Applied Intelligence (2023) 53:30497–30522
https://doi.org/10.1007/s10489-023-05106-1

# Enhanced decomposition-based hybrid evolutionary and gradient-based algorithm for many-objective optimization

**Parviz Mohammad Zadeh**[1] · **Mostafa Mohagheghi**[1]

Accepted: 11 October 2023 / Published online: 20 November 2023


**Abstract**
This paper presents a novel decomposition-based hybrid many-objective optimization method using particle swarm optimization (PSO) and sequential quadratic programming (SQP) algorithms. In the proposed method, the objective function space decomposed into optimization sub-problems. In this respect, the origin of the coordinate systems of the scaled objective function space (i.e., ideal point) and uniformly distributed points in the scaled objective function space are used to generate reference directions. The scaling of the objective function space in each generation leads to the improvement of the accuracy of the ideal point, which is important because of its role in the generation of the reference directions. In the proposed method, optimization of all the reference directions performed simultaneously to increase the accuracy as well as the computational efficiency. In addition, the proposed method is based on the hybridization of multi-objective particle swarm optimization (MOPSO) using the reference direction and local fast and accurate search capabilities of the SQP to provide a computationally more efficient and accurate algorithm for solving many-objective optimization problems. In the proposed method, a new mutation operator was introduced into the MOPSO to enhance the computational performance, uniformity of the solution and to prevent the MOPSO from pre-convergence to the local Pareto front. The proposed method was compared with other many-objective optimization algorithms using several challenging DTLZ benchmark problems and real-world many-objective optimization problems involving 3 to 15 objective functions. The results show that the proposed method has a high diversity and very accurate Pareto solutions. In this respect, the proposed method outperforms other recently developed many-objective optimization algorithms used in this study. Based on the obtained results, the proposed method provides an effective way of solving many-objective optimization problems.

**Keywords** Many-objective optimization · Multi-objective particle swarm optimization (MOPSO) · Sequential quadratic programming (SQP) · Decomposition · Reference point · Multi-objective evolutionary algorithms (MOEAs)

## 1 Introduction

Most real-world engineering design optimization problems (e.g., aerospace [1, 2], automotive [3], and energy systems [4]) involve several and often conflicting design criteria. For example, the design optimization of aircraft structures involves several conflicting objectives such as weight, buckling, and reliability, and hence optimization must be conducted simultaneously to achieve a balanced optimum design among the various conflicting objectives. In the past decades, a significant number of research works have focused on the multi-objective evolutionary algorithms (MOEAs) [5–8], which have been used successfully for solving multi-objective optimization problems [9–11]. However, the increased dimensionality of objective function search space and high computational costs limit the application of MOEAs (e.g., NSGA-II and MOPSO) for solving many-objective optimization problems [12–14].

There are several research works focused on solving many-objective optimization problems using evolutionary algorithms, which can be divided into the following five different categories: (*a*) preference-based relations [15], such as the $\epsilon$-preferred [16], and the expansion dominance [17], that require the use of problem-specific parameters. (*ii*) indicator-based [18–20], which assign fitness to solutions using a scalar performance metric (indicator) [21]. (*iii*) dimensionality reduction [22], which are based on transforming

✉ Parviz Mohammad Zadeh
pmohammadzadeh@ut.ac.ir

[1] Faculty of New Sciences and Technology, North Kargar St. University of Tehran, Tehran, Iran

many-objective optimization problem into a reduced number of objective functions [23–25], and depend on the characteristics of the problem [26], and the selection of parameters to find sub-problems [27]. (*iv*) aggregation-based [28], such as weighted sum [29], and weighted Tchebycheff [30], which transform a many-objective optimization problem into a multi-objective optimization problem [31]. (*v*) decomposition-based algorithms [32–34], which divides the problem into a number of sub-problems [35]. The decomposition-based approach has successfully solved many-objective optimization problems [36, 37]. In general, the decomposition-based approach can be divided into; (*a*) aggregation functions and (*ii*) reference points [38]. Recently, MOEA/D-M2M [39] and the NSGA-III [40], based on the decomposition-based methods, utilize reference points [41].

In addition, several research works focused on using many-objective evolutionary algorithms, such as Pareto-based MOEAs, in this approach, the original Pareto-based dominance relation has been modified to enhance the convergence pressure for solving many-objective optimization problems, such as fuzzy-dominance [42], corner sorting [43], reference point-based dominance [44], and generalized Pareto-optimality [45]. Several research works focused on enhancing diversity management in Pareto-based MOEAs, such as the shift-based density estimation in SPEA2-SDE [46] and the use of associated reference points in NSGA-III [40]. In decomposition based MOEAs, two adaptive generation approaches were developed using weight vectors [47]. For example, the Pareto-based dominance was integrated with the decomposition-based approach in reference [48], using two external archives to ensure convergence and diversity. Despite the above mentioned advantages for solving many-objective optimization problems, these methods suffer from some drawbacks [49]. For instance, these algorithms do not guarantee convergence to the true Pareto-optimal front [50]. To overcome these issues, several research works focused on hybrid based approaches [51]. For example, Zhang et al. [52], proposed a hybrid algorithm by combining NSGA-III and knowledge-based optimization [53], into a many-objective optimization algorithm. In addition, angle-based decomposition [54], and encouragement-based boundary intersection decomposition [55], have been integrated to provide a balance between convergence and diversity [56].

As described above, many-objective optimization problem introduces new challenges to the existing optimization methods. This paper presents a novel decomposition-based hybrid many-objective optimization algorithm to address the challenges associated with the many-objective optimization problems. The main advantages of the proposed algorithm are as follows; (1) Integration of global search capabilities of MOPSO based on reference direction, and local fast and accurate search capabilities of gradient-based, sequential quadratic programming (SQP), to provide a computationally efficient and accurate algorithm for solving many-objective optimization problems. (2) Implementation of decomposition-based techniques to ensure the uniformity of the solution. (3) Implementation of a novel mutation operator for preventing the PSO from pre-convergence to the local optima as well as enhancing the computational performance of the MOPSO. In addition, the MOPSO is based on the reference direction, in which the information between reference points is utilized to enhance the efficiency of the MOPSO. Based on the obtained results, the proposed algorithm outperforms all other well-known many-objective optimization algorithms used in this study in terms of both diversity and convergence.

## 2 Methodology

The proposed methodology is based on the decomposition-based approach. In a decomposition-based approach, a set of reference directions or reference points are generated, in the objective function space, to guarantee uniformity of the results, which form the Pareto front. In reference point-based approaches, an objective function depends on the distance from the reference point.

The proposed methodology also utilizes the normal-boundary intersection (NBI) [57, 58], method to generate uniformly distributed reference points, which defines a set of reference points, as represented by ($W$), covering a hyperplane, as shown in Fig. 1.

In the proposed methodology, a composite objective function is used to define the objective function. In this approach, the distance from a search point in the objective function space to its corresponding reference point is to be minimized. Therefore, reference points must be located between the origin of the objective function space coordinate system and the true Pareto frontier.

### 2.1 Decomposition-based hybrid evolutionary and gradient-based many-objective optimization algorithm

The organization of the proposed decomposition-based hybrid evolutionary and gradient-based many-objective optimization algorithm is shown in Fig. 2 and described as follows:

In Fig. 2, $d_t$ is the search direction and $a_t$ is the step size in iteration $t$. The parameter $W$ represents the number of reference directions and $z$ is used as the number of iterations, for checking that all the reference directions are utilized (i.e., $W = z$) in the SQP.

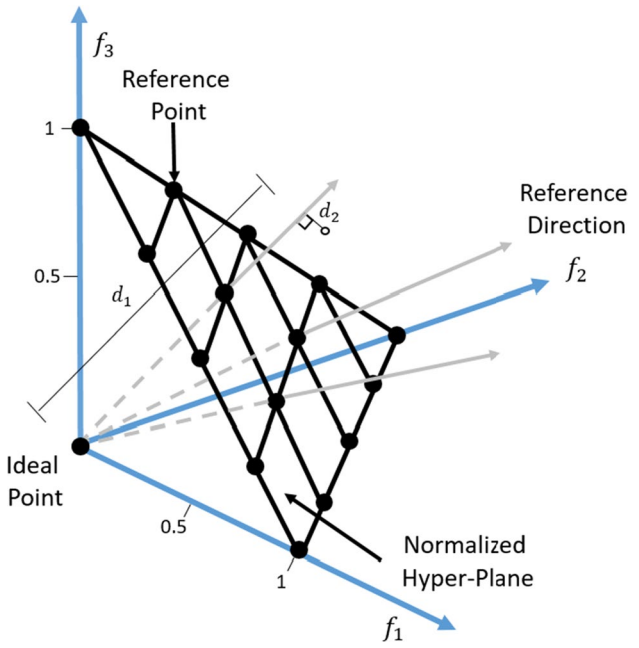



**Fig. 1** A set of neighbouring particles and reference directions in a normalized hyper-plane for $M=3$ ($M$ is the number of the objective functions). There are there are five reference points in the $f_1$-$f_2$ plane

A general many-objective particle swarm optimization (MOPSO) can be stated as below [59]:

$$\begin{aligned} &min: \boldsymbol{f}(\boldsymbol{x}) = \left[f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f3(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})\right], \boldsymbol{x} \in \Omega \\ &\quad s.t \qquad g_q(\boldsymbol{x}) \le 0, \quad q = 1, 2, \ldots, q \\ &\qquad\qquad h_s(\boldsymbol{x}) = 0, \quad s = 1, 2, \ldots, s \\ &side\ constraints: \ x_i^{lb} \le x_i \le \ x_i^{ub}, i = 1, 2, \ldots, n \end{aligned} \tag{1}$$

where $M$ is number of objective functions, $\Omega$ is the feasible search space, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, is the vector of $n$-dimensional decision variable. The mapping function, $\boldsymbol{f}(\boldsymbol{x}) : \Omega \to R^M$, defines $M$ objective functions, which are bounded in the objective space $R^M$. The inequality and equality constraints, which are represented as $g_q : \Omega \to R^q$ and $g_s : \Omega \to R^s$, respectively. In addition, $x_i^{lb}$ and $x_i^{ub}$, are the lower and upper bounds of $i$-th decision variables, respectively.

The proposed decomposition-based hybrid many-objective optimization algorithm for solving the above many-objective optimization problems utilizes several real reference points instead of a hypothetical point, as in the case of the Tchebycheff approach. In addition, the proposed methodology differs from the Tchebycheff approach as how the objective functions are defined. The main stages involved in the proposed methodology are shown in Fig. 2, as described below:

Stage 1: Decomposition of multi-objective optimization problems

In this stage, decomposition of multi-objective optimization problems into a single objective function, and initialization of the parameters of MOPSO (i.e., population size, the maximum number of generations, and stopping criteria), are defined. In the conventional MOPSO, the initial population size is provided as input to the MOPSO algorithm for the first reference point. In the proposed method, several evaluations are performed using the different versions of the MOPSO and Constriction Type 1. In order to update velocity and position of the *i-th* decision variable of the *j*-th particle [59], the following can be stated:

$$\begin{aligned} v_{i,j}^{t+1} &= w\left(v_{i,j}^t + \varphi_1\left(\boldsymbol{Pbest}_{i,j}^t - x_{i,j}^t\right) + \varphi_1\left(\boldsymbol{Gbest}_i^t - x_{i,j}^t\right)\right) \\ x_{i,j}^{t+1} &= v_{i,j}^{t+1}\Delta t + x_{i,j}^t \end{aligned} \tag{2}$$

where $v_{i,j,}$ and $x_{i,j,}$ are the current velocity and position of the *i-th* decision variable of the *j-th* particle. In the proposed method, $\boldsymbol{Pbest}_{i,j}^t$, is the *i-th* decision variable of the best position of the *j-th* particle in iteration $t$. $\boldsymbol{Gbest}_i^t$ is the *i-th* decision variable of the best global position whole population ever achieved. $\phi_1$ and $\phi_2$, are random number between [0, $\varphi/2$]. In this paper, φ is a constant and equals to 4.1. $w$ is the inertia weight parameter, and the value of this parameter used in this paper is 0.7298 [60]. $\Delta t$ is the time step for calculating the position of a particle from its velocity, which is conventionally equals to 1.

In the objective function space, reference points are based on the number of objective functions. The multi-objective optimization problem is decomposed into single-objective optimization problems by utilizing these reference points. Following this, the optimization process is performed for each reference point, as shown in Fig. 2. For all the reference points, the approximate range of the solution calculated using the MOPSO. A composite function defined as an objective function as described below in Stage 2.

Stage 2: Normal boundary intersection method for fitness assignment

In decomposition-based approaches, normal boundary intersection (NBI) [57], is a conventional way to obtain Pareto optimal. To obtain a point of Pareto front, the perpendicular distance to the reference line, $d_2$, should be zero, as shown in Fig. 1. At the same time, the distance between the origin of the coordinate system of the objective function space and the projection of the reference direction to the $f_1$ and $f_2$ plane, and the reference direction, $d_1$, to be minimized. In this respect, based on the preliminary studies and experimentations, $d_2$ is defined as a constraint. The definition of $d_2$ as a constraint is a problem dependent due to the different characteristics of the objective functions. The proposed algorithm utilizes a composite function as

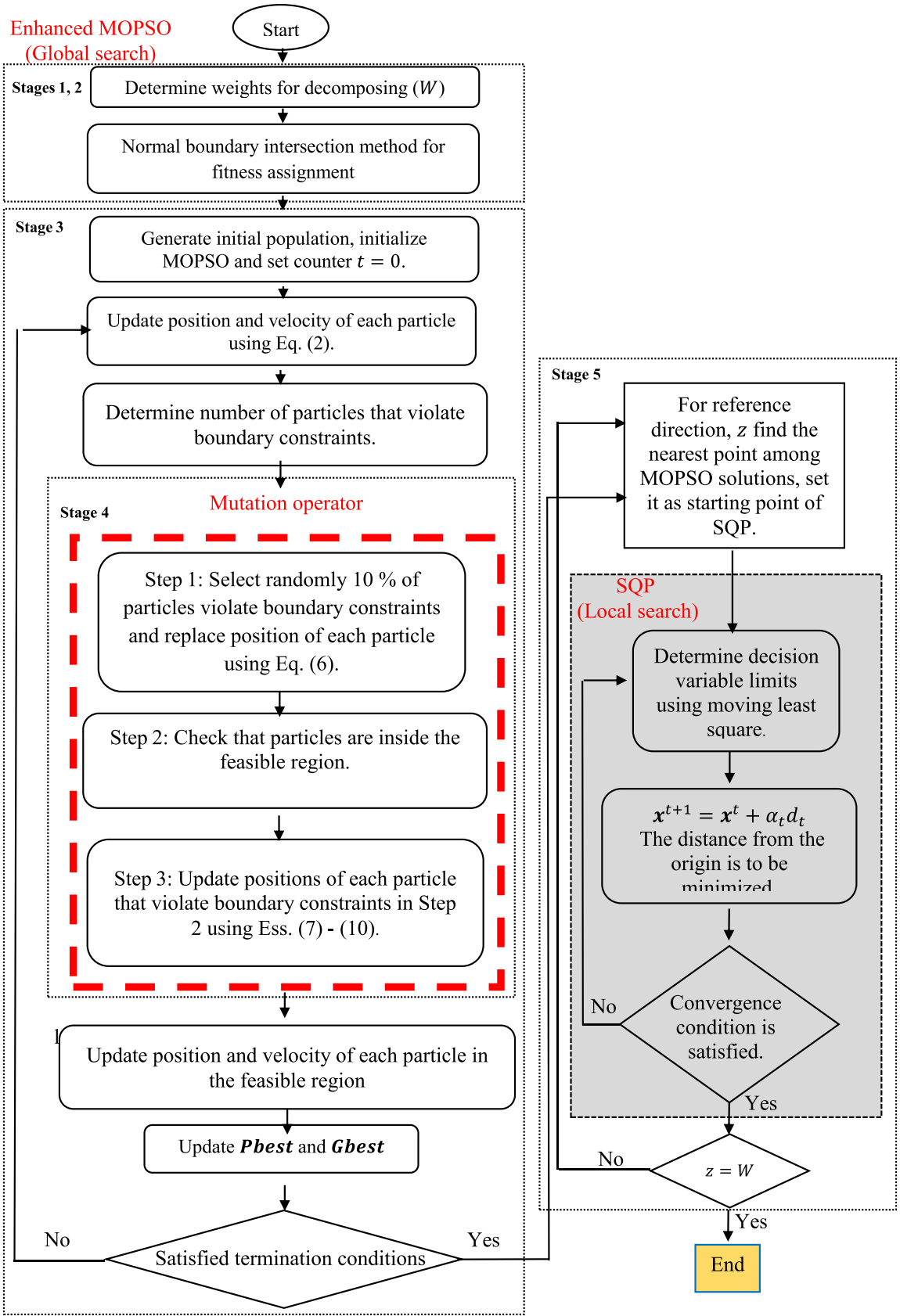



**Fig. 2** The proposed decomposition-based hybrid many-objective optimization algorithm

the objective function for both MOPSO and SQP optimization process. In this respect, in the proposed algorithm, a recently developed composite function ($F(\boldsymbol{x})$) for many-objective optimization is utilized [61]:

$$F(\boldsymbol{x}) = F_1(\boldsymbol{x}) + F_2(\boldsymbol{x}) + F_3(\boldsymbol{x}) \tag{3}$$

where,

$$\begin{aligned} F_1(\boldsymbol{x}) &= \min\left(\frac{f_M(\boldsymbol{x}) - f_M^{\min}}{f_M^{\max} - f_M^{\min}}\right) \\ F_2(\boldsymbol{x}) &= \sum_{M=1}^{M}\left(\frac{f_M(\boldsymbol{x}) - f_M^{\min}}{f_M^{\max} - f_M^{\min}}\right) \\ F_3(\boldsymbol{x}) &= \left(\sum_{M=1}^{M}\left|\frac{f_M(\boldsymbol{x}) - f_M^{0}}{f_M^{\max} - f_M^{0}}\right|^{p}\right)^{1/p} \end{aligned} \tag{4}$$

where $f_M(\boldsymbol{x})$ is the value of the $M$-th objective function for $\boldsymbol{x}$, $f_M^{\min}$ and $f_M^{\max}$ are the minimum and maximum values of $M$-th objective function, $f_M^0$ is the $M$-th component of a reference point, and $M$ represents number of the objective functions.

It should be mentioned that for the first iteration, the values obtained from the first function evaluation are used for determining $f_M^{\max}$ and $f_M^{\min}$, as well as the $F_1(\boldsymbol{x})$, $F_2(\boldsymbol{x})$, and $F_3(\boldsymbol{x})$. In addition, as shown in Eq. (3), in the case of the $F_3(\boldsymbol{x})$, $p$ is a parameter represented by $L_p$, which is a metric distance, and in this work, $p=2$ is used (Euclidian distance is considered).

The problem of the diversity (or the spread) of the solutions on the Pareto optimal frontier is associated with the different cases of distance functions, individually or in combination. The main problem is that the solutions can be sensitive to the choice of reference points. To overcome these problems, in this work, a new composite function (or fitness evaluation function) is utilized. This composite function involves the following three main features; (1) the best objective function value among the $M$ objective functions, (2) the sum of all the objective functions, and (3) the metric distance function, which measures the distance between the objective functions, and their ideal values (aspiration levels, target values or the reference point). According to these definitions, $F_1(x)$ is the best normalized value through the objective functions, $F_2(x)$ is the proximity measure, the sum of the differences between the composite objective functions, and the best value obtained for the corresponding component of the composite function. In addition, $F_3(x)$, is defined as the summation of the normalized Euclidian distance of the obtained point and the intended reference point. It should be noted that in the proposed method, reference points must be located between the origin of the coordinate system of the objective function space and the real Pareto frontier. In this respect, prior to the start of the main stages of the proposed algorithm, the hyper-plane must be contracted to ensure sufficient closeness of the reference points to the origin of the coordinate system of the objective function space.

Stage 3: Enhanced multi-objective particle swarm optimization (MOPSO):

As described above, the proposed hybrid decomposition-based many-objective optimization methodology integrates the global search capability of the MOPSO, with the local search capability of the SQP, as a hybrid many-objective optimization algorithm. In this respect, a new mutation operator (Stage 4) was introduced into the MOPSO to enhance the convergence of the proposed algorithm. In addition, the MOPSO is based on the reference direction, in which the information between reference points is utilized to enhance the efficiency of the MOPSO. The steps involved in the proposed method are described below:

Step 1: If the reference point counter is greater than 1; do.
Step 2: Among all the solutions, find the closest particle solution to the reference point (in the objective function space).
Step 3: The final population values for the closest particle position are set as an initial population used in the MOPSO.

As stated above, in the first step of the proposed algorithm, global optimization is performed utilizing the enhanced MOPSO algorithm for each of the reference points based on the many-objective optimization formulation. The objective function using the penalty function stated as below:

$$T(\boldsymbol{x}) = F(\boldsymbol{x}) + \mathrm{r_k}\left\{\sum_{i=1}^{m}\left[0, g_i(x)\right] + \sum_{j=1}^{l}\left[h_j(\boldsymbol{x})\right]\right\} \tag{5}$$

where $T(x)$ is the new objective function, $r_k$ is the penalty coefficient and $\sum_{i=1}^{m}\left[0, g_i(x)\right] + \sum_{j=1}^{l}\left[h_j(\boldsymbol{x})\right]$, are the penalty terms that are added to the objective function, $F(\boldsymbol{x})$. Penalty terms are proportional to the constraint violation.

In order to avoid the effects of stochastic nature of the MOPSO algorithm on the result, a multi-start strategy is used [62, 63].

In the decomposition-based approaches, the ideal point is an important concept influencing the convergence of the algorithm. In several research works, the ideal point is used to normalize the objective function values, and subsequently to normalize the objective function space. In some algorithms such as MOEA/D and NSGA-III, this point is defined as; $\mathbf{z} = (f_1^{\min}, f_2^{\min}, \ldots, f_i^{\min}, \ldots, f_M^{\min})$, where in the optimization process, $f_M^{\min}$ is the minimum of the $M$-th objective function, and $M$ is the number of objective functions [57]. Based on this concept, the position of the ideal point is

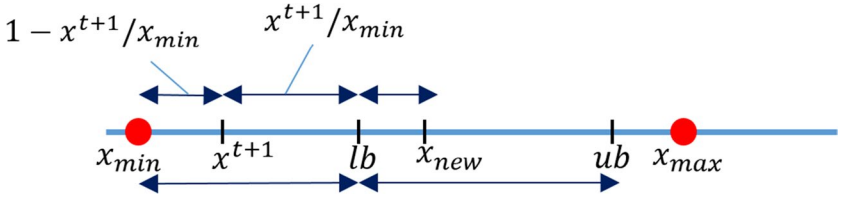



**Fig. 3** Replacing search points using the suggested mutation operator

improved during the search process, and the accuracy of the obtained ideal point gradually becomes close to the true ideal point. In this respect, the accuracy of the obtained ideal point is important for convergence.

In contrast, in some research works, the ideal point is defined as the origin of the coordinate system of the objective function space [58]. By considering the positive values for all the objective functions, the origin of the coordinate system of the objective function space is the absolute minimum for all the objective functions, because $f_M^{\min}$ cannot be less than its corresponding value in the origin of the coordinate system of the objective function space, which is zero. Hence, based on this definition, the position of the ideal point is fixed.

Normalization effectively solves problems with objective functions in different orders of magnitude [61]. In some algorithms, reference directions are generated by connecting the ideal point to each of the reference points [58]. In the present work, the origin of the coordinate system of the objective function space is used as an ideal point. This makes the solution to be independent of the scales of objective functions space.

In the proposed method, the composite function, $F_1(\boldsymbol{x})$, $F_2(\boldsymbol{x})$, and $F_3(\boldsymbol{x})$, Eqs. (3) and (4) are used, and normalization is performed. Therefore, the accuracy of the obtained ideal point gradually becomes close to the true solution point. In addition, in some approaches [61], the ideal point is the base point of the decomposition of the Pareto front. In contrast, in the proposed method, the decomposition performed only at the start of the algorithm and does not depend on the ideal point. This satisfies the requirements of the uniformity of solutions of the Pareto front without the need for updating ideal point, whereas identification process of the ideal point requires additional computation costs [61]. In the proposed method, the search is performed for each weight independently.

**Table 1** The variation of the coefficient δ, with respect to the reduced search space as the number of iterations increases

| Iteration | Δ |
|---|---|
| Number of iterations is less than the maximum number of iterations divided by 3 $\left(\frac{iter_{max}}{3}\right)$ | 0.5 |
| Number of iterations is greater than $\frac{iter_{max}}{3}$ and is less than $\frac{2\times iter_{max}}{3}$ | 0.3 |
| Number of iterations is greater than $\frac{2\times iter_{max}}{3}$ | 0.1 |

In Stage 2 of the proposed method, a parallelization scheme adopted, and the calculations for all the reference directions are conducted simultaneously in each iteration. The projection of the objective function vector on the closest reference direction used as the objective function.

Stage 4: Enhanced mutation operator for MOPSO algorithm

MOPSO algorithm often converges to local optima and suffers from premature convergence [64]. To overcome this challenge, the mutation operator, which is a genetic algorithm operator, was introduced into the MOPSO algorithm. In this respect, several experiments were conducted on the main characteristics of many-objective optimization problems. Based on the experimentation results, an enhanced variant of the mutation operator was introduced into the MOPSO algorithm. The main purpose of this enhanced mutation operator is to prevent the algorithm from premature convergence to the local optima and to improve the global search capability of the MOPSO algorithm. The proposed enhanced mutation operator affects the constraints handling processes. The steps involved in the proposed enhanced mutation operator are described as follows:

Step 1: Select randomly 10% of particles that violate the boundary constraints and replace their positions in random, using the range of feasible distance as stated below:

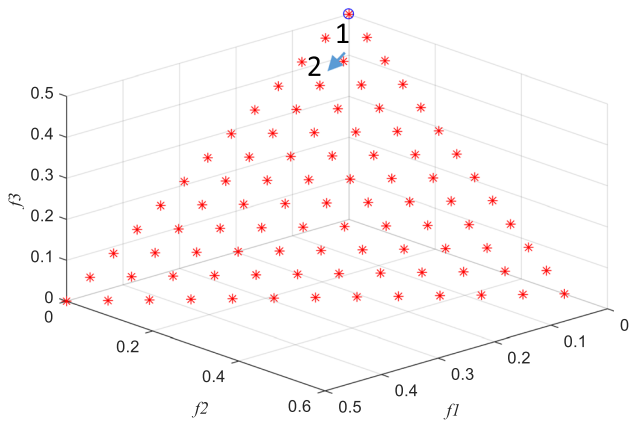



**Fig. 4** Neighboring point concept (using first reference point solution as an initial point of reference point 2 in SQP search)

**Table 2** DTLZ benchmark problems [65] (M = Number of objective functions)

| Test problem | The term used as power ($\alpha$) | Multiplying parameter ($K$) | M | Properties | Objective functions (that should be minimized) |
|---|---|---|---|---|---|
| DTLZ1 | - | 5 | 5, 8, 12, 16 | Multimodal | $f_1(\boldsymbol{x}) = 0.5(1+g)\prod_{i=1}^{M-1} x_i$<br>$f_{m=2:M-1}(\boldsymbol{x}) = 0.5(1+g)\left(\prod_{i=1}^{M-m} x_i\right)(1-x_{M-m+1})$<br>$f_M(\boldsymbol{x}) = 0.5(1+g)(1-x_1)$<br>where $g = 100\left(K + \sum_{i=M}^{M+K-1}\left(x_i-0.5\right)^2 - \cos\left(20\pi\left(x_i-0.5\right)\right)\right)$<br>$\boldsymbol{x} = (x_1, x_2, \ldots, x_{M+K-1})^T \in [0,1]^{M+K-1}$ |
| DTLZ2 | - | 10 | 5, 8, 12, 16 | Unimodal, concave | $f_1(\boldsymbol{x}) = (1+g)\prod_{i=1}^{M-1}\cos(x_i\pi/2)$<br>$f_{m=2:M-1}(\boldsymbol{x}) = (1+g)\left(\prod_{i=1}^{M-m}\cos(x_i\pi/2)\right)\sin\left(x_{M-m+1}\pi/2\right)$<br>$f_M(\boldsymbol{x}) = (1+g)\sin(x_1\pi/2)$<br>where $g = \sum_{i=1}^{M+K-1}\left(x_i-0.5\right)^2$, $\boldsymbol{x} = (x_1, x_2, \ldots, x_{M+K-1})^T \in [0,1]^{M+K-1}$ |
| DTLZ3 | - | 10 | 5, 8, 12, 16 | Multimodal, concave | $f_1(\boldsymbol{x}) = (1+g)\prod_{i=1}^{M-1}\cos(x_i\pi/2)$<br>$f_{m=2:M-1}(\boldsymbol{x}) = (1+g)\left(\prod_{i=1}^{M-m}\cos(x_i\pi/2)\right)\sin\left(x_{M-m+1}\pi/2\right)$<br>$f_M(\boldsymbol{x}) = (1+g)\sin(x_1\pi/2)$<br>where $g = 100\left(K + \sum_{i=M}^{M+K-1}\left(x_i-0.5\right)^2 - \cos\left(20\pi\left(x_i-0.5\right)\right)\right)$<br>$\boldsymbol{x} = (x_1, x_2, \ldots, x_{M+K-1})^T \in [0,1]^{M+K-1}$ |
| DTLZ4 | 100 | 10 | 5, 8, 12, 16 | Concave | $f_1(\boldsymbol{x}) = (1+g)\prod_{i=1}^{M-1}\cos(x_i\pi/2)$<br>$f_{m=2:M-1}(\boldsymbol{x}) = (1+g)\left(\prod_{i=1}^{M-m}\cos(x_i\pi/2)\right)\sin\left(x_{M-m+1}\pi/2\right)$<br>$f_M(\boldsymbol{x}) = (1+g)\sin(x_1\pi/2)$<br>where $g = 100\left(K + \sum_{i=M}^{M+K-1}\left(x_i-0.5\right)^2 - \cos\left(20\pi\left(x_i-0.5\right)\right)\right)$<br>$\boldsymbol{x} = (x_1, x_2, \ldots, x_{M+K-1})^T \in [0,1]^{M+K-1}$ |
| C1-DTLZ1 | - | 5 | 5, 8, 10, 15 | Multimodal | This is a C1-DTLZ1 test problem involving the following constraint:<br>$c(\boldsymbol{x}) = 1 - \frac{f_{M(x)}}{0.6} - \sum_{i=1}^{M-1}\frac{f_i(\boldsymbol{x})}{0.5} \geq 0$ |

**Table 2** (continued)

| Test problem | The term used as power ($\alpha$) | Multiplying parameter ($K$) | M | Properties | Objective functions (that should be minimized) |
|---|---|---|---|---|---|
| C2-DTLZ2 | 1 | 10 | 5, 8, 10, 15 | Unimodal, concave | This is a DTLZ2 test problem involving the following constraint: $c(\boldsymbol{x}) = \max\left\{\max_{i=1\ldots M}\left[\left(f_i(x)-1\right)^2 + \sum_{j=1\mid j\neq i}^{M} f_j^2 - r^2\right], \left[\sum_{i=1}^{M}\left(f_i(x) - 1/\sqrt{M}\right)^2 - r^2\right]\right\} \geq 0$ |
| C3-DTLZ4 | 100 | 10 | 5, 8, 10, 15 | Concave | The objective functions are the same as those were in the DTLZ2, with the following constraint: $c_j(\boldsymbol{x}) = \frac{f_j^2}{4} + \sum_{j=1\mid j\neq i}^{M} f_i^2 - 1^2 \geq 0 \forall j = 1, 2, \ldots, M$ |

$$x_{i,j} = r_{i,j} \times \left(x_{i,j}^{ub} - x_{i,j}^{lb}\right) + x_{i,j}^{lb}; \tag{6}$$

where $r_{i,j}$, is a random real number in the range of [0 1], $x_{i,j}$ is the current generation position of the $i$-th decision variable of the $j$-th particle, and $x_{i,j}^{ub}$ and $x_{i,j}^{lb}$ are the upper and lower bound of the $i$-th decision variable position of the *j-th* particle, respectively.

Step 2: Check that particles are inside the feasible region, which do not violate the boundary constraints.

Step 3: Calculate a ratio ($r_{i,j}$) and multiply it by a random number to produce a new particle position. A schematic view of this concept shown in Fig. 3. In this figure, $x_{i,j}^{t+1}$ presents particles that violate the boundary constraints, $1 - x_{i,j}^{t+1}/x_{i,j(min)}$ and $x_{i,j}^{t+1}/x_{i,j(max)}$, which is equal to the coefficient $r_{i,j}$, which guides the mutated particles to be located inside the feasible region that do not violate boundary constraints.

Indeed, suppose the position of a particle in iteration $t+1$, which violates a boundary constraint. In this case, the position of the particle updated to a feasible point $x_{i,j(new)}$, as described below.

To check, whether the position of the $i$-th decision variable of the $j$-th particle in iteration $t+1$ violates the lower boundary constraint, the following statement can be used:

$$\text{If} x_{i,j}^{t+1} < x_{i,j}^{lb} \tag{7}$$

where, $x_{i,j}$ is the current generation position of the $i$-th decision variable of the $j$-th particle and $t$ is the number of the iteration. $x_{i,j}^{ub}$ and $x_{i,j}^{lb}$ are the upper and lower bounds of the $i$-th decision variable position of the *j-th* particle, respectively.

If the lower boundary constrains violated, then the position of the particle can be updated using the following Eq. (8)

$$r_{i,j} = 1 - x_{i,j}^{t+1}/x_{i,j(min)} \text{ and hence } x_{i,j(new)} = r_{i,j} \times \left(x_{i,j}^{ub} - x_{i,j}^{lb}\right) + x_{i,j}^{lb} \tag{8}$$

Similarly, it can be examined whether the position of the $i$-th decision variable of the $j$-th particle in iteration $t+1$ violates the upper boundary constraint using the following statement:

$$\text{If } x_{i,j}^{t+1} > x_{i,j}^{ub} \text{ then} \tag{9}$$

If the lower boundary constrains violated, then the position of the particle can be updated using the following Eq. (10) as below:

$$r_{i,j} = 1 - x_{i,j}^{t+1}/x_{i,j(max)} \text{ and } x_{i,j(new)} = 1 - r_{i,j} \times \left(x_{i,j}^{ub} - x_{i,j}^{lb}\right) + x_{i,j}^{lb} \tag{10}$$

where $x_{i,j(new)}$, is the new position of the $i$-th decision variable of the $j$-th particle, $t$ is the number of iteration, $x_{i,j}^{lb}$ and $x_{i,j}^{ub}$ are the lower and upper bounds of $i$-th decision variable of the $j$-$th$ particle, respectively. $x_{i,j(min)}$, and $x_{i,j(max)}$, are the minimum and maximum values of the corresponding variables, respectively.

Stage 5: Gradient-based optimization using SQP

In this stage, the results of MOPSO are transferred as input to the SQP. In this respect, as the maximum generation number in MOPSO is reached for the final reference point, the values of obtained decision variables are transferred to the SQP to attain an accurate solution. The optimization using SQP completes when the imposed stopping criteria on SQP are satisfied. The stopping criteria used in this study are satisfaction of the constraints and the maximum number of iterations (i.e., the difference between two consecutive iterations must be less than 1e-6), which depends on the problem and required degree of accuracy, and this is defined by the user.

## 2.2 Sequential Quadratic Programming (SQP)

As described above, the proposed enhanced MOPSO algorithm is used to obtain global optimal solutions (corresponding to reference points), which are used to find a search direction using a linear approximation of constraints and quadratic approximation of objective functions in the SQP algorithm. The solution procedure is based on solving a quadratic programming sub-problem in each direction. Formulation of the SQP problem in iteration $t$ can be stated as:

$$\begin{aligned} \min: \quad & \tfrac{1}{2} d^T H_t d + \nabla F(\boldsymbol{x})^T d \\ \text{constraints}: \quad & \nabla h_a(\boldsymbol{x})^T d_t + h_a(\boldsymbol{x}) = 0 \quad a = 1, \ldots, a \\ & \nabla g_b(\boldsymbol{x})^T d_t + g_b(\boldsymbol{x}) < 0 \quad b = 1, \ldots, b \\ & x^{lb} \le \boldsymbol{x} \le x^{ub} \\ \text{Search space}: \quad & [\boldsymbol{x}_0 - \delta \times x_0, \boldsymbol{x}_0 - \delta \times x_0] \end{aligned} \tag{11}$$

where $d_t$ is the search direction, $t$ and $H_t$ is a positive definite approximation of the Hessian matrix of the Lagrangian function of the sub-problem. $\boldsymbol{x}$ is the vector of $n$ decision variables ($i = 1, 2, \ldots, n$), (the optimal solution from the MOPSO that is considered as the initial search point ($\boldsymbol{x}_0$) to the SQP), $x^{lb}$ and $x^{ub}$ are the lower and upper bounds of the decision variables. $g_b(\boldsymbol{x})$ and $h_a(\boldsymbol{x})$ are the inequality and equality constraints, respectively. $\delta$ is a coefficient that defines the range of search space for the SQP and is set according to Table 1.

The composite function $F(\boldsymbol{x})$ used to transfer the many-objective optimization problem into a single objective optimization problem in each direction in the optimization process. This makes it possible to utilize the SQP algorithm (as a single objective optimization method). The Lagrangian function can be expressed as [62]:

$$L(\boldsymbol{x}, u, v) = F(\boldsymbol{x}) + \sum_{a=1}^{a} u_a h_a(\boldsymbol{x}) + \sum_{b=1}^{b} v_b g_b(\boldsymbol{x}) \tag{12}$$

where $u_a$ and $v_b$ are the Lagrangian multipliers.

The active set strategy used to solve the sub-problem as stated below:

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t + a_t d_t \tag{13}$$

where $a_t$ is the step size parameter in iteration $t$, which is determined using a line search method so that a sufficient decrease in an objective function is obtained [63].

It should be mentioned that the optimization using the SQP was conducted in MATLAB software. In this respect,

**Table 3** Parameter settings of the population sizes and the number of function evaluations

| Benchmark problems | Number of objective functions ($M$) | Population size ($P$) |
|---|---|---|
| DTLZ1-DTLZ4 | 5 | 210 |
| | 8 | 330 |
| | 12 | 364 |
| | 16 | 816 |
| C1-DTLZ1 | 5 | 210 |
| C2-DTLZ2 | 8 | 156 |
| C3-DTLZ4 | 10 | 275 |
| | 15 | 135 |

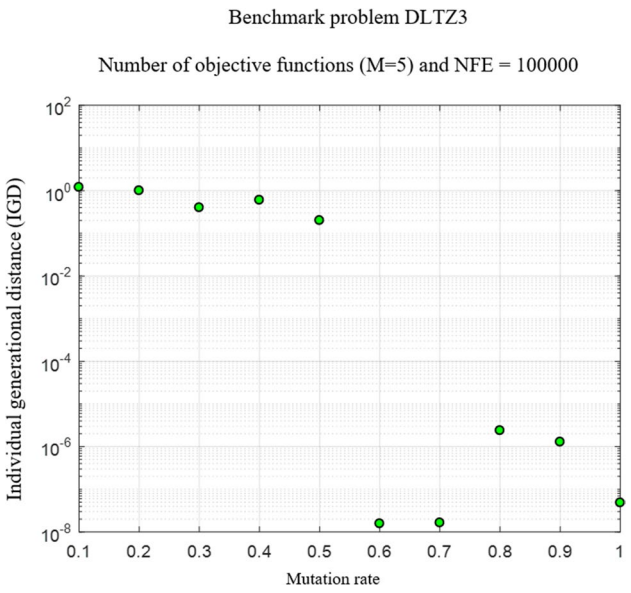



**Fig. 5** Individual generational distance (IGD) values for different mutation rates

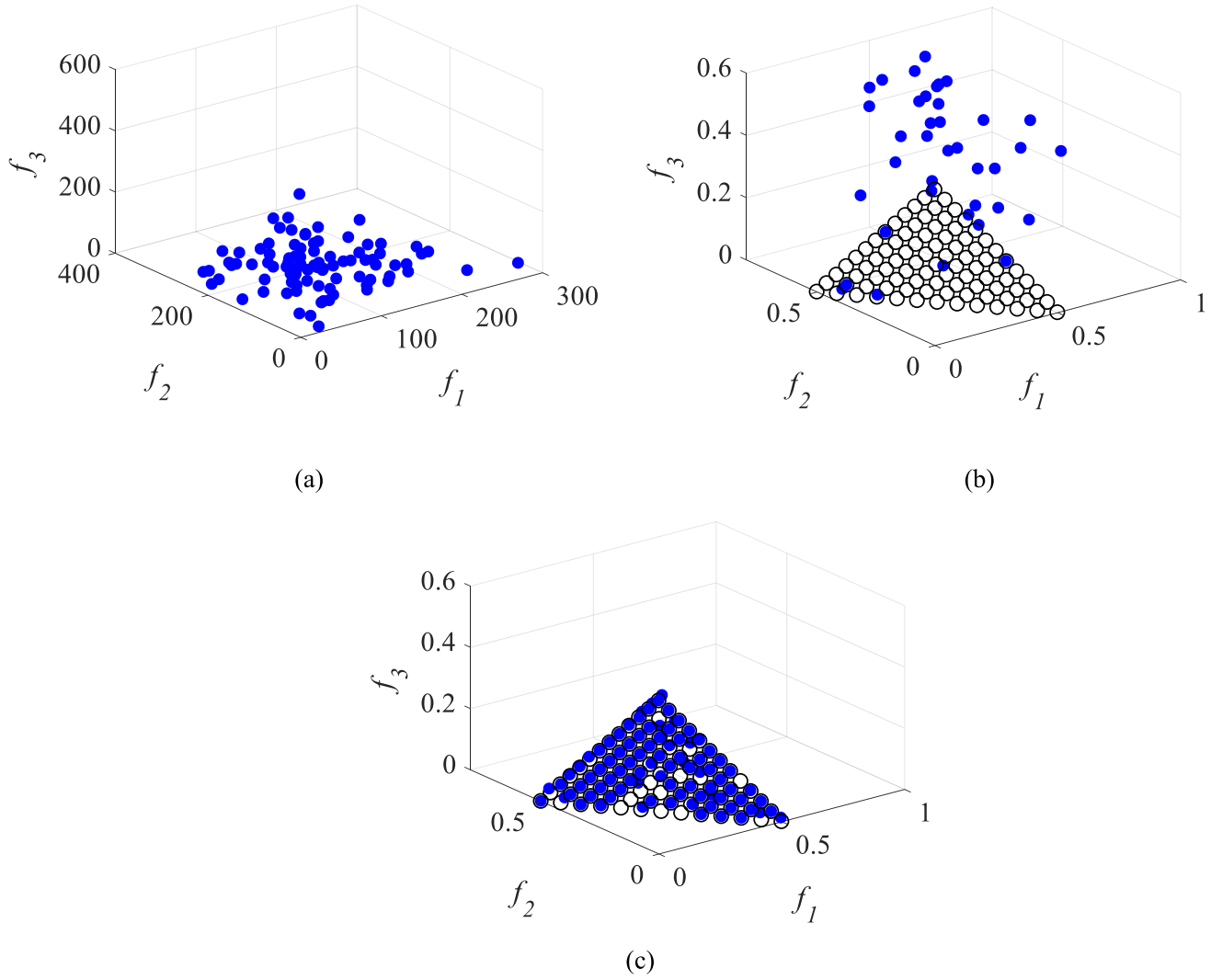



**Fig. 6** DTLZ1 (3 objective functions): **a** Initial population, **b** MOPSO result (without mutation operator), **c** MOPSO result (with mutation operator)

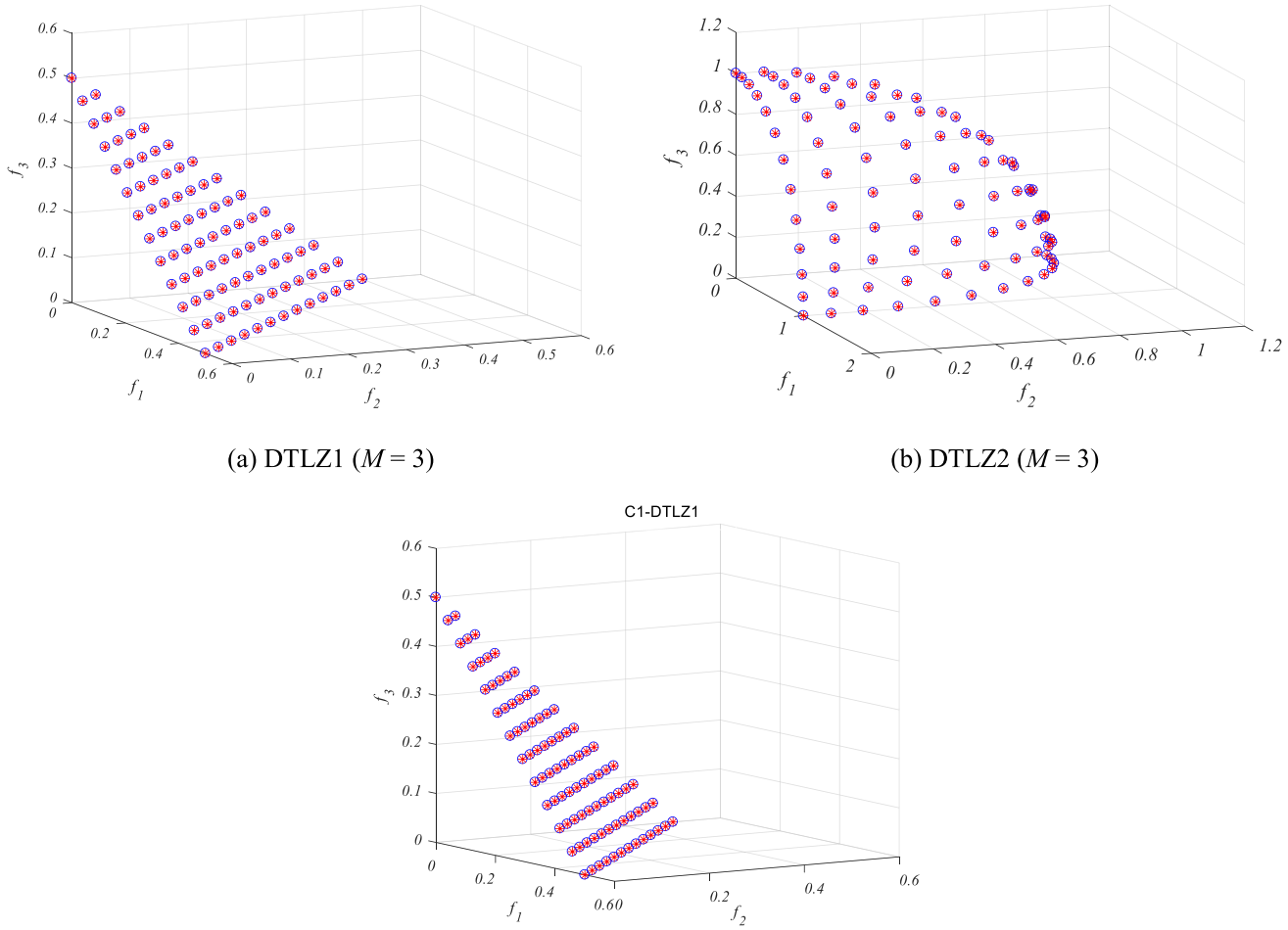
both the gradients and the Hessians approximated numerically for the reference points.

In the proposed algorithm, the moving square bounds method utilized to decrease the range of the decision variables. Towards this, the search space for the SQP is based on the output of the MOPSO using coefficient δ, which decreases through the iterations of the SQP algorithm as defined in Table 1:

In Table 1, $iter_{max}$ is the maximum number of iterations, which is used in the SQP algorithm.

## 2.3 Neighboring point technique

In some problems, specifically those that involve smooth functions (e.g., problems with unimodal functions), only one initial point of the Pareto front can be sufficient for the local search. In this respect, in the first step, a global search is performed using the proposed enhanced MOPSO algorithm, and an initial solution for the first reference point is obtained (point 1 in Fig. 4). In the second step, the result of the MOPSO algorithm used as an initial point for the SQP algorithm, which provides a more accurate solution for the reference point, which obtained in the first step. Following this, for reference point 2, a global search was not utilized, and only the local exploitation search using the SQP algorithm was conducted, in which the reference point 1 solution was considered as an initial point in the SQP algorithm. In this respect, for the next reference points, the closest reference point from the Pareto solutions is used as an initial point for the local search using the SQP algorithm without the need for a global search. In the optimization process, when the SQP algorithm converged, the solution added to the non-dominated solution set.




(a) DTLZ1 ($M = 3$)

(b) DTLZ2 ($M = 3$)

(c) C1-DTLZ1 ($M=3$)

**Fig. 7** Comparison between the proposed Pareto front and exact solution for **a** DTLZ1, **b** DTLZ2, **c** constrained DTLZ (C1-DTLZ1) with 3 objective functions

## 3 Experiments with benchmark problems

### 3.1 Experimental settings

Several experiments were conducted on well-defined optimization benchmark problems such as DTLZ1, DTLZ2, DTLZ3, and DTLZ4, with 3, 5, 8, 12, and 16 objective functions. In addition, C1-DTLZ1, C2-DTLZ2, and C3-DTLZ4, with 5, 8, 10, and 15 objective functions, to demonstrate the capability and computational efficiency of the proposed algorithm. These benchmark problems are mainly used to assess the computational accuracy and efficiency of many-objective optimization algorithms [38], as shown in Table 2.

In this comparative study of the of the performance of the proposed and other algorithms (e.g., RPD-NSGA-II, MOEA/DD, MOEA/D-DE, and NSGA-III, etc.), parameter setting was adopted as shown in Table 3, This involves, the population size, and the maximum number of function evaluations for each number of objective functions $M$. In this experimentation, for the unconstrained benchmark problems (DTLZ1, DTLZ2, DTLZ3, and DTLZ4), the same population sizes as in [33], were used, and for constrained benchmark problems (C1-DTLZ1), the parameter setting is similar to [38]. In comparisons, the median of the performance criteria was obtained using 10 independent iterations.

To determine the number of function evaluations (NFE) of the proposed method (i.e., MOPSO and SQP). The total NFE of MOPSO and SQP are used. In general, for a smooth function, a gradient-based algorithm requires less iteration than an evolutionary algorithm to obtain the optimum point. In this respect and based on the results of several experiments using different benchmark problems, it has been concluded that for a fixed number of function evaluations, if 90% of the number of function calls are assigned to the evolutionary algorithm, the proposed algorithm reaches the most accurate solutions.

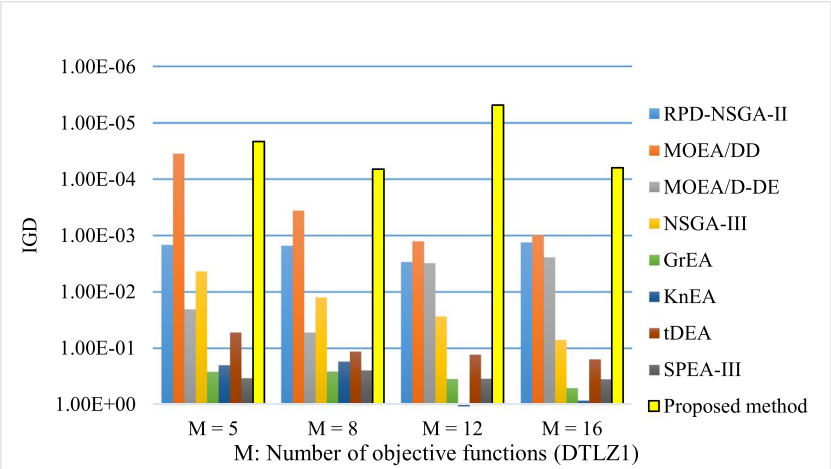



**Fig. 8** Comparison of results of the proposed algorithm and other algorithms for the DTLZ1 benchmark problem

**Table 4** The effect of hybridization of the MOPSO and SQP algorithms on the IGD. (M = Number of objective functions)

| Benchmark problem | M | MOPSO | | Proposed method | | I-DBEA [57] | |
|---|---|---|---|---|---|---|---|
| | | IGD | Simulation time (second) | IGD | Simulation time (second) | IGD | Simulation time (second) |
| DTLZ 1 | 3 | $1.27\times10^{-1}$ | 47.4 | $9.13\times10^{-7}$ | 162.3 | $1.75\times10^{-3}$ | 153.4 |
| DTLZ 1 | 5 | $6.32\times10^{-1}$ | 30.6 | $2.14\times10^{-5}$ | 106.8 | $9.43\times10^{-4}$ | 156.5 |
| DTLZ 2 | 3 | $5.47\times10^{-2}$ | 58.6 | $9.58\times10^{-14}$ | 53.8 | $6.37\times10^{-4}$ | 171.9 |
| DTLZ 2 | 5 | $1.58\times10^{-1}$ | 52.7 | $8.70\times10^{-14}$ | 48.4 | $1.12\times10^{-3}$ | 231.2 |

**Table 5** The effect of the proposed mutation operator in the MOPSO algorithm on the spacing and IGD performance metrics using the C1-DTLZ1 benchmark problem. (NFE = Number of function evaluation)

| NFE | 5,000 | | | 10,000 | | | 20,000 | | |
|---|---|---|---|---|---|---|---|---|---|
| Algorithm | Criteria | | Simulation time (second) | Criteria | | Simulation time (second) | Criteria | | Simulation time (second) |
| | Spacing | IGD | | Spacing | IGD | | Spacing | IGD | |
| GrEA | NaN | NaN | NaN | NaN | NaN | NaN | 2.0759e-2 | 6.8772e-2 | 16.5 |
| IBEA | NaN | NaN | NaN | NaN | NaN | NaN | 2.2003e-1 | 1.7383e-2 | 17.9 |
| KnEA | NaN | NaN | NaN | NaN | NaN | NaN | 2.6951e-2 | 3.2562e-2 | 16.0 |
| MOEAD | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| NSGAII | NaN | NaN | NaN | NaN | NaN | NaN | 2.0115e-2 | 3.0906e-2 | 16.1 |
| tDEA | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 6.0589e-2 | 16.3 |
| EMOPSO | 3.167e-2 | - | 51 | 2.1231e-2 | - | 103 | 1.9305e-2 | - | 15.0 |
| MOPSO + the proposed mutation operator | **3.989e-1** | **1.3594e + 0** | **56** | **1.9400e-2** | **2.0600e-1** | **142** | **1.0100e-2** | **8.7000e-2** | **15.3** |

Values in boldface show the best performance of the algorithms as shown in the tables

In the process of the parameters setting of the proposed algorithm, consideration was given to establishing the best value of the mutation rate in the proposed enhanced MOPSO algorithm. In this respect, an experiment was carried out using the proposed MOPSO algorithm with ten different mutation rate values (i.e., between 0.1 and 1). The results of this experimentation are shown in Fig. 5. As can be seen from Fig. 5, for the mutation rate of 0.6 and 0.7, the accuracy of results is better than other values of the mutation rate. Therefore, the parameter of the mutation rate was set to 0.6 for solving the benchmark problems.

Experiments with a benchmark problem with and without adopting the proposed mutation operator in the proposed MOPSO are performed to demonstrate the capability of the proposed mutation operator. The effect of the proposed mutation operator on the convergence of solutions to the

**Table 6** Neighboring points affect the number of function evaluations (NFE) for DTLZ1. (M = Number of objective functions)

| Problem | The proposed algorithm without the neighbouring point technique | | The proposed algorithm with the neighbouring point technique | |
|---|---|---|---|---|
| | IGD | NFE | IGD | NFE |
| DTLZ1 (M = 3) | $8.14\times10^{-5}$ | 1,000,000 | $9.13\times10^{-7}$ | 100,000 |

Pareto optimal front is shown in Fig. 6, using the benchmark problem, DTLZ-1, with 3 objective functions and 91 reference points.

The results of this experimentation and association of the solutions corresponding to each reference point, with 10,000 fitness evaluations, are shown in Fig. 7. In this respect, Fig. 5(a) and (b), show that the solutions without the mutation operator limited particles are generated in the feasible region, which are not close to the Pareto front, whilst Fig. 7(c), with the mutation operator show that a large number of particles are generated in the feasible region. This experimentation, as described above, clearly illustrates the search capability of the proposed mutation operator in the EMOPSO algorithm. The proposed mutation operator provides fast convergence of the particles to the feasible region and hence generation of the Pareto front.

## 3.2 Performance metrics

In general, there are different performance metrics for evaluating multi-objective optimization algorithms, for

**Table 7** Median IGD values for the DTLZ4 benchmark problem (bold items show the best performance). (M = Number of objective functions). (NFE = Number of function evaluations)

| Problem | NFE | 50,000 | | 75,000 | | 100,000 | |
|---|---|---|---|---|---|---|---|
| | Method | IGD | Simulation time (second) | IGD | Simulation time (second) | IGD | Simulation time (second) |
| DTLZ4 (M = 12) | RPD-NSGA-II | $3.39\times10^{-3}$ | 32.3 | $3.44\times10^{-3}$ | 47.8 | $3.13\times10^{-3}$ | 63.5 |
| | MOEA/DD | $5.90\times10^{-4}$ | 353.6 | $2.81\times10^{-4}$ | 527.7 | $2.60\times10^{-4}$ | 705.4 |
| | MOEA/D-DE | $6.26\times10^{-3}$ | 47.6 | $6.69\times10^{-3}$ | 71.2 | $6.79\times10^{-3}$ | 94.2 |
| | NSGA-III | $7.77\times10^{-3}$ | 17.9 | $7.07\times10^{-3}$ | 26.3 | $7.56\times10^{-3}$ | 41.4 |
| | GrEA | $4.5379\times10^{-1}$ | 18.8 | $4.5400\times10^{-1}$ | 28.7 | $4.5413\times10^{-1}$ | 36.3 |
| | KnEA | $4.9407\times10^{-1}$ | 19.4 | $4.9607\times10^{-1}$ | 28.5 | $4.8998\times10^{-1}$ | 39.9 |
| | tDEA | $5.00\times10^{-1}$ | 18.2 | $4.98\times10^{-1}$ | 27.6 | $4.98\times10^{-1}$ | 37.1 |
| | SPEA-III | $9.58\times10^{-1}$ | 17.8 | $6.75\times10^{-1}$ | 25.8 | $1.63\times10^{-2}$ | 35.3 |
| | MOSA | $1.87\times10^{0}$ | 18.5 | $7.643\times10^{-2}$ | 27.7 | $4.56\times10^{-3}$ | 37.1 |
| | Proposed method | $\mathbf{2.79\times10^{-2}}$ | **17.5** | $\mathbf{2.03\times10^{-10}}$ | **26.6** | $\mathbf{4.23\times10^{-11}}$ | **46.3** |

Values in boldface show the best performance of the algorithms as shown in the tables

**Table 8** Median IGD values with NFE = 100,000 for the DTLZ1 benchmark problem (bold items show the best performance). (*M* = Number of objective functions)

| Problem | DTLZ1 (*M* = 5) | | DTLZ1 (*M* = 8) | | DTLZ1 (*M* = 12) | | DTLZ1 (*M* = 16) | |
|---|---|---|---|---|---|---|---|---|
| | IGD | Simulation time (second) | IGD | Simulation time (second) | IGD | Simulation time (second) | IGD | Simulation time (second) |
| RPD-NSGA-II | 1.46 e -3 | 104.4 | 1.52 e -3 | 145.1 | 2.95 e -3 | 200.0 | 1.33 e -3 | 290.9 |
| MOEA/DD | 3.54 e -5 | 102.2 | 3.60 e -4 | 144.2 | 1.27 e -3 | 194.3 | 9.73 e -4 | 284.3 |
| MOEA/D-DE | 2.03 e -2 | 107.1 | 5.27 e -2 | 148.5 | 3.10 e -3 | 204.5 | 2.43 e -3 | 301.3 |
| NSGA-III | 4.33 e -3 | 111.0 | 1.26 e -2 | 156.9 | 2.74 e -2 | 209.5 | 7.18 e -2 | 309.1 |
| GrEA | 2.5874 e -1 | 112.2 | 2.5588 e -1 | 158.2 | 3.4611 e -1 | 214.2 | 5.0454 e -1 | 315.9 |
| KnEA | 1.9834 e -1 | 116.1 | 1.7076 e -1 | 162.2 | 1.2822 e 0 | 221.2 | 8.3971 e -1 | 325.0 |
| tDEA | 5.2796 e -2 | 110.7 | 1.1517 e -1 | 155.8 | 1.3037 e -1 | 211.6 | 1.5574 e -1 | 311.5 |
| SPEA-III | 3.36 e -1 | 117.0 | 2.44 e -1 | 163.1 | 3.46 e -1 | 224.3 | 3.51 e -1 | 329.3 |
| MOSA | 5.06 e -3 | 117.9 | | | | | | |
| Proposed method | **2.14 e -5** | **106.8** | **6.67 e -5** | **150.7** | **4.81 e -6** | **204.1** | **6.24 e -5** | **299.5** |

Values in boldface show the best performance of the algorithms as shown in the tables

**Table 9** Median IGD values for the DTLZ2 benchmark problem (bold items show the best performance). (M = Number of objective functions)

| Problem | DTLZ2 (M = 5) | | DTLZ2 (M = 8) | | DTLZ2 (M = 12) | | DTLZ2 (M = 16) | |
|---|---|---|---|---|---|---|---|---|
| | IGD | Time (second) | IGD | Time (second) | IGD | Time (second) | IGD | Time (second) |
| RPD-NSGA-II | 1.48 e -3 | 56.3 | 2.98 e -3 | 70.4 | 4.40 e -3 | 84.3 | 1.78 e -3 | 133.2 |
| MOEA/DD | 2.03 e -5 | 483.4 | 5.94 e -5 | 595.1 | 2.14 e -4 | 711.6 | 2.38 e -3 | 2317.1 |
| MOEA/D-DE | 1.61 e -3 | 87.3 | 4.23 e -2 | 88.9 | 4.43 e -3 | 76.3 | 2.44 e -3 | 108.9 |
| NSGA-III | 2.01 e -3 | 39.7 | 4.32 e -3 | 42.8 | 7.34 e -3 | 58.7 | 9.20 e -3 | 60.2 |
| Proposed method | **8.7 e -14** | **48.4** | **7.07 e -14** | **50.6** | **1.31 e -13** | **52.6** | **1.79 e -11** | **57.4** |

Values in boldface show the best performance of the algorithms as shown in the tables

**Table 10** Median IGD values for the DTLZ3 benchmark problem (bold items show the best performance). (*M* = Number of objective functions)

| Problem | DTLZ3 (M = 5) | | DTLZ3 (M = 8) | | DTLZ3 (M = 12) | | DTLZ3 (M = 16) | |
|---|---|---|---|---|---|---|---|---|
| | IGD | Time (second) | IGD | Time (second) | IGD | Time (second) | IGD | Time (second) |
| RPD-NSGA-II | 1.39 e -3 | 52.5 | 3.30 e -3 | 47.5 | 1.83 e -3 | 60.0 | 2.67 e -3 | 105.9 |
| MOEA/DD | 3.13 e -4 | 295.8 | 7.02 e -4 | 488.3 | 2.48 e -3 | 681.9 | 1.62 e -3 | 1752.6 |
| MOEA/D-DE | 1.17 e -2 | 59.9 | 4.05 e -2 | 63.3 | 1.92 e -3 | 67.1 | 2.80 e -3 | 85.4 |
| NSGA-III | 2.10 e -3 | 34.8 | 5.14 e -3 | 36.8 | 1.00 e -2 | 55.7 | 2.16 e -2 | 61.7 |
| Proposed method | **9.54 e -7** | **35.0** | **5.92 e -7** | **33.1** | **8.98 e -6** | **52.6** | **1.31 e -7** | **53.4** |

Values in boldface show the best performance of the algorithms as shown in the tables

**Table 11** Median IGD values for the DTLZ4 benchmark problem (bold items show the best performance). (M = Number of objective functions)

| Problem | DTLZ4 (M = 5) | | DTLZ4 (M = 8) | | DTLZ4 (M = 12) | | DTLZ4 (M = 16) | |
|---|---|---|---|---|---|---|---|---|
| | IGD | Time (second) | IGD | Time (second) | IGD | Time (second) | IGD | Time (second) |
| RPD-NSGA-II | 1.01 e -3 | 46.3 | 3.47 e -3 | 55.5 | 3.66 e -3 | 63.5 | 2.23 e -3 | 81.3 |
| MOEA/DD | 3.81 e -5 | 419.8 | 1.29 e -4 | 617.4 | 2.08 e -4 | 705.4 | 7.81 e -4 | 2483 |
| MOEA/D-DE | 4.17 e -3 | 89.1 | 7.31 e -2 | 92.6 | 6.98 e -3 | 94.2 | 3.66 e -3 | 116.0 |
| NSGA-III | 2.01 e -3 | 39.6 | 4.32 e -3 | 40.5 | 7.07 e -3 | 41.4 | 8.72 e -3 | 63.1 |
| Proposed method | **3.56 e -14** | **32.3** | **2.69 e -11** | **50.3** | **4.23 e -11** | **46.3** | **8.34 e -10** | **51.0** |

Values in boldface show the best performance of the algorithms as shown in the tables

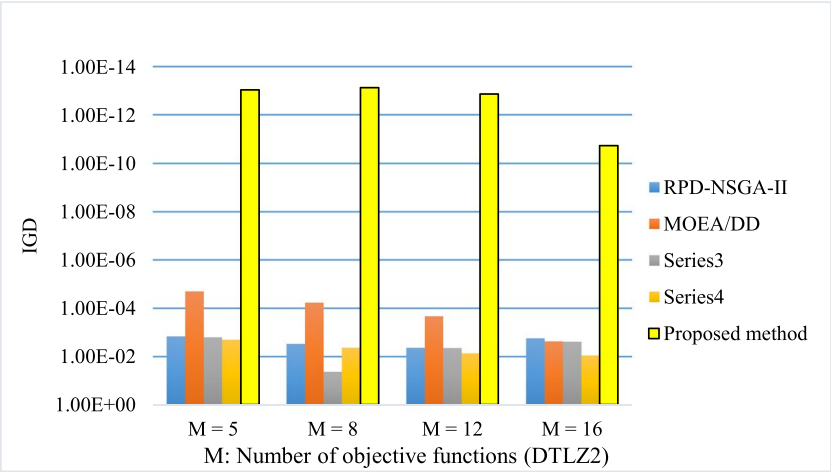



**Fig. 9** Comparison of results of the proposed algorithm with other algorithms for DTLZ2 benchmark problem

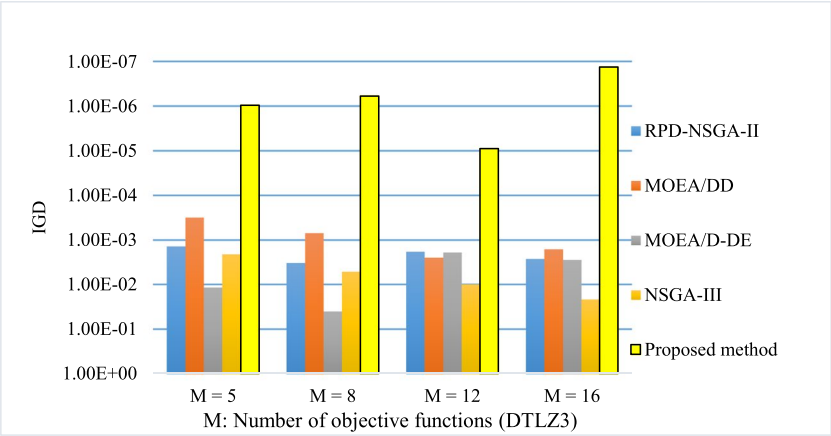



**Fig. 10** Comparison of results of the proposed algorithm and other algorithms for the DTLZ3 benchmark problem

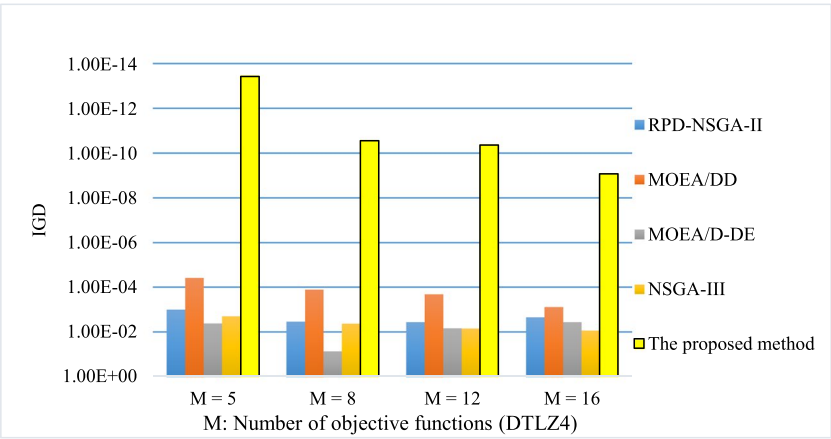



**Fig. 11** Comparison of the results of the proposed algorithm and other algorithms for DTLZ4 benchmark problem

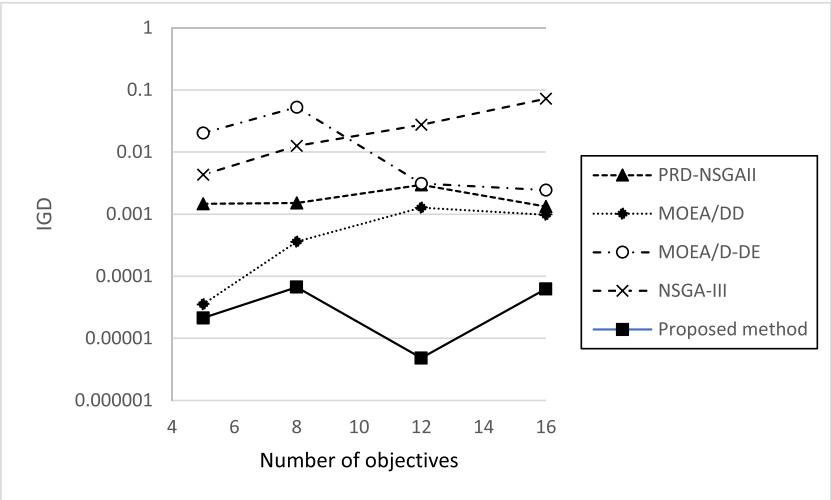



**Fig. 12** Mean values for the IGD metric of DTLZ1 problem

example, spacing metric [66], pure diversity (PD) [67], generational distance (GD) [66], coverage of two sets [68] individual generational distance (IGD) [69], hyper volume (HV) [70]. In this paper, spacing, several function evaluations (NFE), GD, and IGD metric functions are used to measure the performance of the proposed algorithm. IGD metric evaluates the convergence of the algorithm to the Pareto-optimal front. In addition, when the set of targeted Pareto-optimal points is defined with a suitable distribution, the IGD metric examines the

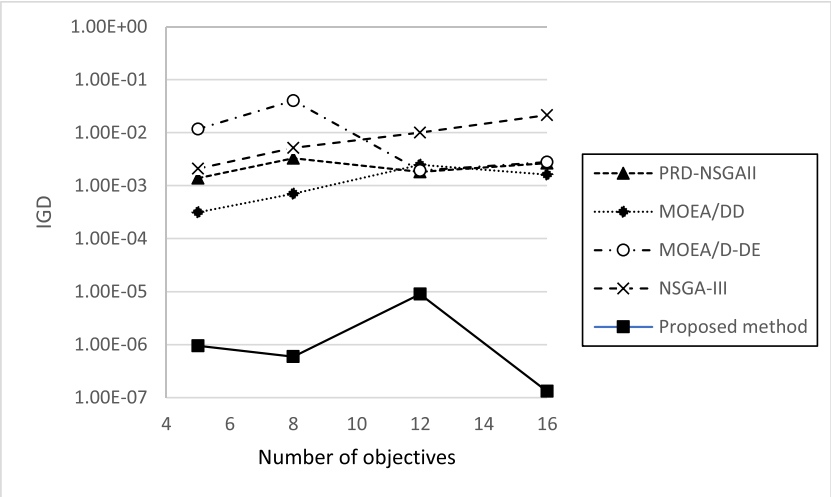



**Fig. 13** Mean values for the IGD metric of the DTLZ3 problem

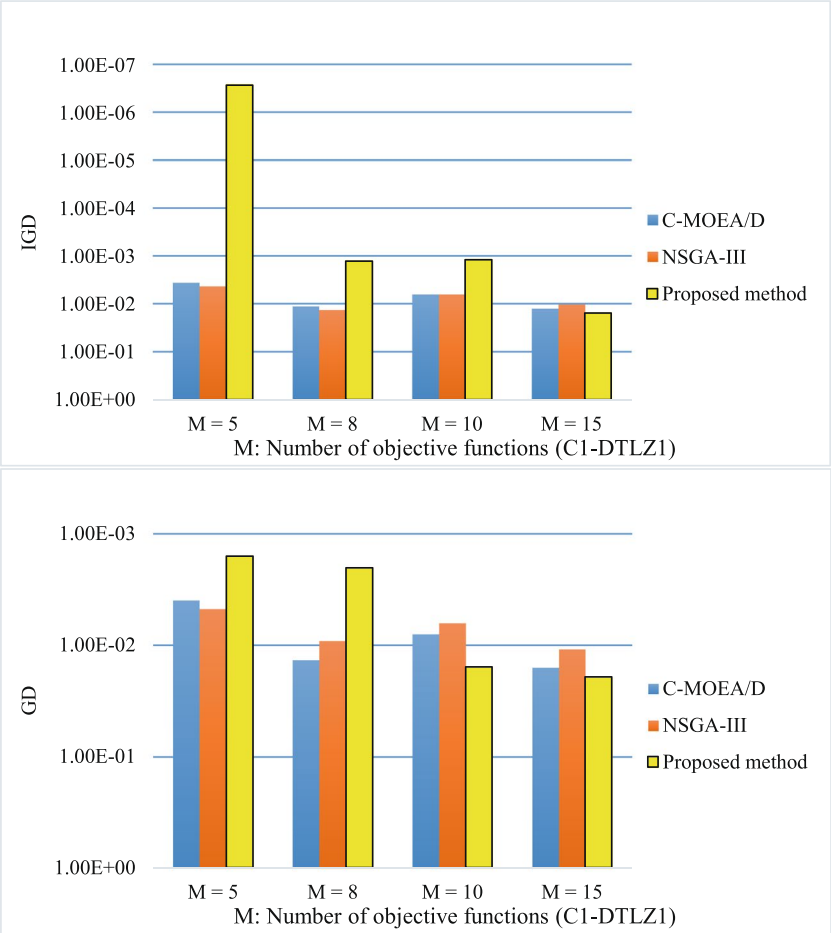



**Fig. 14** Comparison of results of the proposed method and other algorithms for the C1-DTLZ1 benchmark problem

goodness of distribution of obtained solutions for the entire Pareto front. Another performance metric used to assess many-objective optimization algorithms is the GD metric, which measures how far the obtained solutions are away from the true Pareto front, which can reflect the convergence of the algorithm. For unconstrained test problems, the solutions are close to the reference directions. Hence, the value of the GD is equal to the IGD. Therefore, GD values are presented for the constrained problems.

### 3.3 Validation

This section presents the validation and results of the proposed algorithm with other many-objective optimization algorithms using several challenging many-objective optimization benchmark problems. The obtained solutions for DTLZ1—DTLZ4 benchmark problems, with 3 objective functions, using the proposed algorithm and exact solutions, are shown in Fig. 7. The results of the proposed algorithm are shown with blue circles, and the exact solutions are characterized with red stars (Fig. 7). As can be seen from Fig. 8, the results of the proposed algorithm and those of exact solutions are in close agreement.

In addition, the IGD values that are calculated from the results of the MOPSO, and the proposed methods are compared with the recent many-objective optimization algorithms, as reported in I-DBEA [57]. The results of this comparative study using three and five-objective functions, DTLZ1-DTLZ2 problems are shown in Table 4. It can be seen from Table 4, that MOPSO has performed a global search, and the obtained results are relatively close to the exact solution; however, the IGD performance metric obtained from the hybrid MOPSO-SQP (after applying SQP in the second step of the hybrid algorithm), show a considerable consistency of the obtained results with the exact solutions.

The effect of the proposed mutation operator in the MOPSO algorithm on the spacing and IGD performance metrics using a many-objective optimization benchmark problem C1-DTLZ1, compared to other many-objective optimization algorithms, is shown in Table 5. In Table 5, the best IGD and spacing performance metrics of the proposed and other algorithms shown in bold characters. As it can be seen from Table 5, the IBEA with 20,000 function evaluation has a slightly better IGD value, and EMOPSO with 5000 function evaluation spacing performance metric value,

**Table 12** Median IGD and GD values for C1-DTLZ1 benchmark problem (bold items show the best performance). (M = Number of objective functions)

| Problem | Max.Gen/ Pop.Size | Method | IGD | GD | Simulation Time (second) |
|---|---|---|---|---|---|
| C1-DTLZ1 (M = 5) | 600/210 | C-MOEA/D | 3.64 e -3 | 3.96 e -3 | 136.5 |
| | | NSGA-III | 4.35 e -3 | 4.73 e -3 | 277.9 |
| | | Proposed method | 2.62 e -7 | 1.6 e -3 | 61.2 |
| C1-DTLZ1 (M = 8) | 800/156 | C-MOEA/D | 1.14 e -2 | 1.36 e -2 | 145.1 |
| | | NSGA-III | 1.36 e -2 | 9.15 e -3 | 205.9 |
| | | Proposed method | 1.26 e -3 | 2.04 e -3 | 44.7 |
| C1-DTLZ1 (M = 10) | 1000/275 | C-MOEA/D | 6.41 e -3 | 7.96 e -3 | 343.8 |
| | | NSGA-III | 6.36 e -3 | 6.36 e -3 | 226.1 |
| | | Proposed method | 1.18 e -3 | 1.58 e -2 | 82.6 |
| C1-DTLZ1 (M = 15) | 1500/135 | C-MOEA/D | 1.26 e -2 | 1.59 e -2 | 278.0 |
| | | NSGA-III | 1.04 e -2 | 1.09 e -2 | 232.8 |
| | | Proposed method | 1.54 e -2 | 1.94 e -2 | 131.2 |

**Table 13** Median IGD and GD values for the C2-DTLZ2 benchmark problem (bold items show the best performance). (*M* = Number of objective functions)

| Problem | Max.Gen/ Pop.Size | Method | IGD | GD | Simulation Time (second) |
|---|---|---|---|---|---|
| C2-DTLZ2 (M = 5) | 350/210 | C-MOEA/D | 7.30 e -4 | 1.68 e -1 | 92.9 |
| | | NSGA-III | 3.87 e -3 | 1.98 e -1 | 29.4 |
| | | **Proposed method** | **1.65 e -3** | **7.34 e -5** | 39.5 |
| C2-DTLZ2 (M = 8) | 500/156 | C-MOEA/D | 4.97 e -3 | 2.60 e -1 | 105.4 |
| | | NSGA-III | 2.35 e -2 | 4.73 e -1 | 40.8 |
| | | **Proposed method** | **2.68 e -3** | **6.1 e -5** | 52.4 |
| C2-DTLZ2 (M = 10) | 750/275 | C-MOEA/D | **1.49 e -3** | 1.91 e -1 | 287.9 |
| | | NSGA-III | 2.69 e -2 | 4.72 e -1 | 85.4 |
| | | **Proposed method** | 7.44 e -3 | **3.45 e -4** | 71.7 |
| C2-DTLZ2 (M = 15) | 1000/135 | C-MOEA/D | **2.13 e -2** | 3.83 e -1 | 168.1 |
| | | NSGA-III | 3.54 e -2 | 3.84 e -1 | 91.4 |
| | | **Proposed method** | 1.96 e -1 | **3.02 e -4** | 110.7 |

Values in boldface show the best performance of the algorithms as shown in the tables

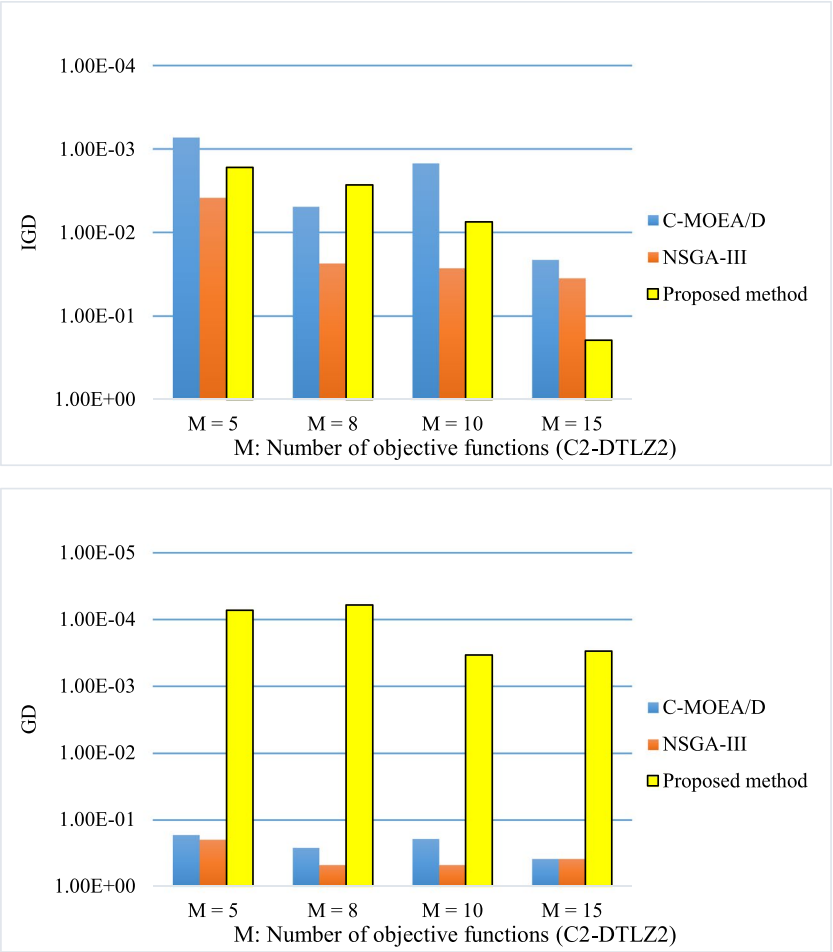



**Fig. 15** Comparison of results of the proposed method and other algorithms for the C2-DTLZ2 benchmark problem

receptively, but in all other cases, the proposed method performs better than other algorithms used in this case study.

### 3.4 Effect of the neighboring point technique in the proposed algorithm

To illustrate the performance of the proposed neighboring point technique, the proposed algorithm with and without adopting the neighboring point technique is implemented on the DTLZ1 benchmark problem, and the results are shown in Table 6. As it can be seen from Table 6, the proposed method with the neighboring point technique performs better than without the neighboring point technique.

#### 3.4.1 Examining the performance of the proposed method using unconstrained benchmark problems

This section presents a comparative study of the proposed method with other many-objective optimization algorithms using unconstrained many-objective optimization benchmark problems as described in Table 2. The results of the proposed method are compared with the results of RPD-NSGA-II and MOEA/DD, MOEA/D-DE [71], and NSGA-III, as shown in Tables 7, 8, 9, 10 and 11 and Figs. 9, 10, 11, 12, 13, 14.

In this comparative study, both NFE and IGD performance metrics are used to evaluate the performance of the above algorithms, as shown in Tables 7–11 and Figs. 9–14. MOEA/DD (Table 7) performs slightly better than other algorithms when NFE is equal to 50,000, but the proposed method gives better accuracy for NFE of 75,000 and 1,000,000. For NFE = 75,000 and NFE = 100,000, the results of the proposed method have a high degree of accuracy than other algorithms such as; RPD-NSGA-II [33], MOEA/DD [72], MOEA/D-DE [73], NSGA-III [38], GrEA [74], KnEA [74], tDEA [75], SPEA-III [76], and MOSA [77].

As described above, among the different performance metrics for comparing the results of many-objective optimization algorithms, the IGD performance metric was selected because it reflects both convergence and divergence of results.

In addition, the proposed method was applied to the DTLZ1, DTLZ2, DTLZ3, and DTLZ4 many-objective optimization benchmark problems with a number of function

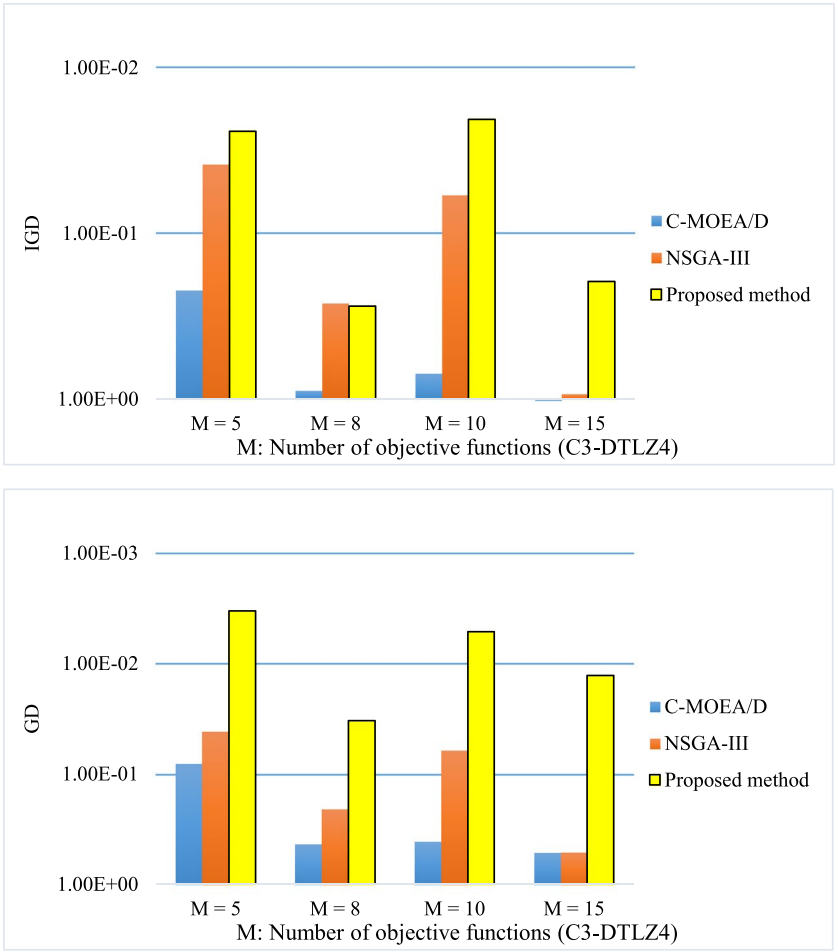



**Fig. 16** Comparison of results of the proposed method and other algorithms for the C3-DTLZ4 benchmark problem

**Table 14** Median IGD and GD values for the C3-DTLZ4 benchmark problem (bold items show the best performance). ($M$ = Number of objective functions)

| Problem | Max.Gen/ Pop.Size | Method | IGD | GD | Simulation Time (second) |
|---|---|---|---|---|---|
| C3-DTLZ4 (M = 5) | 1250/210 | C-MOEA/D | 2.21 e -1 | 8.01 e -2 | 364.0 |
| | | NSGA-III | 3.86 e -2 | 4.09 e -2 | 98.4 |
| | | **Proposed method** | **2.42 e -2** | **3.36 e -3** | 159.9 |
| C3-DTLZ4 (M = 8) | 2000/156 | C-MOEA/D | 8.92 e -1 | 4.28 e -1 | 378.7 |
| | | NSGA-III | **2.65 e -1** | 2.07 e -1 | 129.5 |
| | | **Proposed method** | 2.75 e -1 | **3.30 e -2** | 250.0 |
| C3-DTLZ4 (M = 10) | 3000/275 | C-MOEA/D | 7.01 e -1 | 4.06 e -1 | 1307.8 |
| | | NSGA-III | 5.93 e -2 | 6.06 e -2 | 307.4 |
| | | **Proposed method** | **2.05 e -2** | **5.18 e -3** | 302.8 |
| C3-DTLZ4 (M = 15) | 4000/135 | C-MOEA/D | 1.45 e 0 | 5.12 e -1 | 698.6 |
| | | NSGA-III | 9.32 e -1 | 5.08 e -1 | 339.9 |
| | | **Proposed method** | **1.95 e -1** | **1.29 e -2** | 385.9 |

Values in boldface show the best performance of the algorithms as shown in the tables

evaluations of 100,000, and the results of the proposed method and other algorithms are shown in Table 8–11 and Figs. 9–14. Table 8 shows the results of the proposed and other algorithms using the DTLZ1 benchmark problem and the IGD performance metric values. The IGD accuracy is in the order of $10^{-5}$ to $10^{-6}$. In the DTLZ2 benchmark problem, $K = 10$, and hence the number of decision variables for each objective function is more than that in the case of DTLZ1.

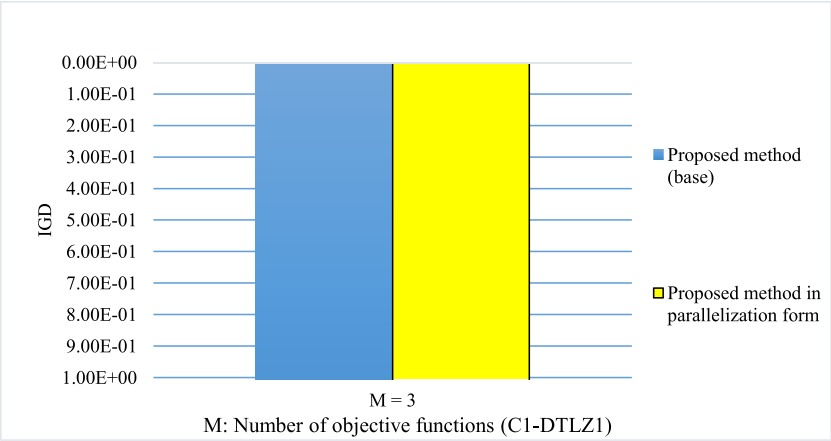



**Fig. 17** Comparison of results with, and without parallelization of the proposed method

**Table 15** Comparison of the proposed method with and without parallelization of the median IGD values for the C1-DTLZ1 benchmark problem (bold items show better performance). ($M$ = Number of objective functions). (NFE = Number of function evaluations)

| Problem | Method | IGD | NFE |
|---|---|---|---|
| C1-DTLZ1 (M = 3) | The proposed method (without parallelization) | 1.5156e0 | 477,360 |
| | The proposed method with parallelization | **1.1043e0** | **5000** |

Values in boldface show the best performance of the algorithms as shown in the tables

But it defines unimodal problems, and because of this characteristic, more accuracy is achievable (with the order of $10^{-11}$).

The benchmark many-objective optimization problem DTLZ3 is more challenging because this benchmark problem has more than $30^9$ local Pareto optimal fronts [73], which is considered the most challenging many-objective optimization bench mark problem in the DTLZ family in terms of the difficulties associated with the convergence. This characteristic of the benchmark problem necessitates using a multi-start strategy in the global search stage of the proposed method.

Figures 9-14 show the IGD performance metrics for DTLZ, DTLZ2, DTLZ3, and DTLZ4 many-objective benchmark problems. These figures show that the proposed method outperforms all other algorithms in terms of convergence to the optimal Pareto frontier in all cases used in the above comparative study.

### 3.4.2 Examining the performance of the proposed method using constrained benchmark problems

This section presents the computational performance of the proposed method. In this respect, the proposed method was

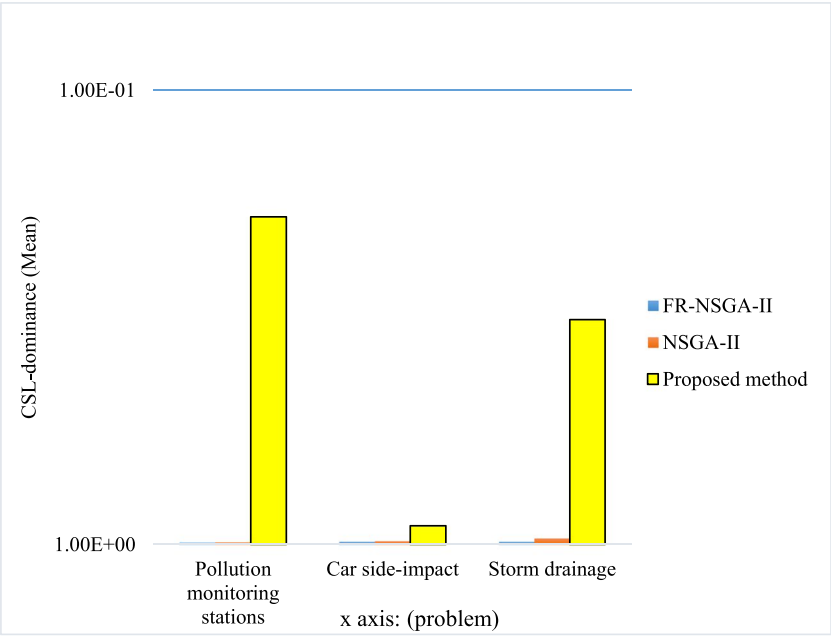



**Fig. 18** Comparison of results of the proposed method, FR-NSGA-II, and NSGA-II algorithms using $CS_{L\text{-dominance}}$ metric

**Table 16** Real-world test problems characteristics

| Real-world test problem | Number of decision variables | Number of objective functions | Number of constraints |
|---|---|---|---|
| Location problem of pollution monitoring stations [78] | 2 | 5 | - |
| Car side-impact problem [79] | 7 | 11 | 10 |
| Storm drainage [80] | 3 | 5 | 7 |

**Table 17** Comparison of the proposed method, FR-NSGA-II, and NSGA-II algorithms using $CS_{L\text{-dominance}}$ metric (bold items show the best performance)

| Problem | Algorithm | $CS_{L\text{-dominance}}$ | |
|---|---|---|---|
| | | Mean | Std |
| Pollution monitoring stations | **Proposed method** | **0.9887** | 0.0048 |
| | FR-NSGA-II | 0.9871 | 0.0315 |
| | NSGA-II | 0.1900 | 0.3010 |
| Car side-impact | **Proposed method** | **0.9839** | 0.0031 |
| | FR-NSGA-II | 0.981 | 0.0094 |
| | NSGA-II | 0.9090 | 0.1908 |
| Storm drainage | **Proposed method** | **0.9846** | 0.0081 |
| | FR-NSGA-II | 0.9673 | 0.0153 |
| | NSGA-II | 0.32 | 0.1443 |

Values in boldface show the best performance of the algorithms as shown in the tables

applied to C1-DTLZ1 and C2-DTLZ2, constrained many-objective optimization benchmark problems. The results of the proposed method and other many-objective optimization algorithms for the above benchmark problems are shown in Tables 12 and 13. These results (Tables 12 and 13), show that for the benchmark problem C1-DTLZ1, with 15 objective functions, the NSGA-III has a better GD performance metric (Table 13 and Fig. 15). The C-MOEA/D algorithm for the benchmark problem C2-DTLZ2, with 15 and 10 objective functions has a better IGD performance metric. However, the proposed method in all other cases outperforms all other algorithms used in this study, both in terms of IGD, and GD performance metrics, as shown in Tables 12 and 13 and Figs. 15 and 16.

The results of the proposed methods and other algorithms for C3-DTLZ4, many-objective optimization benchmark

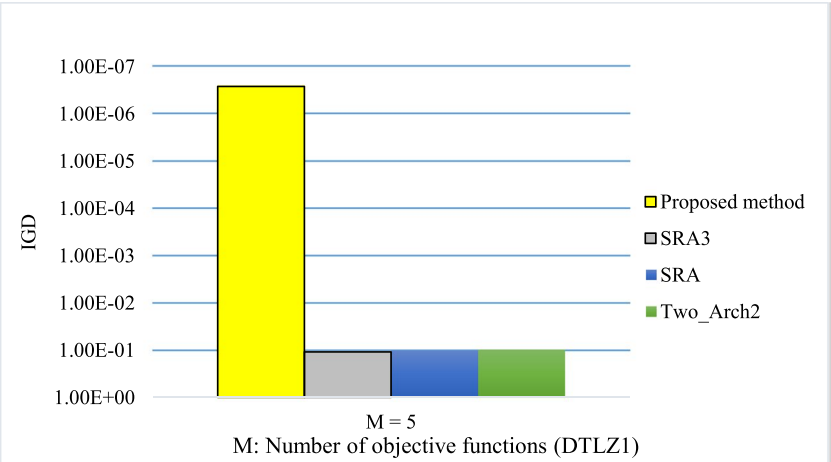



**Fig. 19** Comparison of results of the proposed and SEA3, SRA and Two_Arch2 method with five objective functions using DTLZ1 benchmark problem

**Table 18** Median IGD values for the DTLZ1 benchmark problem (bold items show the best performance). (M = Number of objective functions)

| Problem (Number of objective functions) | Number of function evaluations (NFE) | 90,000 |
|---|---|---|
| | Method | IGD |
| DTLZ1 (M = 5) | SRA3 | $1.09\times10^{-1}$ |
| | SRA | $1.02\times10^{-1}$ |
| | Two_Arch2 | $9.64\times10^{-2}$ |
| | **Proposed method** | **2.14 e -5** |

**Table 19** Median IGD values for the DTLZ2 benchmark problem (bold items show the best performance). (M = Number of objective functions)

| Problem (Number of objective functions) | Number of function evaluations (NFE) | 90,000 |
|---|---|---|
| | Method | IGD |
| DTLZ2 (M = 5) | SRA3 | $1.80\times10^{-1}$ |
| | SRA | $1.76\times10^{-1}$ |
| | Two_Arch2 | $1.52\times10^{-1}$ |
| | **Proposed method** | **8.7 e -14** |

Values in boldface show the best performance of the algorithms as shown in the tables

**Table 20** Median IGD values for the DTLZ3 benchmark problem (bold items show the best performance). (M = Number of objective functions)

| Problem (Number of objective functions) | Number of function evaluations (NFE) | 90,000 |
|---|---|---|
| | Method | IGD |
| DTLZ3 (M = 5) | SRA3 | $1.78 \times 10^{-1}$ |
| | SRA | $2.01 \times 10^{-1}$ |
| | Two_Arch2 | $1.89 \times 10^{-1}$ |
| | **Proposed method** | **9.54 e -7** |

Values in boldface show the best performance of the algorithms as shown in the tables

**Table 21** Median IGD values for the DTLZ4 benchmark problem (bold items show the best performance). (M = Number of objective functions

| Problem (Number of objective functions) | Number of function evaluations (NFE) | 90,000 |
|---|---|---|
| | Method | IGD |
| DTLZ4 (M = 5) | SRA3 | $1.79 \times 10^{-1}$ |
| | SRA | $1.78 \times 10^{-1}$ |
| | Two_Arch2 | $1.53 \times 10^{-1}$ |
| | Proposed method | **3.56 e -14** |

Values in boldface show the best performance of the algorithms as shown in the tables

problem is shown in Table 14 and Fig. 17. These results show that the proposed method significantly outperforms other many-objective optimization algorithms in all cases with 5,8, 10 and15, and objective functions in terms of both IGD and GD performance metrics used in this study (Table 14 and Fig. 17).

As described earlier in this paper, a parallelization scheme is adopted in the proposed method, in which optimization of all reference directions (weight vectors) is performed simultaneously. Table 15 shows a comparison between the parallelization and without the parallelization of the proposed method on a benchmark problem C1-DTLZ1, with three objective functions. The obtained results (Table 15) show that the parallelization of the optimization has significantly enhanced the performance of the proposed method in terms of the NFE and, at the same time, has a comparable IGD performance metric concerning the case without parallelization of the proposed method as shown in Fig. 18.

### 3.4.3 Investigating the performance of the proposed method using real-world problems

This section presents applications of the proposed method to real-world many-objective optimization problems involving 5 and 11 objective functions. The computational performance of the proposed method compared with the other algorithms, in particular with one of the recent many-objective optimization algorithms (i.e., FR-NSGA-II [56]). The number of decision variables, objective functions, and constraints of the test problems is shown in Table 16. In the test problems of car side-impact and storm drainage problems, a penalty method was used for handling the constraints in MOPSO. In this respect, each test problem has been executed 10 times, and following this, average and standard deviations have been calculated to represent the computational performance of the algorithms used in this study.

Because true Pareto fronts are not known for real-world problems, the IGD metric is not suitable for comparing the performance of the algorithms. Therefore, the $CS_{L\text{-dominance}}$ metric, which is a modified version of two set coverage (CS) metrics, was used [56].

The results of the $CS_{L\text{-dominance}}$ metric for comparison of the proposed method and other algorithms using the above real-world test cases are given in Table 17 and Fig. 19. In all cases, the number of function evaluations is fixed to be 100,000. According to these results, the proposed method performs better than the FR-NSGA-II

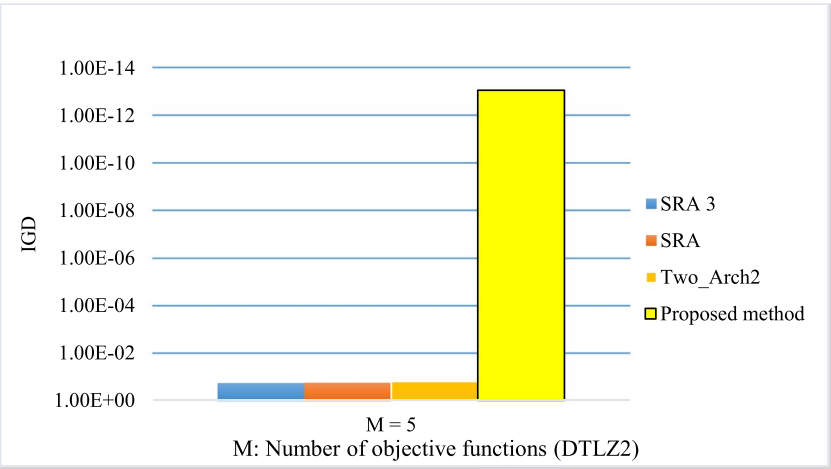



**Fig. 20** Comparison of results of the proposed and SEA3, SRA and Two_Arch2 methods with five objective functions using DTLZ2 benchmark problem

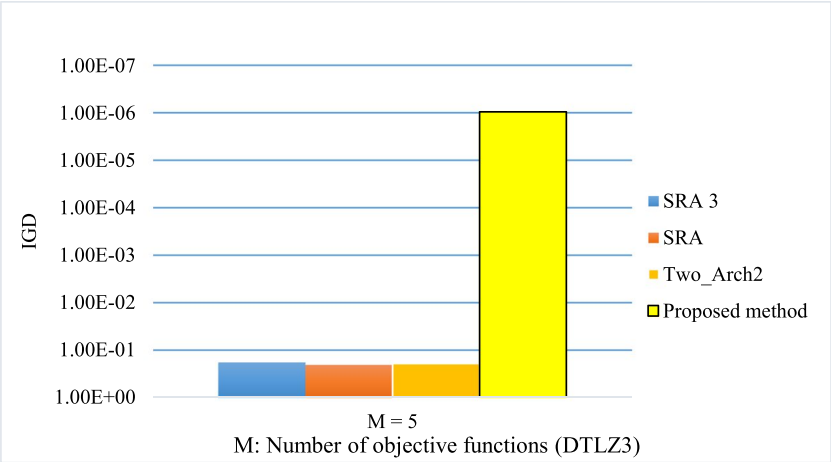



**Fig. 21** Comparison of results of the proposed and SEA3, SRA and Two_Arch2 methods with five objective functions using DTLZ3 benchmark problem

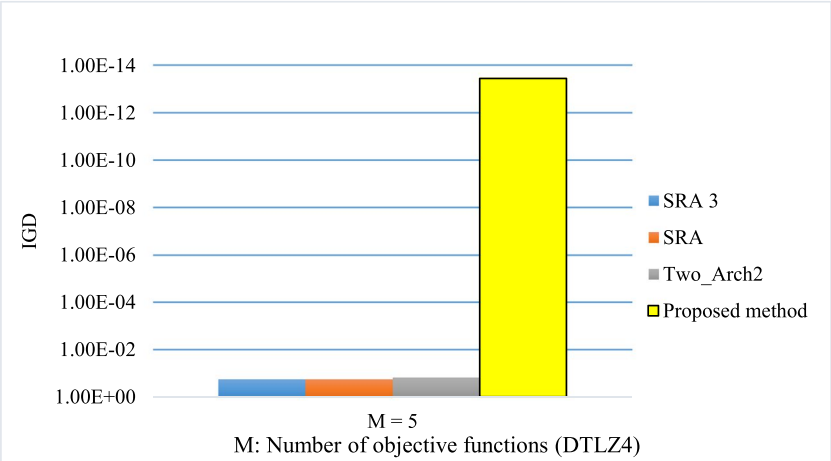



**Fig. 22** Comparison of results of the proposed and SEA3, SRA and Two_Arch2 methods with five objective functions using DTLZ4 benchmark problem

and NSGA-II in terms of dominance for solving these real-world many-objective optimization problems. For calculating $CS_{L\text{-dominance}}$, the two last sets of Pareto solutions, which are obtained using the SQP algorithm of the proposed method, were compared based on the CS definition. As seen from Table 18, for the first test problem, CSL-dominance, on average, is equal to 0.9887 for the proposed method, compared to the 0.9871 and 0.1900 values using FR-NSGA-II and NSGA-II, respectively. Similarly, with 11 objective car side-impact problems, the CSL-dominance obtained from the proposed method on average is 0.9839, compared to 0.981 for the FR-NSGA-II. Finally, for storm drainage application, the proposed algorithm reaches $CS_{L\text{-dominance}} = 0.9846$ on average, while this metric for FR-NSGA-II and NSGA-II when implemented are; 0.9673 and 0.32, respectively. The results of this experimentation of the proposed method's performance are shown in Fig. 19, in which the proposed method significantly outperforms the other algorithms used in this experimentation.

#### 3.4.4 Comparative study of the performance of the proposed and other methods

In addition to comparative studies presented in the previous sections of this paper, in this section further performance of proposed many-objective optimization algorithms with SRA3, SRA, two_Arch2 [81], using DTLZ benchmark problems (Table 1), are compared. As described in the previous comparative studies, the results of this comparative study (Tables 18, 19, 20 and 21 and Fig. 19, 20, 21 and 22), also show that the proposed method outperforms SRA3, SRA and Two_Ach2.

## 4 Conclusions

This paper presented a novel decomposition-based hybrid many-objective optimization algorithm for solving many-objective optimization problems. Although several many-objective optimization algorithms have been developed; however, many-objective optimization algorithms present

new challenges in terms of the computational costs, convergence, and diversity of solutions. To address these research challenges, this paper focused on developing computationally efficient with a high degree of accuracy for solving many-objective optimization problems. The proposed method combines the global search capability of the PSO algorithm with the fast and accurate local search capability of the SQP for solving many-objective optimization problems. In addition, a new mutation operator introduced to enhance the performance of the MOPSO. The main feature of the proposed mutation operator is considering particles that violate the side constraint and returning them to the optimization process by replacing their position with a feasible location. In addition, a multi-start strategy utilized in the proposed method to overcome the stochastic nature of this evolutionary algorithm and to support the proposed method to reach more repeatable solutions.

The accuracy and computational performance of the proposed method were evaluated and compared with other many-objective optimization algorithms using several challenging many-objective optimization benchmarks and real-world test problems. The results of these experimental studies are shown in Tables 4–17 and Fig. 7–19. The results show that the proposed method outperforms other many-objective optimization algorithms in all cases used in this paper.

**Acknowledgements** The authors are grateful to Mr. Ehsan Abbasali for the assistance given in the preparation of the second revision of the article.

**Data availability** Data will be made available on reasonable request.

## References

1. Rouhi M, Ghayoor H, Hoa SV, Hojjati M (2015) Multi-objective design optimization of variable stiffness composite cylinders. Compos Part B Eng 69:249–255
2. Kalantari M, Dong C, Davies IJ (2016) Multi-objective analysis for optimal and robust design of unidirectional glass/carbon fibre reinforced hybrid epoxy composites under flexural loading. Compos Part B Eng 84:130–139
3. Farmani MR, Roshanian J, Babaie M, Zadeh PM (2012) Multi-objective collaborative multidisciplinary design optimization using particle swarm techniques and fuzzy decision making. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 226(9):2281–2295. https://doi.org/10.1177/0954406211432981
4. Reddy MJ, Nagesh Kumar D (2007) Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation. Hydrol. Process. 21(21):2897–2909
5. Coello CAC (2006) Evolutionary multi-objective optimization: a historical view of the field. IEEE Comput Intell Mag 1(1):28–36
6. Deb K, Sindhya K, Hakanen J (2016) Multi-objective optimization. In Decision Sciences: Theory and Practice, CRC Press, pp. 145–184. https://doi.org/10.1201/9781315183176-4
7. Maoguo G, Licheng J, Dongdong Y, Wenping M (2009) Research on evolutionary multi-objective optimization algorithms. J Softw 20(2). https://doi.org/10.3724/SP.J.1001.2009.03483
8. Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. Struct Multidiscip Optim 26(6):369–395
9. Salazar-Lechuga M, Rowe JE (2005) Particle swarm optimization and fitness sharing to solve multi-objective optimization problems, in Congress on Evolutionary Computation (CEC'2005) pp. 1204–1211
10. Coello CAC, Lamont GB, Van Veldhuizen DA et al (2007). Evolutionary algorithms for solving multi-objective problems, vol 5. Springer Nature
11. Deb K (2001) Multi-objective optimization using evolutionary algorithms, vol 16. John Wiley & Sons Ltd, New York
12. Li M, Yang S, Liu X (2014) Diversity comparison of Pareto front approximations in many-objective optimization. IEEE Trans Cybern 44(12):2568–2584
13. Jain H, Deb K (2014) An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach. IEEE Trans Evol Comput 18(4):602–622
14. Chen S (2012) Particle swarm optimization with pbest crossover, in 2012 IEEE Congress on Evolutionary Computation, pp. 1–6
15. Fu G-Z, Li Y-F, Tao Y, Huang H-Z (2018) An interactive preference-based evolutionary algorithm for multi-criteria satisficing optimization. J Intell Fuzzy Syst 34(4):2503–2511
16. Sülflow A, Drechsler N, Drechsler R (2007) Robust multi-objective optimization in high dimensional spaces, in International conference on evolutionary multi-criterion optimization, pp. 715–726
17. Sato H, Aguirre HE, Tanaka K (2007) Controlling dominance area of solutions and its impact on the performance of MOEAs, in International conference on evolutionary multi-criterion optimization pp. 5–20
18. Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search, in International Conference on Parallel Problem Solving from Nature, pp. 832–842
19. Wang Y, Emmerich M, Deutz A, Bäck T (2019) Diversity-indicator based multi-objective evolutionary algorithm: DI-MOEA, in International Conference on Evolutionary Multi-Criterion Optimization, pp. 346–358
20. Sun Y, Yen GG, Yi Z (2018) IGD indicator-based evolutionary algorithm for many-objective optimization problems. IEEE Trans Evol Comput 23(2):173–187. https://doi.org/10.1109/TEVC.2018.2791283
21. Hughes EJ (2008) Fitness assignment methods for many-objective problems. Multiobjective Problem Solving from Nature. Springer, pp 307–329
22. Hegazy AE, Makhlouf MA, El-Tawel GS (2018) Dimensionality reduction using an improved whale optimization algorithm for data classification. Int J Mod Educ Comput Sci 10(7):37
23. Grishagin V, Israfilov R, Sergeyev Y (2018) Convergence conditions and numerical comparison of global optimization methods based on dimensionality reduction schemes. Appl Math Comput 318:270–280
24. Van Der Maaten L, Postma E, den Herik J (2009) Dimensionality reduction: a comparative. J Mach Learn Res 10(66–71):13
25. Gebken B, Peitz S, Dellnitz M (2019) On the hierarchical structure of pareto critical sets. J Glob Optim 73(4):891–913
26. Laparra V, Malo J, Camps-Valls G (2015) Dimensionality reduction via regression in hyperspectral imagery. IEEE J Sel Top Signal Process 9(6):1026–1036
27. Han E-H, Karypis G, Kumar V, Mobasher B (1997) Clustering in a high-dimensional space using hypergraph models. Technical Report, Department of Computer Science, University of Minnesota, Minneapolis, MN.

28. Cai L, Qu S, Cheng G (2018) Two-archive method for aggregation-based many-objective optimization. Inf Sci (NY) 422:305–317
29. Cai X, Sun H, Zhang Q, Huang Y (2018) A grid weighted sum pareto local search for combinatorial multi and many-objective optimization. IEEE Trans Cybern 49(9):3586–3598
30. Pescador-Rojas M, Gómez RH, Montero E, Rojas-Morales N, Riff M-C, Coello CAC (2017) An overview of weighted and unconstrained scalarizing functions, in International Conference on Evolutionary Multi-Criterion Optimization, pp. 499–513
31. von Lücken C, Barán B, Brizuela C (2014) A survey on multi-objective evolutionary algorithms for many-objective problems. Comput Optim Appl 58(3):707–756
32. Habib A, Singh HK, Chugh T, Ray T, Miettinen K (2019) A multiple surrogate assisted decomposition based evolutionary algorithm for expensive multi/many-objective optimization. IEEE Trans Evol Comput 23(6). https://doi.org/10.1109/TEVC.2019.2899030
33. Elarbi M, Bechikh S, Gupta A, Ben Said L, Ong Y-S (2018) A new decomposition-based NSGA-II for many-objective optimization. IEEE Trans Syst Man Cybern Syst 48(7):1191–1210. https://doi.org/10.1109/TSMC.2017.2654301
34. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731
35. Chand S, Wagner M (2015) Evolutionary many-objective optimization: a quick-start guide. Surv Oper Res Manag Sci 20(2):35–42
36. Liang Z, Hu K, Ma X, Zhu Z (2019) A many-objective evolutionary algorithm based on a two-round selection strategy. IEEE Trans Cybern
37. Liu H-L, Gu F, Zhang Q (2014) Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. IEEE Trans Evol Comput 18(3):450–455. https://doi.org/10.1109/tevc.2013.2281533
38. Deb K, Jain H (2014) 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):577–601
39. Yi J-H et al (2018) Behavior of crossover operators in NSGA-III for large-scale optimization problems. Inf Sci (NY)
40. He Z, Yen GG, Zhang J (2013) Fuzzy-based Pareto optimality for many-objective evolutionary algorithms. IEEE Trans Evol Comput 18(2):269–285
41. Wang H, Yao X (2013) Corner sort for Pareto-based many-objective optimization. IEEE Trans Cybern 44(1):92–102
42. Zou J et al (2019) An adaptation reference-point-based multi-objective evolutionary algorithm. Inf Sci (NY) 488:41–57
43. Zhu C, Xu L, Goodman ED (2015) Generalization of Pareto-optimality for many-objective evolutionary optimization. IEEE Trans Evol Comput 20(2):299–315
44. Li M, Yang S, Liu X (2013) Shift-based density estimation for Pareto-based algorithms in many-objective optimization. IEEE Trans Evol Comput 18(3):348–365
45. Liu H-L, Chen L, Zhang Q, Deb K (2017) Adaptively allocating search effort in challenging many-objective optimization problems. IEEE Trans Evol Comput 22(3):433–448
46. Li K, Deb K, Zhang Q, Kwong S (2014) An evolutionary many-objective optimization algorithm based on dominance and decomposition. IEEE Trans Evol Comput 19(5):694–716
47. Ge H, Zaho M, Sun M, Wang Z, Tan G, Zhang Q, Philip Chen CL (2018) A many-objective evolutionary algorithm with two interacting processes: Cascade clustering and reference point incremental learning. IEEE Trans Evol Comput 23(4). https://doi.org/10.1109/TEVC.2018.2874465
48. Sharma D, Kumar A, Deb K, Sindhya K (2007) 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
49. Ibrahim A, Martin MV, Rahnamayan S, Deb K (2017) Fusion-based hybrid many-objective optimization algorithm, in 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 2372–2381
50. Zhang B, Shafi K, Abbass HA (2016) Hybrid knowledge-based evolutionary many-objective optimization, in 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 1007–1014
51. Sobieszczanski-Sobieski J, Morris A, Van Tooren M (2015) Multidisciplinary design optimization supported by knowledge based engineering. John Wiley & Sons
52. Su Y, Wang J, Ma L, Wang X, Lin Q, Chen J (2018) A novel many-objective optimization algorithm based on the hybrid angle-encouragement decomposition, in International Conference on Intelligent Computing, pp. 47–53
53. Xiang Y, Zhou Y, Li M, Chen Z (2016) A vector angle-based evolutionary algorithm for unconstrained many-objective optimization. IEEE Trans Evol Comput 21(1):131–152
54. Su Y et al (2019) A hybridized angle-encouragement-based decomposition approach for many-objective optimization problems. Appl Soft Comput 78:355–372
55. Dennis J, Das I (1998) Normal-boundary intersection: a new method for generating Pareto optimal points in nonlinear multicriteria optimization problems. SIAM J Optim 8(3):631–657
56. Mohammadi S, Monfared MAS, Bashiri M (2017) An improved evolutionary algorithm for handling many-objective optimization problems. Appl Soft Comput 52:1239–1252
57. Asafuddoula M, Ray T, Sarker R (2015) A decomposition-based evolutionary algorithm for many objective optimization. IEEE Trans Evol Comput 19(3):445–460
58. Singh HK, Bhattacharjee KS, Ray T, Mostaghim S (2018) Investigation of a Simple Distance Based Ranking Metric for Decomposition-Based Multi/Many-Objective Evolutionary Algorithms, in Australasian Joint Conference on Artificial Intelligence, pp. 384–396
59. Wickramasinghe UK, Carrese R, Li X (2010) Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm, in Evolutionary Computation (CEC), 2010 IEEE Congress on, pp. 1–8
60. Ahmed KN, Razak TA (2016) Fast and effective spatial clustering using multi-start particle swarm optimization technique. Int J Eng Technol 8(2):1229–1237
61. Asafuddoula M, Singh HK, Ray T (2018) An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors. IEEE Trans Cybern 48(8):2321–2334
62. Fakoor M, Zadeh PM, Eskandari HM (2017) Developing an optimal layout design of a satellite system by considering natural frequency and attitude control constraints. Aerosp Sci Technol 71:172–188
63. Zadeh PM, Sokhansefat T, Kasaeian AB, Kowsary F, Akbarzadeh A (2015) Hybrid optimization algorithm for thermal analysis in a solar parabolic trough collector based on nanofluid. Energy 82:857–864
64. Deb K, Padhye N (2014) Enhancing performance of particle swarm optimization through an algorithmic link with genetic algorithms. Comput Optim Appl 57(3):761–794
65. Luo C, Shimoyama K, Obayashi S (2015) A study on many-objective optimization using the kriging-surrogate-based evolutionary algorithm maximizing expected hypervolume improvement. Math Prob Eng 2015(4):1–15. https://doi.org/10.1155/2015/162712
66. Coello CAC, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 8(3):256–279

67. Wang H, Jin Y, Yao X (2016) Diversity assessment in many-objective optimization. IEEE Trans Cybern 47(6):1510–1522
68. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—a comparative case study, in International Conference on Parallel Problem Solving from Nature, pp. 292–301
69. Zhou A, Jin Y, Zhang Q, Sendhoff B, Tsang E (2006) Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion, in 2006 IEEE international conference on evolutionary computation, pp. 892–899
70. While L, Hingston P, Barone L, Huband S (2006) A faster algorithm for calculating hypervolume. IEEE Trans Evol Comput 10(1):29–38
71. Li H, Zhang Q (2009) Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Trans Evol Comput 13(2):284–302
72. Li K, Deb K, Zhang Q, Kwong S (2015) Combining dominance and decomposition in evolutionary many-objective optimization. IEEE Trans Evol Comput 19(5):694–716
73. Tušar T, Filipič B (2007) Differential evolution versus genetic algorithms in multiobjective optimization, in International Conference on Evolutionary Multi-Criterion Optimization, pp. 257–271
74. Yang S, Li M, Liu X, Zheng J (2013) A grid-based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 17(5):721–736
75. Yuan Y, Xu H, Wang B, Yao X (2015) A new dominance relation-based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 20(1):16–37
76. Cao P, Fan Z, Gao R, Tang J (2017) A manufacturing oriented single point search hyper-heuristic scheme for multi-objective optimization, in International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. 58134, p. V02BT03A031
77. Gu Q, Chen S, Jiang S, Xiong N (2021) Improved strength Pareto evolutionary algorithm based on reference direction and coordinated selection strategy. Int J Intell Syst 36(9):4693–4722
78. Miettinen K, Lotov AV, Kamenev GK, Berezkin VE (2003) Integration of two multiobjective optimization methods for nonlinear problems. Optim Methods Softw 18(1):63–80
79. Sinha A, Saxena DK, Deb K, Tiwari A (2013) Using objective reduction and interactive procedure to handle many-objective optimization problems. Appl Soft Comput 13(1):415–427
80. Saxena DK, Deb K (2008) Dimensionality reduction of objectives and constraints in multi-objective optimization problems: A system design perspective, in Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on, pp. 3204–3211
81. Wang Z, Yao X (2022) An efficient multi-indicator and many-objective optimization algorithm based on two-archive. Neural Evol Comput. https://doi.org/10.48550/arXiv.2201.05435