2}OEA were proposed. After solving 2-objective ZDT problems and 3-objective DTLZ problems, it was found that the performance of RM^{2}OEA-OF variant generated the best values of IGD$^+$ indicator for most of the problems. Since their IGD$^+$ values were quite close for RM^{2}OEA-OF and $\theta-$DEA, the obtained PO solutions were qualitatively found similar. However, the convergence plots suggested that RM^{2}OEA-OF required fewer generations than $\theta-$DEA. Since the local search module involved extra computations for the MOM method, the number of function evaluations and computation time were relatively more for RM^{2}OEA-OF than $\theta-$DEA. In some problems, however, RM^{2}OEA-OF outperformed $\theta-$DEA in terms of a number of generations required, which was even less than 50%. One limitation observed from the results of ZDT3 problem is that RM^{2}OEA and its variants may not be suitable for disconnected PO fronts. From the results, it can be concluded that a fewer local search solutions can also be useful for better and quicker convergence unless a good and effective set of solutions is selected for local search. In future work, the local search module can be extended for constraint MOOPs. The same module can be tested on engineering optimization problems for better convergence. The proposed algorithm can also be extended for solving many-objective optimization problems when the number of objectives is more three.

References

1. Barakat, N., Sharma, D.: Evolutionary multi-objective optimization for bulldozer and its blade in soil cutting. International Journal of Management Science and Engineering Management **14**(2), 102–112 (2019)

2. Bosman, P.A.N.: On gradients and hybrid evolutionary algorithms for real-valued multiobjective optimization. IEEE Transactions on Evolutionary Computation **16**(1), 51–69 (2012)

3. Chen, B., Zeng, W., Lin, Y., Zhang, D.: A new local search-based multiobjective optimization algorithm. IEEE Transactions on Evolutionary Computation **19**(1), 50–73 (2015). DOI 10.1109/TEVC.2014.2301794

4. Das, I., Dennis, J.E.: Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. SIAM Journal on Optimization **8**(3), 631–657 (1998). DOI 10.1137/S1052623496307510

5. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Systems **9**(2), 115–148 (1995)

6. Deb, K., Miettinen, K., Chaudhuri, S.: Toward an estimation of Nadir objective vector using a hybrid of evolutionary and local search approaches. IEEE Transactions on Evolutionary Computation **14**(6), 821–841 (2010). DOI 10.1109/TEVC.2010.2041667

7. Deb, K., Miettinen, K., Sharma, D.: A hybrid integrated multi-objective optimization procedure for estimating nadir point. In: M. Ehrgott, C.M. Fonseca, X. Gandibleux, J.K. Hao, M. Sevaux (eds.) Evolutionary Multi-Criterion Optimization, pp. 569–583. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)

8. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: A. Abraham, L. Jain, R. Goldberg (eds.) Evolutionary multiobjective optimization: theoritical advances and applications, pp. 105–145. Springer-Verlag, London (2005)

9. Ishibuchi, H., Masuda, H., Nojima, Y.: A study on performance evaluation ability of a modified inverted generational distance indicator. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 695–702. ACM (2015)

10. Ishibuchi, H., Murata, T.: Multi-objective genetic local search algorithm. IEEE Transaction on Evolutionary Computation **28**, 119–124 (1996)

11. Ishibuchi, H., Narukawa, K.: Some issues on the implementation of local search in evolutionary multiobjective optimization. In: K. Deb (ed.) Genetic and Evolutionary Computation – GECCO 2004, pp. 1246–1258. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)

12. Ishibuchi, H., Yoshida, T., Murata, T.: Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Transactions on Evolutionary Computation **7**(2), 204–222 (2003)

13. Jaszkiewicz, A.: Genetic local search for multi-objective combinatorial optimization. European Journal of Operational Research **137**(1), 50 – 71 (2002). DOI https://doi.org/10.1016/S0377-2217(01)00104-7. URL http://www.sciencedirect.com/science/article/pii/S0377221701001047

14. Ke, L., Zhang, Q., Battiti, R.: Hybridization of decomposition and local search for multiobjective optimization. IEEE Transactions on Cybernetics **44**(10), 1808–1820 (2014)

15. Kumar, A., Sharma, D., Deb, K.: A hybrid multi-objective optimization procedure using pcx based nsga-ii and sequential quadratic programming. In: 2007 IEEE Congress on Evolutionary Computation, pp. 3011–3018 (2007). DOI 10.1109/CEC.2007.4424855

16. Lara, A., Sanchez, G., Coello, C.A.C., Schutze, O.: HCS: A new local search strategy for memetic multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation **14**(1), 112–132 (2010). DOI 10.1109/TEVC.2009.2024143

17. Murata, T., Nozawa, H., Tsujimura, Y., Gen, M., Ishibuchi, H.: Effect of local search on the performance of cellular multiobjective genetic algorithms for designing fuzzy rule-based classification systems. In: Proceedings of the 2002

Congress on Evolutionary Computation, 2002. CEC '02, vol. 1, pp. 663–668 (2002). DOI 10.1109/CEC.2002.1007005

18. Purshouse, R.C., Fleming, P.J.: On the evolutionary optimization of many conflicting objectives. IEEE Transactions on Evolutionary Computation **11**(6), 770–784 (2007). DOI 10.1109/TEVC.2007.910138

19. Ram, L., Sharma, D.: Evolutionary and gpu computing for topology optimization of structures. Swarm and Evolutionary Computation **35**, 1 – 13 (2017). DOI https://doi.org/10.1016/j.swevo.2016.08.004

20. Ravindran, A., Ragsdell, K., Reklaitis, G.: Engineering Optimization: Methods and Applications: Second Edition. John Wiley and Sons (2007). DOI 10.1002/9780470117811

21. Sharma, D., Deb, K.: Generation of compliant mechanisms using hybrid genetic algorithm. Journal of The Institution of Engineers (India): Series C **95**(4), 295–307 (2014). DOI 10.1007/s40032-014-0127-z

22. Sharma, D., Deb, K., Kishore, N.: Towards generating diverse topologies of path tracing compliant mechanisms using a local search based multi-objective genetic algorithm procedure. In: Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on, pp. 2004 –2011. IEEE (2008)

23. Sharma, D., Deb, K., Kishore, N.N.: Domain-specific initial population strategy for compliant mechanisms using customized genetic algorithm. Structural and Multidisciplinary Optimization **43**(4), 541–554 (2011)

24. Sharma, D., Deb, K., Kishore, N.N.: Customized evolutionary optimization procedure for generating minimum weight compliant mechanisms. Engineering Optimization **46**(1), 39–60 (2014)

25. Sharma, D., Kumar, A., Deb, K., Sindhya, K.: Hybridization of sbx based nsga-ii and sequential quadratic programming for solving multi-objective optimization problems. In: 2007 IEEE Congress on Evolutionary Computation, pp. 3003–3010 (2007). DOI 10.1109/CEC.2007.4424854

26. Sindhya, K., Miettinen, K., Deb, K.: A hybrid framework for evolutionary multi-objective optimization. IEEE Transaction on Evolutionary Computation **17**(4), 495–511 (2013)

27. Yuan, Y., Xu, H., Wang, B., Yao, X.: A new dominance relation-based evolutionary algorithm for many-objective optimization. IEEE Transactions on Evolutionary Computation **20**(1), 16–37 (2016). DOI 10.1109/TEVC.2015.2420112

28. Zhang, Q., Li, H.: Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation **11**(6), 712–731 (2007). DOI 10.1109/TEVC.2007.892759

29. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms : Empirical results. Evolutionary Computation Journal **8**(2), 125–148 (2000)

Table 3 The IGD$^+$ values of three variants of RM^{2}OEA and $\theta-$DEA are presented for ZDT1 problem. The grey colored cells show inferior IGD$^+$ values of RM^{2}OEA than $\theta-$DEA.

N_{LS}	Value type	RM^{2}OEA-OP	RM^{2}OEA-ENV	RM^{2}OEA-OF	Adaptive (RM^{2}OEA-OF)
	Best			5.684e-03	
$\theta-$DEA	Median			6.513e-03	
	Worst			8.709e-03	
	Best	2.319e-03	5.913e-03	2.429e-03	3.120e-03
20	Median	2.957e-03	6.832e-03	3.037e-03	2.960e-03
	Worst	3.662e-03	8.536e-03	3.909e-03	3.399e-03
	Best	2.504e-03	5.680e-03	2.394e-03	3.051e-03
40	Median	3.027e-03	6.451e-03	3.072e-03	2.707e-03
	Worst	3.656e-03	8.018e-03	3.561e-03	3.625e-03
	Best	2.773e-03	5.180e-03	2.472e-03	3.176e-03
60	Median	3.222e-03	6.583e-03	3.080e-03	3.125e-03
	Worst	3.773e-03	8.622e-03	4.004e-03	3.400e-03
	Best	2.294e-03	5.553e-03	2.390e-03	3.413e-03
80	Median	2.974e-03	6.821e-03	2.909e-03	3.017e-03
	Worst	3.699e-03	8.006e-03	3.551e-03	3.446e-03
	Best	2.799e-03	5.070e-03	2.472e-03	2.861e-03
100	Median	3.268e-03	6.584e-03	2.997e-03	2.717e-03
	Worst	3.854e-03	8.125e-03	3.504e-03	3.624e-03

Table 4 The IGD$^+$ values of three variants of RM^{2}OEA and $\theta-$DEA are presented for ZDT2 problem. The grey colored cells show inferior IGD$^+$ values of RM^{2}OEA than $\theta-$DEA.

N_{LS}	Value type	RM^{2}OEA-OP	RM^{2}OEA-ENV	RM^{2}OEA-OF	Adaptive (RM^{2}OEA-OF)
	Best			3.503e-03	
$\theta-$DEA	Median			3.942e-03	
	Worst			5.063e-03	
	Best	2.160e-03	3.362e-03	2.172e-03	2.198e-03
20	Median	2.206e-03	4.043e-03	2.218e-03	2.273e-03
	Worst	2.256e-03	5.152e-03	2.525e-03	4.811e-03
	Best	2.116e-03	3.364e-03	2.179e-03	2.081e-03
40	Median	2.188e-03	4.195e-03	2.206e-03	2.219e-03
	Worst	2.314e-03	5.651e-03	2.334e-03	4.811e-03
	Best	2.107e-03	3.132e-03	2.156e-03	2.089e-03
60	Median	2.178e-03	4.064e-03	2.203e-03	2.266e-03
	Worst	2.214e-03	4.948e-03	2.349e-03	4.811e-03
	Best	2.161e-03	3.361e-03	2.130e-03	2.049e-03
80	Median	2.218e-03	4.021e-03	2.199e-03	2.248e-03
	Worst	3.697e-03	9.520e-03	2.236e-03	4.811e-03
	Best	2.149e-03	3.270e-03	2.186e-03	2.073e-03
100	Median	2.220e-03	4.055e-03	2.206e-03	2.234e-03
	Worst	2.346e-03	4.587e-03	2.256e-03	4.811e-03

Table 5 The IGD$^+$ values of three variants of RM^{2}OEA and $\theta-$DEA are presented for ZDT3 problem. The grey colored cells show inferior IGD$^+$ values of RM^{2}OEA than $\theta-$DEA.

N_{LS}	Value type	RM^{2}OEA-OP	RM^{2}OEA-ENV	RM^{2}OEA-OF	Adaptive (RM^{2}OEA-OF)
$\theta-$DEA	Best			1.351e-03	
	Median			1.878e-03	
	Worst			2.380e-03	
20	Best	1.526e-03	1.311e-03	1.349e-03	1.440e-03
	Median	1.937e-03	1.795e-03	1.779e-03	1.868e-03
	Worst	2.550e-03	2.355e-03	2.638e-03	3.765e-03
40	Best	1.467e-03	1.485e-03	1.326e-03	1.317e-03
	Median	2.017e-03	1.855e-03	1.888e-03	1.885e-03
	Worst	2.815e-03	2.317e-03	2.611e-03	2.823e-03
60	Best	1.565e-03	1.410e-03	1.166e-03	1.509e-03
	Median	2.236e-03	1.728e-03	1.766e-03	1.830e-03
	Worst	3.394e-03	2.524e-03	2.531e-03	3.528e-03
80	Best	1.561e-03	1.206e-03	1.430e-03	1.260e-03
	Median	2.555e-03	1.776e-03	1.802e-03	1.674e-03
	Worst	3.586e-03	3.193e-03	2.346e-03	2.884e-03
100	Best	1.620e-03	1.399e-03	1.359e-03	1.254e-03
	Median	2.404e-03	1.805e-03	1.708e-03	1.797e-03
	Worst	3.322e-03	2.644e-03	2.580e-03	2.996e-03

Table 6 The IGD$^+$ values of three variants of RM^{2}OEA and $\theta-$DEA are presented for ZDT6 problem. The grey colored cells show inferior IGD$^+$ values of RM^{2}OEA than $\theta-$DEA.

N_{LS}	Value type	RM^{2}OEA-OP	RM^{2}OEA-ENV	RM^{2}OEA-OF	Adaptive (RM^{2}OEA-OF)
$\theta-$DEA	Best			1.763e-03	
	Median			1.954e-03	
	Worst			2.038e-03	
20	Best	1.587e-03	1.741e-03	1.589e-03	1.554e-03
	Median	1.600e-03	1.953e-03	1.599e-03	1.643e-03
	Worst	1.614e-03	2.050e-03	1.609e-03	2.196e-03
40	Best	1.587e-03	1.752e-03	1.590e-03	1.589e-03
	Median	1.596e-03	1.947e-03	1.600e-03	1.670e-03
	Worst	1.628e-03	2.027e-03	1.608e-03	1.957e-03
60	Best	1.584e-03	1.864e-03	1.587e-03	1.601e-03
	Median	1.598e-03	1.982e-03	1.600e-03	1.664e-03
	Worst	1.622e-03	2.067e-03	1.611e-03	1.978e-03
80	Best	1.600e-03	1.834e-03	1.587e-03	1.563e-03
	Median	1.631e-03	1.974e-03	1.600e-03	1.626e-03
	Worst	1.687e-03	2.052e-03	1.611e-03	1.808e-03
100	Best	1.600e-03	1.848e-03	1.592e-03	1.565e-03
	Median	1.627e-03	1.967e-03	1.602e-03	1.631e-03
	Worst	1.679e-03	2.023e-03	1.611e-03	1.829e-03

Table 7 The IGD$^+$ values of three variants of RM^{2}OEA and $\theta-$DEA are presented for DTLZ1 problem. The grey colored cells show inferior IGD$^+$ values of RM^{2}OEA than $\theta-$DEA.

N_{LS}	Value type	RM^{2}OEA-OP	RM^{2}OEA-ENV	RM^{2}OEA-OF	Adaptive (RM^{2}OEA-OF)
	Best			4.560e-04	
$\theta-$DEA	Median			1.416e-03	
	Worst			3.250e-03	
	Best	2.940e-04	1.310e-04	1.770e-04	2.180e-04
15	Median	4.770e-04	3.910e-04	3.760e-04	4.230e-04
	Worst	2.444e-03	1.569e-03	1.592e-03	2.170e-02
	Best	3.540e-04	1.620e-04	1.710e-04	2.110e-04
36	Median	7.010e-04	3.400e-04	3.650e-04	3.680e-04
	Worst	3.353e-02	1.357e-03	1.266e-02	1.402e-02
	Best	2.630e-04	1.450e-04	1.320e-04	2.000e-04
51	Median	4.600e-04	3.070e-04	3.320e-04	3.550e-04
	Worst	1.954e-02	1.183e-03	3.657e-03	4.537e-03
	Best	2.790e-04	1.960e-04	1.540e-04	1.980e-04
72	Median	4.690e-04	2.890e-04	2.890e-04	3.450e-04
	Worst	2.800e-03	1.053e-03	7.276e-03	1.051e-02
	Best	3.350e-04	1.500e-04	1.370e-04	2.120e-04
91	Median	4.660e-04	3.680e-04	2.780e-04	3.290e-04
	Worst	8.611e-02	1.508e-03	1.799e-03	9.708e-03

Table 8 The IGD$^+$ values of three variants of RM^{2}OEA and $\theta-$DEA are presented for DTLZ2 problem. The grey colored cells show inferior IGD$^+$ values of RM^{2}OEA than $\theta-$DEA.

N_{LS}	Value type	RM^{2}OEA-OP	RM^{2}OEA-ENV	RM^{2}OEA-OF	Adaptive (RM^{2}OEA-OF)
	Best			5.630e-04	
$\theta-$DEA	Median			7.890e-04	
	Worst			2.023e-03	
	Best	4.410e-04	5.790e-04	4.570e-04	4.030e-04
15	Median	5.730e-04	7.760e-04	5.400e-04	5.060e-04
	Worst	7.540e-04	1.644e-03	6.980e-04	6.480e-04
	Best	1.747e-03	6.300e-04	3.900e-04	4.040e-04
36	Median	2.158e-03	8.150e-04	4.980e-04	4.730e-04
	Worst	2.672e-03	1.844e-03	5.950e-04	9.920e-04
	Best	2.926e-03	6.220e-04	3.790e-04	3.930e-04
51	Median	3.384e-03	7.670e-04	5.010e-04	4.640e-04
	Worst	3.802e-03	7.061e-03	6.280e-04	1.134e-03
	Best	4.629e-03	5.790e-04	4.130e-04	3.700e-04
72	Median	5.447e-03	7.910e-04	5.000e-04	4.740e-04
	Worst	6.461e-03	1.248e-03	6.850e-04	6.280e-04
	Best	5.232e-03	5.880e-04	3.960e-04	4.090e-04
91	Median	6.018e-03	7.840e-04	4.850e-04	4.880e-04
	Worst	6.962e-03	1.242e-03	6.270e-04	6.340e-04

Table 9 The IGD$^+$ values of three variants of RM^{2}OEA and $\theta-$DEA are presented for DTLZ3 problem. The grey colored cells show inferior IGD$^+$ values of RM^{2}OEA than $\theta-$DEA.

N_{LS}	Value type	RM^{2}OEA-OP	RM^{2}OEA-ENV	RM^{2}OEA-OF	Adaptive (RM^{2}OEA-OF)
$\theta-$DEA	Best			8.560e-04	
	Median			3.708e-03	
	Worst			1.117e-02	
15	Best	6.520e-04	7.920e-04	1.740e-04	3.200e-04
	Median	1.459e-03	2.960e-03	2.430e-04	4.600e-04
	Worst	4.202e-02	8.296e-03	1.207e-03	3.578e-03
36	Best	7.660e-04	7.880e-04	1.610e-04	2.450e-04
	Median	1.757e-03	3.498e-03	2.350e-04	4.260e-04
	Worst	9.226e-03	9.010e-03	4.626e-03	1.443e-03
51	Best	7.610e-04	6.570e-04	1.810e-04	2.670e-04
	Median	1.584e-03	3.153e-03	2.170e-04	3.840e-04
	Worst	2.042e-02	9.048e-03	5.490e-04	1.611e-03
72	Best	7.180e-04	8.370e-04	1.380e-04	2.460e-04
	Median	1.371e-03	3.208e-03	1.980e-04	4.010e-04
	Worst	8.303e-03	9.505e-03	3.850e-04	8.080e-04
91	Best	7.990e-04	5.930e-04	1.470e-04	2.930e-04
	Median	1.773e-03	3.523e-03	2.040e-04	4.100e-04
	Worst	2.269e-02	1.784e-02	1.052e-03	1.355e-03

Table 10 The IGD$^+$ values of three variants of RM^{2}OEA and $\theta-$DEA are presented for DTLZ4 problem. The grey colored cells show inferior IGD$^+$ values of RM^{2}OEA than $\theta-$DEA.

N_{LS}	Value type	RM^{2}OEA-OP	RM^{2}OEA-ENV	RM^{2}OEA-OF	Adaptive (RM^{2}OEA-OF)
$\theta-$DEA	Best			1.460e-04	
	Median			1.810e-04	
	Worst			2.460e-04	
15	Best	1.250e-04	1.500e-04	1.440e-04	2.080e-04
	Median	1.710e-04	1.860e-04	1.680e-04	3.230e-04
	Worst	2.360e-04	2.420e-04	2.270e-04	8.230e-04
36	Best	2.190e-04	1.470e-04	1.400e-04	2.220e-04
	Median	5.740e-04	1.880e-04	1.750e-04	2.980e-04
	Worst	6.370e-04	2.350e-04	4.270e-04	1.362e-03
51	Best	1.970e-04	1.410e-04	1.390e-04	2.220e-04
	Median	2.340e-04	1.900e-04	1.730e-04	3.030e-04
	Worst	2.990e-04	2.340e-04	2.180e-04	6.840e-04
72	Best	3.970e-04	1.450e-04	1.280e-04	1.870e-04
	Median	8.530e-04	1.840e-04	1.720e-04	3.370e-04
	Worst	9.280e-04	2.340e-04	2.170e-04	2.352e-01
91	Best	8.270e-04	1.410e-04	1.400e-04	2.150e-04
	Median	1.184e-03	1.790e-04	1.730e-04	3.050e-04
	Worst	1.321e-03	2.460e-04	2.380e-04	5.390e-04

(a) $\theta-$DEA

(b) RM²OEA-OF($N_{LS} = 20$)
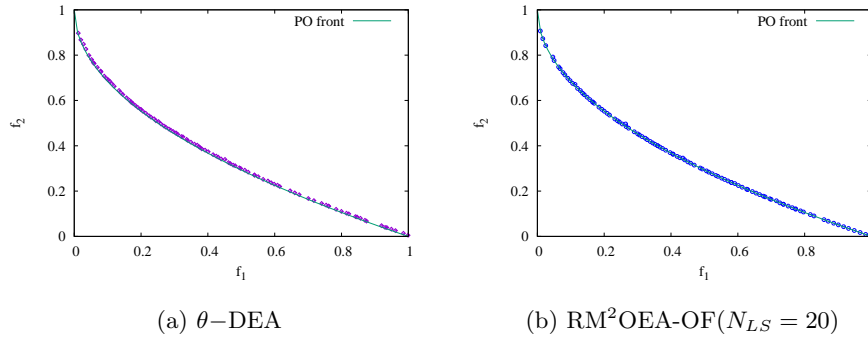
Fig. 2 The obtained PO solution of ZDT1 problem using $\theta-$DEA and RM²OEA-OF.

(a) $\theta-$DEA

(b) RM²OEA-OF($N_{LS} = 20$)
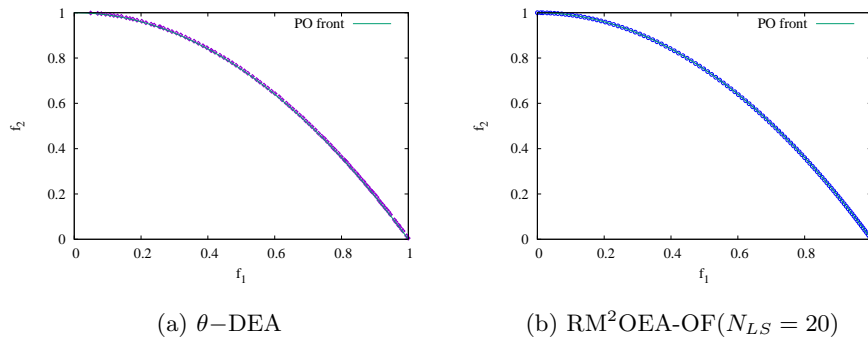
Fig. 3 The obtained PO solution of ZDT2 problem using $\theta-$DEA and RM²OEA-OF.

(a) $\theta-$DEA

(b) RM²OEA-OF($N_{LS} = 20$)
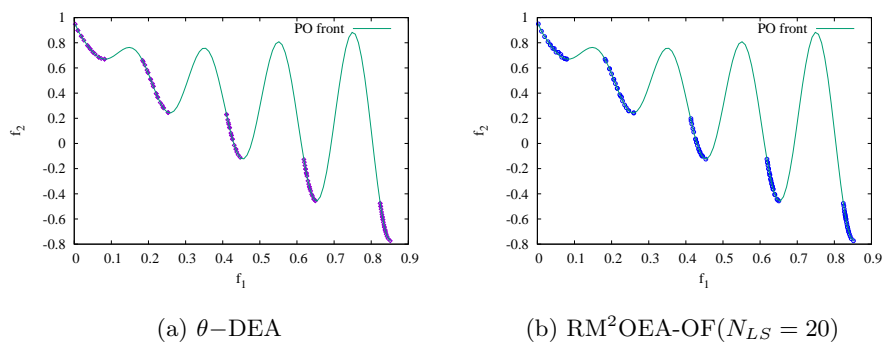
Fig. 4 The obtained PO solution of ZDT3 problem using $\theta-$DEA and RM²OEA-OF.

(a) $\theta-$DEA   (b) RM^{2}OEA-OF($N_{LS} = 20$)
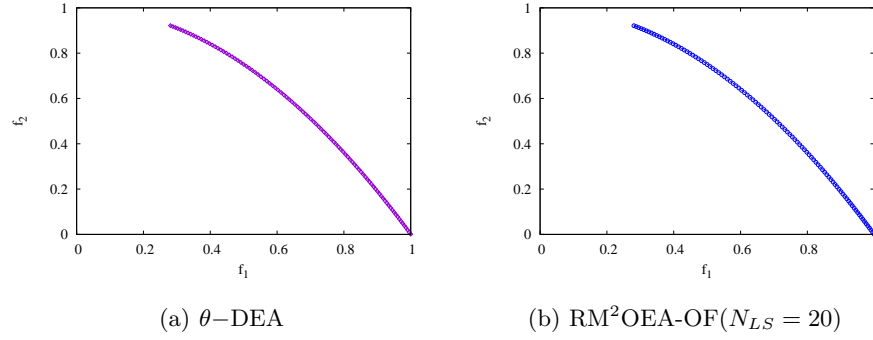
Fig. 5 The obtained PO solution of ZDT6 problem using $\theta-$DEA and RM^{2}OEA-OF.

(a) $\theta-$DEA   (b) RM^{2}OEA-OF($N_{LS} = 20$)
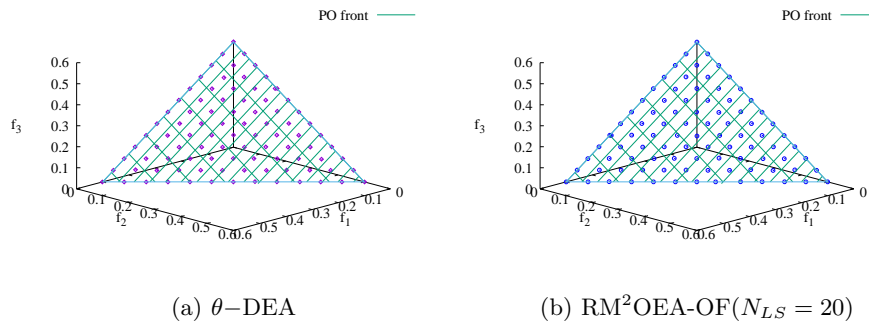
Fig. 6 The obtained PO solution of DTLZ1 problem using $\theta-$DEA and RM^{2}OEA-OF.

(a) $\theta-$DEA   (b) RM^{2}OEA-OF($N_{LS} = 20$)
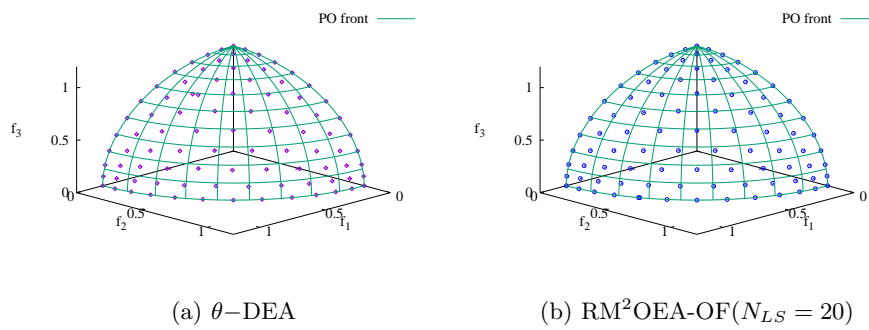
Fig. 7 The obtained PO solution of DTLZ2 problem using $\theta-$DEA and RM^{2}OEA-OF.

(a) $\theta-$DEA
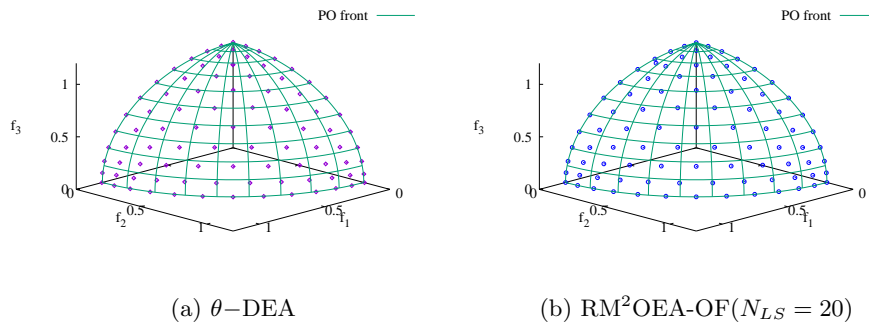
(b) RM^{2}OEA-OF($N_{LS} = 20$)

Fig. 8 The obtained PO solution of DTLZ3 problem using $\theta-$DEA and RM^{2}OEA-OF.

(a) $\theta-$DEA
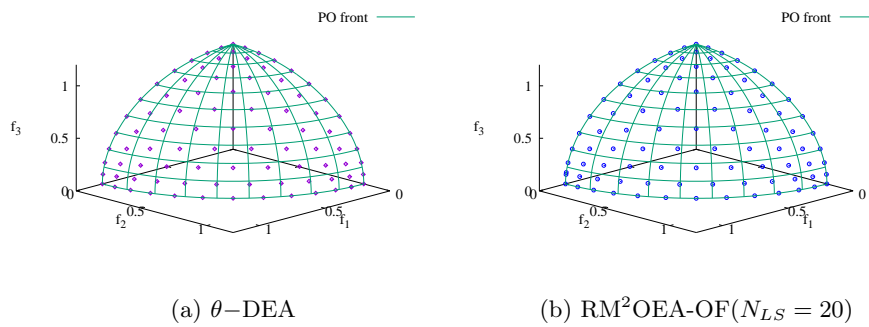
(b) RM^{2}OEA-OF($N_{LS} = 20$)

Fig. 9 The obtained PO solution of DTLZ4 problem using $\theta-$DEA and RM^{2}OEA-OF.

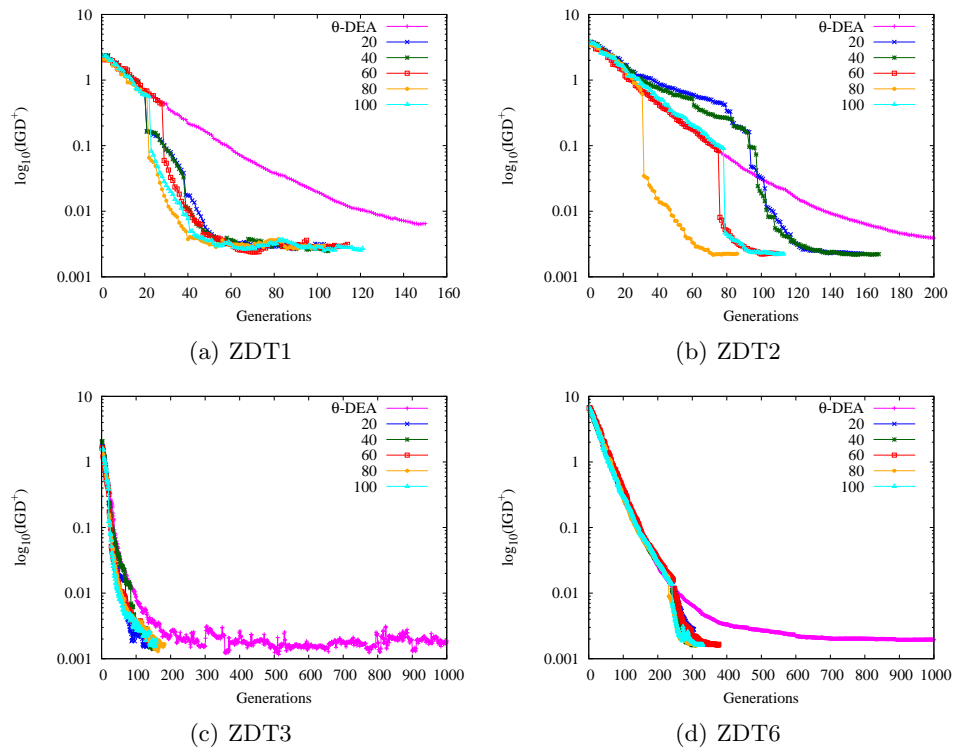(a) ZDT1

(b) ZDT2

(c) ZDT3

(d) ZDT6

Fig. 10 The convergence plots for ZDT problems are shown for $N_{\mathrm{LS}} = 20, 40, 60, 80$ and 100 values.
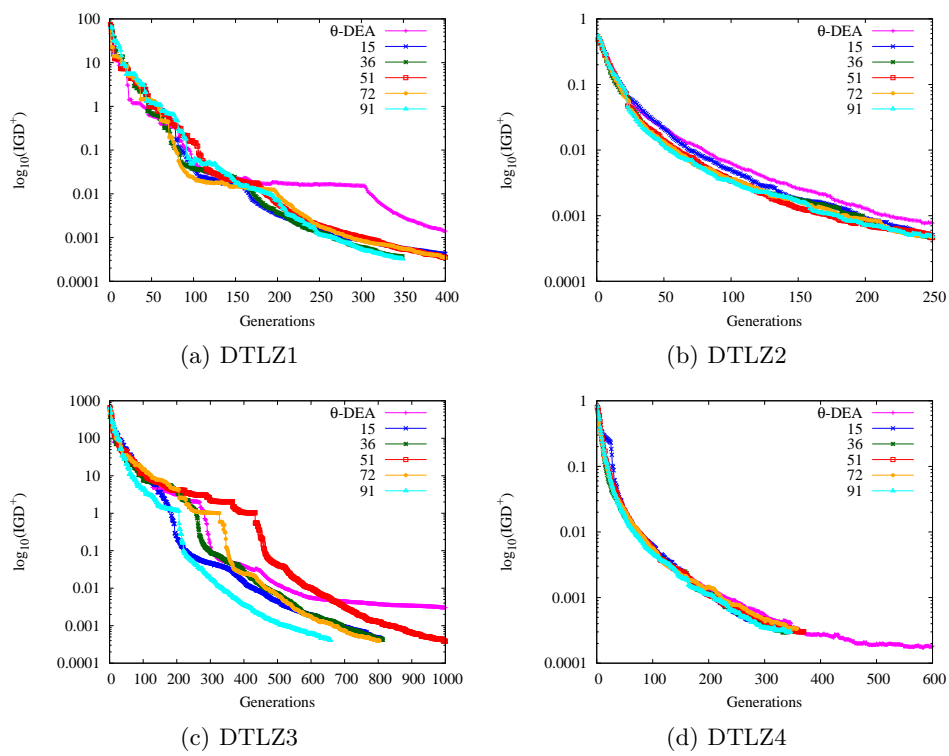
Fig. 11 The convergence plots for DTLZ problems are shown for $N_{\text{LS}} = 15, 36, 51, 72$ and 91 values.

Table 11 Number of generations required (T_{req}), function evaluations (N_{LS}), and computational time of $\theta-$DEA and RM^{2}OEA-OF are presented.

Parameters↓/N_{LS} →	$\theta-$DEA	20	40	60	80	100
ZDT1						
T_{req}	150	109	107	114	103	121
N_F	15600	134654	253338	321944	497256	555636
Time (s)	0.27541	0.24401	0.27851	0.30444	0.33711	0.41222
ZDT2						
T_{req}	200	158	168	109	86	113
N_F	20800	50690	83752	210772	429722	349840
Time (s)	0.39121	0.28887	0.35138	0.25822	0.28311	0.30487
ZDT3						
T_{req}	1000	136	162	148	182	158
N_F	100000	64592	136896	187596	278834	315794
Time (s)	1.80315	0.25417	0.31872	0.31262	0.40621	0.36356
ZDT6						
T_{req}	1000	344	312	377	325	331
N_F	100000	44528	39992	60684	48052	52334
Time (s)	1.49738	0.53585	0.46524	0.59838	0.51058	0.49767
Parameters↓/N_{LS} →	$\theta-$DEA	15	36	51	72	91
DTLZ1						
T_{req}	400	400	349	400	400	349
N_F	36800	52128	55234	72414	98476	125446
Time (s)	0.74954	0.79902	0.67349	0.77894	0.81808	0.72023
DTLZ2						
T_{req}	250	250	250	250	250	250
N_F	23000	94842	226414	362314	555782	644554
Time (s)	0.53868	0.58229	0.63465	0.70363	0.81148	0.84365
DTLZ3						
T_{req}	1000	809	814	1000	804	658
N_F	92000	85778	98568	139074	137810	143622
Time (s)	1.92902	1.55781	1.56201	2.04005	1.59781	1.33415
DTLZ4						
T_{req}	600	335	342	370	358	346
N_F	55200	41742	61342	66236	90950	98486
Time (s)	1.29538	0.74872	0.76548	0.83021	0.82583	0.83276