

Research Article

A Reference Point-Based Evolutionary Algorithm Solves Multi and Many-Objective Optimization Problems: Method and Validation

Mohammed Jameel ¹ and Mohamed Abouhawwash ^{2,3}

¹Department of Mathematics, Sana'a University, Sana'a 13509, Yemen

²Department of Mathematics, Faculty of Science, Mansoura University, Mansoura 35516, Egypt

³Department of Computational Mathematics Science and Engineering (CMSE), Michigan State University, East Lansing, MI 48824, USA

Correspondence should be addressed to Mohammed Jameel; mohjameel555@gmail.com

Received 8 August 2022; Revised 30 November 2022; Accepted 2 December 2022; Published 25 January 2023

Academic Editor: Rodolfo E. Haber

Copyright © 2023 Mohammed Jameel and Mohamed Abouhawwash. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The integration of a decision maker's preferences in evolutionary multi-objective optimization (EMO) has been a common research scope over the last decade. In the published literature, several preference-based evolutionary approaches have been proposed. The reference point-based non-dominated sorting genetic (R-NSGA-II) algorithm represents one of the well-known preference-based evolutionary approaches. This method mainly aims to find a set of the Pareto-optimal solutions in the region of interest (ROI) rather than obtaining the entire Pareto-optimal set. This approach uses Euclidean distance as a metric to calculate the distance between each candidate solution and the reference point. However, this metric may not produce desired solutions because the final minimal Euclidean distance value is unknown. Thus, determining whether the true Pareto-optimal solution is achieved at the end of optimization run becomes difficult. In this study, R-NSGA-II method is modified using the recently proposed simplified Karush-Kuhn-Tucker proximity measure (S-KKTPM) metric instead of the Euclidean distance metric, where S-KKTPM-based distance measure can predict the convergence behavior of a point from the Pareto-optimal front without prior knowledge of the optimum solution. Experimental results show that the algorithm proposed herein is highly competitive compared with several state-of-the-art preference-based EMO methods. Extensive experiments were conducted with 2 to 10 objectives on various standard problems. Results show the effectiveness of our algorithm in obtaining the preferred solutions in the ROI and its ability to control the size of each preferred region separately at the same time.

1. Introduction

Most real-world optimization problems usually contain two or more conflicting objective functions. These objective functions must be optimized simultaneously. This type of problem is known as a multi-objective optimization problem (MOP).

In MOPs with contradictory objectives, a single solution that can be considered the best does not always exist. Instead, a set of solutions represents the best compromises among the different objectives. This set, which belongs to the search space, is known as the Pareto set (or efficient set), whereas its images, which belong to the objective space, are known as Pareto front (PF) [1, 2]. Several evolutionary multi-objective

optimization (EMO) algorithms, such as NSGA-II [3], SPEA2 [4], and MOEA/D [5], have been suggested in the past two decades or more. Classical EMO mainly aims to obtain a set of well-converged and well-distributed non-dominated solutions that approach the entire PF [6, 7]. Researchers have devoted their effort to developing algorithms in recent years [8–14].

The proportion of non-dominant solutions rises as the number of objectives increases, which is one of the fundamental problems with all EMO approaches. Due to the insufficient selection pressure caused by a high percentage of non-dominant solutions, the EMO approach cannot advance in finding the optimal spots. Incorporating the decision maker's preferences into the algorithm is a practical

way to deal with this issue. A new kind of ranking mechanism [9] can be used to make selection pressure stronger and steer the optimization approach to search in a specified region.

In real life, the DM is always focused on some specified subsets of the obtained solutions. The techniques of preference-based MCDM aim to find a part of the PF, whereas EMO algorithms aim to obtain a well-distributed set of points close to the whole PF. We call the part of the optimal solutions that is near to or lies on the PF a region of interest (ROI) [15]. Solutions within the ROI satisfy the DM's need. However, this scenario does not mean that any efficient solutions outside the ROI are not the optimal solutions to the problem. The preference information given by the DM in the EMO can enable a highly efficient search. Many different approaches to preference information given by DMs exist, such as reference point (RP), preference angle, and reference weights. One of the most utilized approaches in preference-based EMO algorithms is the RP. As mentioned above, EMO tries to find well-distributed multiple efficient solutions across the whole PF, as displayed in Figure 1(a). Also, this figure illustrates the feasible objective region and the unfeasible objective region. On the contrary, preference-based EMO algorithms concentrate on a certain part of the true PF based on a preference point (reference point) determined by DM. The non-dominated points cluster near the RP, as shown in Figure 1(b).

The following is a typical classification of methods based on preferences, depending on how they are expressed by the DM [16–18]: (i) a priori methods, where preferences are expressed before calculating PO solutions, for example, through a utility function [19] or by an RP [20]; (ii) a posteriori methods, in which the DM chooses the solution of her/his preference after a set of efficient solutions has been calculated (for example, [21, 22]); (iii) interactive methods, where the DM guides the search with a utility function, and this function may change during the optimization process because of the new information acquired (for example, [23, 24]); and finally, (iv) methods not based on preferences, where additional information on preferences is not available, and the idea is to find a balance between the objectives [25].

Over the past two decades, researchers have focused their attention on preference-based EMO approaches. These approaches have been actively developed, and they mainly focus on specific parts of the PF. Depending on the preference information supplied by the DM, these algorithms seek to find an ROI that is close to/on the true PF.

Numerous algorithms of preference-based EMO have been introduced. Deb and Sundar [26] suggested the RP-based NSGA-II (R-NSGA-II), which focuses on obtaining a preferred ROI during the evolutionary process. By including the RP's location information in the Pareto dominance, Molina et al. [27] initiated a concept of Pareto dominance termed *g*-dominance. Ben Said et al. [28] presented a novel variant of the Pareto-dominant relationship, called *r*-dominance, with which we can obtain good convergence to the PF. Ruiz et al. [29] proposed WASF-GA, another variant of the preference-based MOEA algorithm. Yu et al. [30] suggested a different representative preference-based

decomposition MOEA by decomposing the preference information into several scalar optimization problems. Recently, new R-NSGA-II modified methods have been proposed to assist DMs in convergent to Pareto-dominance compliant solutions in a specific area of interest [31–33].

Although many preference-based algorithms use various metrics to select preferred solutions, some of these metrics require prior knowledge of the PF while others require specific parameters [34, 35]. S-KKTPM does not require prior knowledge of the PF or any parameters.

Herein, we introduce a novel preference-based NSGA-II algorithm. The Euclidean distance was utilized in the original R-NSGA-II study as a metric between two trade-off solutions. In our study, we use the simplified Karush–Kuhn–Tucker proximity metric (S-KKTPM) instead of the Euclidean distance metric. S-KKTPM can anticipate the convergence behavior of a point from PF without prior knowledge of the optimum solution [36, 37]. The Karush–Kuhn–Tucker (KKT) conditions occupy a significant role in optimization theory. KKT proximity measure was proposed through these conditions. Incorporating S-KKTPM within the R-NSGA-II provides theoretical convergence properties for the final preferred points. The main contributions of the introduced algorithm are listed below:

- (i) We introduce a new RP-based algorithm called RS-KKTPM, based on the S-KKPM metric, by integrating S-KKPM with NSGA-II to obtain the PO solutions in ROI.
- (ii) Obtaining different ranges of ROI in a single run.
- (iii) Adding flexibility for several RPs at the same time.
- (iv) Obtaining excellent performance when the RP is located in different regions.
- (v) Obtaining a good balance between convergence and diversity aspects around the RP.
- (vi) Solving different shapes of PF (e.g., convex, concave, and discontinuous) with a different number of objective functions (up to 10 objectives).
- (vii) Making the results competitive compared with those of the other preference mechanisms on many-objective problems.

The layout of this work is as follows. Section 2 reviews some fundamental definitions. An overview of the works relevant to this paper is mentioned in Section 3. In Section 4, the R-NSGA-II algorithm is combined with the S-KKTPM metric. In the following section, the obtained experiments and the results are discussed and described. Section 6 summarizes the paper's achievements and presents some upcoming works. Table 1 displays the nomenclature/abbreviations used in this study.

2. Basic Definitions

An MOP contains a set of n decision variables, M objective functions, J constraints of inequality, and P constraints of equality. MOP can be defined as follows [28].

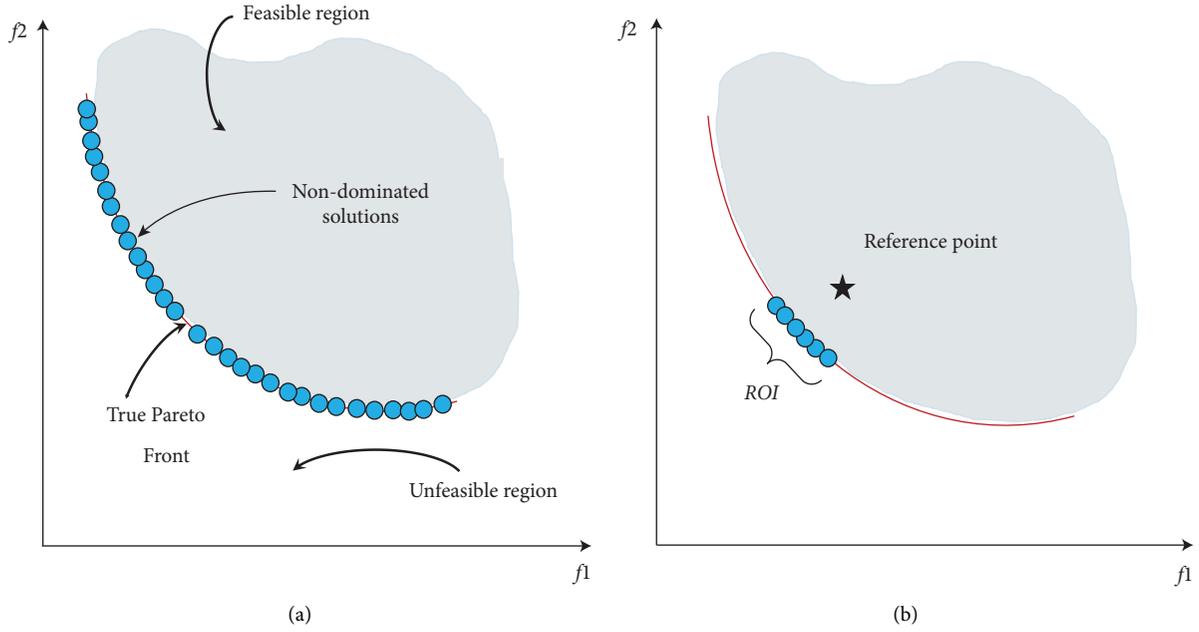


FIGURE 1: Objective space: (a) evenly distributed across the entire PF; (b) crusting near an RP.

$$\begin{aligned} & \text{Minimize}_{(\mathbf{x})} f(\mathbf{x}) = ((f_1)(\mathbf{x}), \dots, f_M(\mathbf{x}))^T \\ & \text{subject to } g_j(\mathbf{x}) \leq 0, j = 1, 2, \dots, J, \\ & \quad h_i(\mathbf{x}) = 0, i = 1, 2, \dots, P, \end{aligned} \quad (1)$$

where $x = (x_1, \dots, x_n)^T$ is an n -dimensional decision variable vector, $f_k: \mathbb{R}^n \rightarrow R (k = 1, 2, \dots, M)$ are the objective functions, and g_j and $h_i: \mathbb{R}^n \rightarrow R (j = 1, 2, \dots, J; i = 1, 2, \dots, P)$ are the constraints of the problem.

In an MOP with contradictory objectives, the search space is only partially ordered, and two solutions may be indifferent to each other. A single decision variable simultaneously optimizing all the objectives is unusual. Consequently, for MOPs, the \langle, \rangle , and $=$ operators are extended as follows.

Definition 1 (Pareto dominance relation). Given two solutions $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, \mathbf{x} is said to dominate \mathbf{y} in the Pareto sense (denoted by $f(\mathbf{x}) \prec f(\mathbf{y})$) if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \forall i \in \{1, \dots, M\}$ and $\exists j \in \{1, \dots, M\}$ where $f_j(\mathbf{x}) < f_j(\mathbf{y})$.

Definition 2 (non-dominated solution). A solution $\mathbf{x} \in \Omega \subseteq \mathbb{R}^n$ (Ω is the feasible space) is said to be non-dominated if and only if there does not exist another solution $\mathbf{y} \in \Omega$ such that $f(\mathbf{x}) \prec f(\mathbf{y})$.

Definition 3 (Pareto-optimal (PO)). A solution $\mathbf{x} \in \Omega$ is said to PO if $f(\mathbf{x})$ is non-dominated with respect to Ω .

The set of solutions in the search space is called the Pareto solution set (PS). In contrast, the set of all non-

dominated vectors in the objective space corresponding to the PS is called the PF [38].

Definition 4 (PS and PF). The PS is defined as follows:

$$PS = \{\mathbf{x} \in \Omega \mid \mathbf{x} \text{ is PO}\}. \quad (2)$$

The corresponding PF is defined as follows:

$$PF = \{\mathbf{u} = f(\mathbf{x}) \in \mathbb{R}^M \mid \mathbf{x} \in PS\}, (\mathbb{R}^M \text{ is the objective space}). \quad (3)$$

Definition 5 (RP). An RP $f(\mathbf{x}_{RP})$ is defined in objective space, where $f(\mathbf{x}_{RP})$ is provided by the DM.

Definition 6 (ROI). The ROI is the projection of the set of preferred efficient solutions in the objective space, i.e., $ROI = \{f(\mathbf{x}) \mid \|f(\mathbf{x}) - f(\mathbf{x}_{RP})\| < \delta, \mathbf{x} \in PS\}$, where $f(\mathbf{x}_{RP})$ is the closest to the RP $f(\mathbf{x}_{RP})$. δ denotes the radius of the ROI, which is determined by DM.

3. Related Works

In Section 3.1, KKT optimality conditions are briefly reviewed. Section 3.2 presents R-NSGA-II in detail.

3.1. KKT Conditions. KKT conditions play an important role in optimization theory. Through these conditions, it is possible to know if the solution produced by the EMO algorithm is the PO solution or not. For the MOP with inequality constraints, KKT conditions are defined as follows [39]:

TABLE 1: List of nomenclature and abbreviations.

Nomenclature	
n	Number of decision variables
M	Number of objective functions
J	Number of inequality constraints
P	Number of equality constraints
$<$	Pareto-dominance relation
Ω	Feasible search space
δ	Radius of the ROI
\mathbb{R}^M	Objective space
N	Population size
\mathbb{R}^n	Decision variable space
β_i	Lagrange multiplier of i th objective function
γ_j	Lagrange multiplier if j th inequality constraint
ξ_k	KKTPM value
$\tilde{\xi}_k$	S-KKTPM value
P_c	Crossover probability
P_m	Mutation probability
ϵ	Size of the preferred region
Abbreviations	
BNH	Constrained bi-objective test problem by Binh and Korn
DM	Decision maker
DTLZ	Deb–Thiele–Laumanns–Zitzler test problem set
EMO	Evolutionary multi-objective optimization
g-NSGA-II	g-Dominance relation-based non-dominated sorting genetic algorithm-II
KKT	Karush–Kuhn–Tucker
KKTPM	KKT proximity metric
S-KKTPM	Simplified KKT proximity metric
MCDM	Multi-criteria decision making
MOP	Multi-objective optimization problem
MOEA/D	Multi-objective evolutionary algorithm based on decomposition
MOEA/D-PRE	MOEA/D based on preference information
NSGA-II	Non-dominated sorting genetic algorithm-II
NSGA-III	Non-dominated sorting genetic algorithm-III
OSY	Constrained bi-objective test problem by Osyczka and Kundu
PF	Pareto front
PO	Pareto-optimal
PS	Pareto set
R-IGD	R-metric based on inverse generational distance
R-HV	R-metric based on hypervolume
r-NSGA-II	r-Dominance relation-based NSGA-II
R-NSGA-II	Reference point-based NSGA-II
R-NSGA-III	Reference point-based NSGA-III
ROI	Region of interest
RP	Reference point
RS-KKTPM	Reference point based on S-KKTPM
SBX	Simulated binary crossover
SRN	A constrained bi-objective test problem by Srinivas and Deb
SPEA2	Improved version of strength Pareto evolutionary algorithm
TNK	A constrained bi-objective test problem by Tanaka
WV-MOEA-P	Weight vector-based multi-objective optimization algorithm with preference
ZDT	Zitzler–Deb–Thiele test problem set

$$\sum_{i=1}^M \beta_i \nabla f_i(\mathbf{x}) + \sum_{j=1}^J \gamma_j \nabla g_j(\mathbf{x}) = \mathbf{0}, \quad (4) \quad \gamma_j \geq 0, j = 1, 2, \dots, J, \quad (7)$$

$$g_j(\mathbf{x}) \leq 0, j = 1, 2, \dots, J, \quad (5) \quad \beta_i \geq 0, i = 1, 2, \dots, M, \quad (8)$$

$$\gamma_j g_j(\mathbf{x}) = 0, j = 1, 2, \dots, J, \quad (6) \quad \beta \neq \mathbf{0}.$$

The parameters β_i and γ_j are called the Lagrange multipliers for the i th objective function and j th inequality

constraint, respectively. Any solution \mathbf{x} that satisfies each the above conditions is called a KKT point. Equations (4) and (6) are called the equilibrium and complimentary slackness equations, respectively. The conditions stated in equation (5) ensure feasibility for \mathbf{x} while the conditions stated in equation (7) ensure that the parameters γ_j are non-negative. The conditions stated in equation (8) also ensure that the parameters β_i are non-negative, but at least one of them must be non-zero. In the following section, we briefly discuss R-NSGA-II algorithm.

3.2. R-NSGA-II Algorithm. As mentioned in Section 1, classical EMO algorithms mainly aim to develop a finite number of random solutions into a set of non-dominant solutions that converge and distribute across the entire PF over several generations. On the contrary, preference-based algorithms aim to produce non-dominated solutions centered around the desired part (s) of the PF based on the preference information supplied by the DM. This information can be given in several techniques: RPs, aspiration levels, weights, and reference direction [2]. RPs are one of the most used techniques in preference-based EMO algorithms. Usually, an RP is said to be achievable if it lies in the feasible objective space; otherwise, it is said to be unachievable.

In 2006, Deb and Sundar [26] put forward R-NSGA-II method, which presents the DM's preferences as one or more RPs. The method is based on the benchmark manner, which is based on preference information [40]. It is a modification of the widely used EMO approach NSGA-II, in which an Euclidean distance metric is applied instead of the crowding distance metric from the RP that indicates DM's preference. The primary notion behind R-NSGA-II is to give preference to parents who have short Euclidean distances to the RP. The following is the description of the R-NSGA-II procedure: P_t (of size N) is a randomly generated parent population. A new offspring population Q_t (of size N) is generated using the number of operations (binary tournament selection, recombination, and mutation). Thereafter, the populations P_t and Q_t are combined, and the resulting population $R_t = P_t + Q_t$ (size $2N$) is classified according to dominance in fronts. The new population P_{t+1} is built starting with the fronts with the lowest rank until reaching a front F_i , which cannot be accepted without making the size of population to exceed N . Next, the preference operator is applied to the front F_i to maintain the size of the new population. The final front F_i , which cannot be fully accepted, is then considered, and the remaining slots are filled according to an environmental selection approach. The Euclidean distance for each RP is calculated with respect to each solution of the front F_i . For each RP, the solution closest to the said point takes the preferred distance value of 1. The solutions that are closest to all of the RPs are given the shortest preferred distance. The preferred distance value of 2 is then applied to the solutions with the next smallest distance to each RP, and the process is repeated for the remaining F_i solutions. In the generation of the new population of descendants, the preferred solutions in the selection by the tournament are those with a lower value of preferred distance.

The idea of ϵ -based selection strategy is utilized to maintain diversity in the solutions close to each RP. First, a solution of the front F_i is randomly chosen. Next, the Euclidean distance in the objective space of all the solutions is computed with respect to the chosen solution. After that, the points that have a sum of the normalized difference in the objective search space values less than or equal to ϵ from the selected point are given an artificial large distance to remove them from the competition; in this method, only a solution within ϵ -neighborhood is relevant. The process continues by randomly choosing another solution different from the previous one, to which the concept of ϵ -based selection strategy described above is applied again.

3.2.1. Advantages and Disadvantages. Compared with classical RP-based algorithms, R-NSGA-II works well for high-dimensional MOPs; it is suitable for any frontier shape, several objectives, and variables. It also shows some advantages: the classical methods depend on the reference direction (weight vector); however, R-NSGA-II is independent of the weight vector. Moreover, the classical methods in most cases can only find efficient solutions for different RPs by applying the algorithm to each RP for several times, whereas R-NSGA-II can produce a set of efficient points for different RPs in a single simulation. RPs can exist anywhere in the objective space (achievable or unachievable). However, it requires a parameter ϵ to maintain a diversity of selected solutions near the RPs.

As mentioned above, the crowding distance metric of NSGA-II has been replaced using the Euclidean distance metric in R-NSGA-II to obtain the solutions closest to the RPs assigned by DMs. However, the final minimal Euclidean distance value is unknown. Thus, ascertaining whether the efficient solution is accomplished at the end of an optimization run is difficult. In other words, the Euclidean distance metric does not have any information about the proximity of a solution to the PF. Additionally, in the case of achievable RPs, the Euclidean distance metric may not be monotonically reduced to its minimum value. One major disadvantage of this method is that the DM cannot control the size of each preferred region separately. Furthermore, the DM cannot smoothly control the obtained PO solutions within each desired region. Below, we introduce a new approach that is based on integrating the S-KKTPM metric with the R-NSGA-II algorithm.

4. The Introduced R-NSGA-II with S-KKTPM

In Section 4.1, the development of the KKT-proximity measure is introduced. Section 4.2 presents the proposed RS-KKTPM in detail.

4.1. S-KKTPM. KKT conditions are necessary to know whether the solution obtained by the EMO algorithm is a KKT point. Hence, they play an important role in optimization theory [39, 41]. During the last decade, a KKT-proximity measure has been developed utilizing KKT optimality theory. In 2013, a KKT-based proximity metric

(KKTTPM) was suggested by Dutta et al. [42] to calculate a KKTTPM value for any iteration (or solution) \mathbf{x}^k for a single-objective optimization problem. Deb and Abouhawwash et al. [37, 43] extended the above KKTTPM for MOPs. Their expansion, which is based on the incorporation of the KKTTPM metric via scalarization approaches, aims to relate the convergence property of a solution from a specific

optimal solution. Other information on KKTTPM for MOPs can be found in [37, 43].

In 2021, Eichfelder and Warnow [44] proposed a new KKTTPM metric for MOPs without using any scalarization approach. The authors defined the following methodology for calculating the KKTTPM value for any solution $\mathbf{x}^k \in \mathbb{R}^n$, for the MOP mentioned in equation (1):

$$\begin{aligned}
& \text{minimize}_{(\xi_k, \beta, \gamma)} \xi_k, \\
& \text{subject to} \left\| \sum_{i=1}^M \beta_i \nabla f_i(\mathbf{x}^k) + \sum_{j=1}^J \gamma_j \nabla g_j(\mathbf{x}^k) \right\|^2 \leq \xi_k, \\
& \sum_{j=1}^J \gamma_j g_j(\mathbf{x}^k) \geq -\xi_k, \\
& g_j(\mathbf{x}^k) \leq \xi_k, \forall j, \\
& \sum_{i=1}^M \beta_i = 1, \\
& \beta_i \geq 0, \forall i \text{ and } \gamma_j \geq 0, \forall j,
\end{aligned} \tag{9}$$

where J and M , respectively, are the numbers of constraint and objective functions. The value ξ_k obtained after the optimization is the KKTTPM at the point \mathbf{x}^k . First-order derivatives of constraint and objective functions are necessary to solve this problem. KKTTPM metric can be utilized

to single, multi, and many-objective optimization problems. The above problem has $(M + J + 1)$ variables, $(M + 2J + 2)$ inequality constraints, and one equality constraint. To reduce the number of constraints in the optimization problem mentioned above, we propose to redefine it as follows:

$$\begin{aligned}
& \text{minimize}_{(\hat{\xi}_k, \hat{\beta}, \hat{\gamma})} \hat{\xi}_k, \\
& \text{subject to} \left\| \sum_{i=1}^M \hat{\beta}_i \nabla f_i(\mathbf{x}^k) + \sum_{j=1}^J \hat{\gamma}_j \nabla g_j(\mathbf{x}^k) \right\|^2 \leq \hat{\xi}_k, \\
& \sum_{j=1}^J \hat{\gamma}_j g_j(\mathbf{x}^k) \geq -\hat{\xi}_k, \\
& \sum_{i=1}^M \hat{\beta}_i = 1, \\
& \hat{\beta}_i \geq 0, \forall i \text{ and } \hat{\gamma}_j \geq 0, \forall j,
\end{aligned} \tag{10}$$

where the variable vector of the above optimization problem is $(\hat{\xi}_k, \hat{\beta}, \hat{\gamma})$. The value of $\hat{\xi}_k$, which solves the above problem, is referred to as the simplified KKTTPM (S-KKTTPM). The primary goal of reducing the number of constraints is to save the computational cost of an optimization problem. The above problem has $(M + J + 1)$ variables, $(M + J + 2)$ inequality constraints, and one equality constraint. The number J of inequality constraints has been reduced compared to the optimization problem mentioned in

equation (9) without affecting the optimization process. To ensure that values of both $\hat{\xi}_k$ and ξ_k obtained after the optimization are identical at point \mathbf{x}^k , first we consider the ZDT1 unconstrained problem with thirty variables [45]. We ran NSGA-II for 200 generations in this problem, with a population size of 40. Figures 2 and 3 illustrate the $\hat{\xi}_k$ and ξ_k values versus generation numbers for efficient solutions to the unconstrained ZDT1 problem. The minimum, 25th

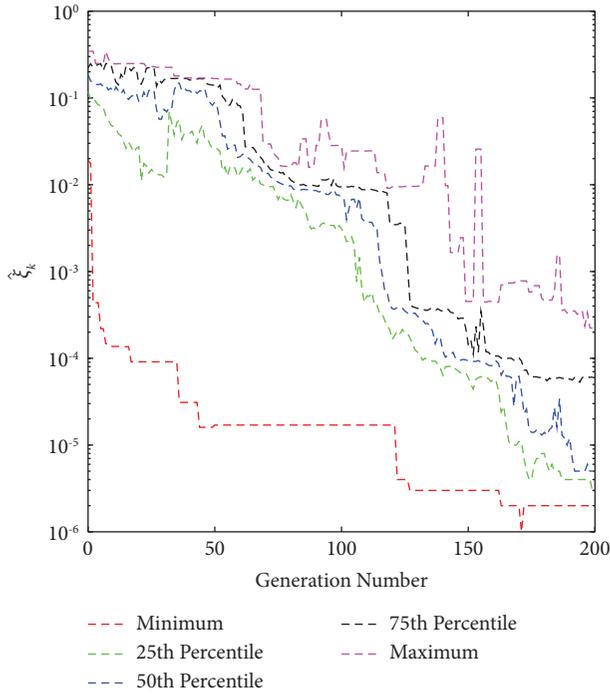


FIGURE 2: $\hat{\xi}_k$ values versus generations for unconstrained ZDT1 problem.

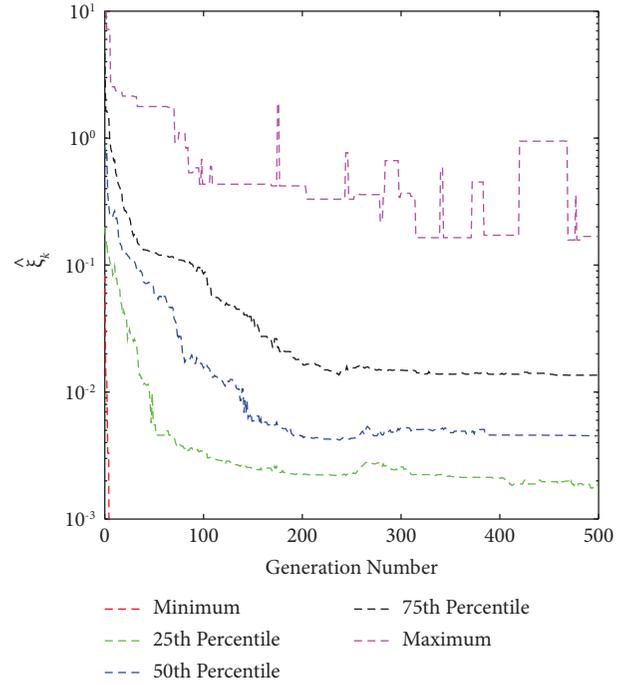


FIGURE 4: $\hat{\xi}_k$ values versus generations for constrained SRN problem.

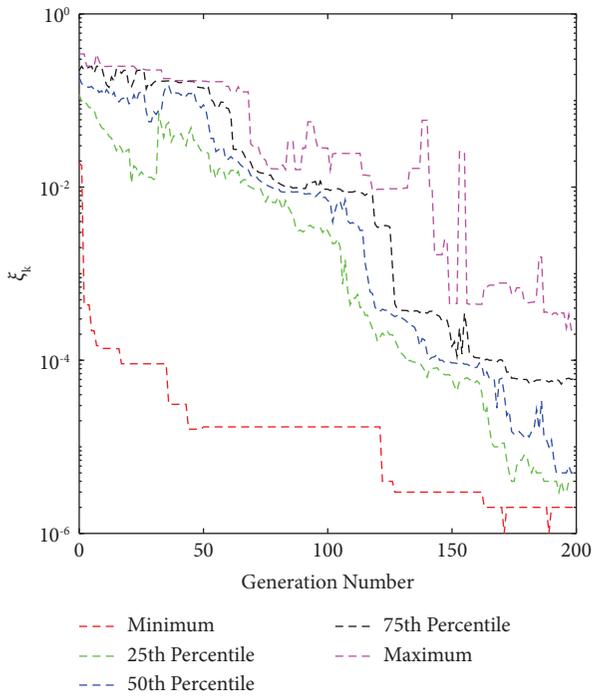


FIGURE 3: ξ_k values versus generations for unconstrained ZDT1 problem.

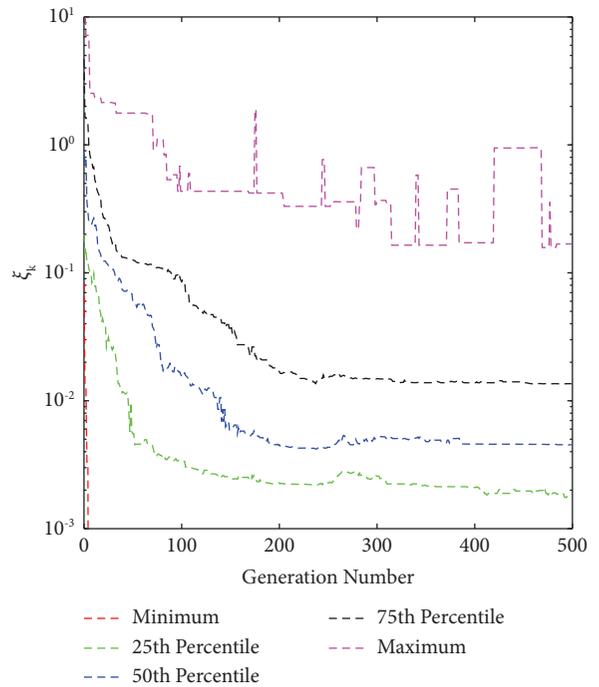


FIGURE 5: ξ_k values versus generations for constrained SRN problem.

percentile, 50th percentile, 75th percentile, and maximum $\hat{\xi}_k$ and ξ_k values are also plotted for all PO solutions at each generation. Both figures show a constant reduction as the generation number increases. With a correlation coefficient of 0.9996, both figures show that the values and patterns of

$\hat{\xi}_k$ and ξ_k are identical. Second, we consider the SRN unconstrained problem with two variables and two constraints [46]. In this problem, we also ran NSGA-II until generation 500, with a population size of 200. $\hat{\xi}_k$ and ξ_k values versus

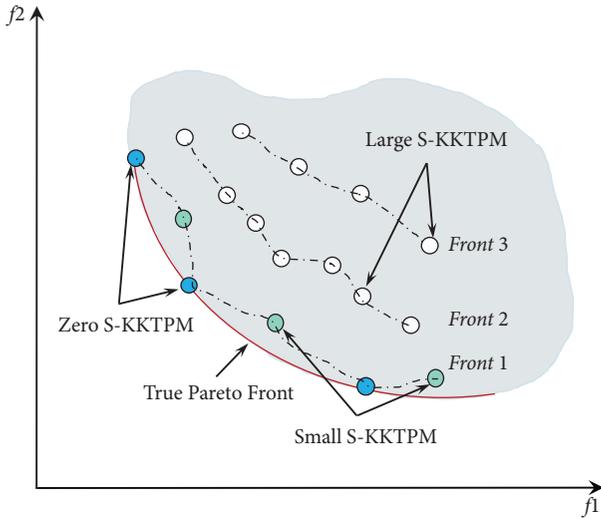


FIGURE 6: S-KKTPM values for a set of non-dominated points.

generations for obtained solutions are displayed in Figures 4 and 5. Both figures also show that the values and patterns of $\hat{\xi}_k$ and ξ_k are congruent, with a correlation coefficient of 0.9999.

An advantage of S-KKTPM metric given in equation (10) is that it predicts the convergence behavior of a point from the PF without prior knowledge of the PO solution. Now, we describe several features of the S-KKTPM [37, 43, 44]:

- (i) It can be utilized as a termination condition for the algorithm of optimization.
- (ii) It is applicable in high-dimensional MOPs; S-KKTPM is suitable for any frontier shape, large number of objectives, and variables.
- (iii) It provides a monotonous characteristic of the S-KKTPM surface over the objective space. S-KKTPM value decreases monotonously almost to zero as the iterate approaches the efficient solution. Figure 6 displays the S-KKTPM values for a set of efficient solutions located at different positions in objective space; for example, S-KKTPM value is zero in the true PO solutions (marked by blue circles), which lie on the PF. For efficient solutions, which are close to the PF (marked by green circles), S-KKTPM value is small. For far-away solutions from the PF (marked by white circles), S-KKTPM value is large.
- (iv) Calculating S-KKTPM value does not require any parameters, such as weight vector and ideal point, unlike when calculating values in other versions of KKT proximity measure.

In this study, we use S-KKTPM optimization problem to calculate ξ_k value at iterate \mathbf{x}^k . We used MATLAB `fmincon()` algorithm optimization to solve S-KKTPM optimization problem (see Algorithm 1).

4.2. The Proposed RS-KKTPM. To make R-NSGA-II solutions preferred and acceptable to DMs and to easily control the size of each region, S-KKTPM metric is integrated with the-NSGA-II algorithm.

In this study, we refer to the RP-based S-KKTPM as RS-KKTPM. The introduced algorithm allows DMs to apply any number of RPs. RS-KKTPM also allows the DMs to control the size of the preferred parts separately. In the introduced algorithm, we replace the Euclidean distance metric, utilized in R-NSGA-II, with S-KKTPM metric. Solutions with small S-KKTPM values are chosen in the introduced method. The preference operator is utilized in this algorithm to select a subset of solutions from the final front that cannot be accommodated totally to maintain the size of population in the novel population. Instead of using the preference distance as in R-NSGA-II, this preference operator uses the preference S-KKTPM metric.

We now characterize an iteration of the introduced R-NSGA-II with S-KKTPM process in which the DM provides one or more RPs in the following section (see Algorithm 2). Both parents and children are merged as usual, and the non-dominated sorting strategy is employed to classify the merged population into non-domination levels (so-called fronts).

The following are the primary ideas underlying selecting the preferred set of solutions within the preferred range:

- (i) Solutions closest to RP are always prioritized.
- (ii) Preferred-region sizing strategy is used to control the preferred range near RP.
- (iii) ϵ -based selection strategy is utilized to keep the spread of solutions within the range assigned by the DMs.

The following changes are made to the original NSGA-II niching approach to integrate the three notions mentioned above:

Step 1. Generating a desired region for each RP. The Euclidean distance between all members from the merged population and an RP is computed to specify the desired region. Then, the member that has the least Euclidean distance to RP is identified. The specified member (or point) is called mid-point as illustrated in Figure 7.

Step 2. Determining the size of the desired region for each RP. Here, we introduce a new strategy to determine the size of each desired area as follows. The solution within δ distance of the mid-point is chosen to be in the desired area. Parameter δ is given by the DM, which determines the size of the ROI, as illustrated in Figure 7. This figure also shows how to choose a population of size eight from the merged population containing 17 members. All solutions in the first front are selected, as shown in Figure 7. Then, we need only two solutions from the second front. The remaining two solutions are chosen (from the second front) as follows. The S-KKTPM value is calculated for each solution (\mathbf{x}) within the ROI. Then, the minimum of the appointed ranks is appointed as the S-KKTPM value to a solution (\mathbf{x}). If the solution (\mathbf{x}) is not within the preferred region, we set a high value for S-KKTPM (see Algorithm 3). In this manner, the smallest S-KKTPM

Input: \mathbf{x}
Output: S-KKTPM value
(1) **begin**
(2) Calculate $\nabla f_i(\mathbf{x})$ and $\nabla g_j(\mathbf{x})$, $i = 1, \dots, M$, $j = 1, \dots, J$.
(3) Solve equation (10) utilizing MATLAB's *fmincon()* function to find $\tilde{\xi}_k$
(4) **end**

ALGORITHM 1: S-KKTPM pseudocode.

Input: Population size (N), set of reference points $RP \leftarrow \{RP_1, RP_2, \dots, RP_L\}$, Generation number, $P_{\text{Crossover}}$, ϵ Parameter
Output: Children

- (1) Create initial parent population P_t of size N ;
- (2) **Repeat**
- (3) Generate offspring population Q_t from P_t by applying selection, crossover, and mutation operators;
- (4) Combine P_t and Q_t population (i.e., $R_t = P_t + Q_t$);
- (5) Classify R_t into different fronts (F_1, F_2 , etc., where F_1 is the best non-dominated front, F_2 is the next best non-dominated front, and so on) utilizing non-dominated sorting algorithm;
- (6) Calculate the S-KKTPM metric values of each front individual using the updated niching strategy specified in Algorithm 3;
- (7) Create a new parent population by choosing individuals, which are closer to the better front and have the lowest S-KKTPM value;
- (8) **Until** (maximum number of generations)

ALGORITHM 2: RS-KKTPM pseudocode.

Input: Population $P \leftarrow \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, set of reference points $RP \leftarrow \{RP_1, RP_2, \dots, RP_L\}$, set of preference radius $\delta \leftarrow \{\delta_1, \delta_2, \dots, \delta_L\}$

Output: Offspring solutions;

- (1) **begin**
- (2) **for** $i \leftarrow 1$ to L **do**
- (3) **for** $j \leftarrow 1$ to N **do**
- (4) $\text{dis}(RP_i, \mathbf{x}_j) = \text{Euclidean distance between } RP_i \text{ and } \mathbf{x}_j$;
- (5) **end for**
- (6) $\text{dis}_{RP_i, \mathbf{x}_{\min_{\text{dis}_i}}} = \min \{ \text{dis}(RP_i, \mathbf{x}_j) \}$;
- (7) **end for**
- (8) $\text{mid-point } MP \leftarrow \{ \mathbf{x}_{\min_{\text{dis}_1}}, \mathbf{x}_{\min_{\text{dis}_2}}, \dots, \mathbf{x}_{\min_{\text{dis}_L}} \}$;
- (9) **for** $i \leftarrow 1$ to L **do**
- (10) **for** $j \leftarrow 1$ to N **do**
- (11) $\text{dis}(MP_i, \mathbf{x}_j) = \text{Euclidean distance between } MP_i \text{ and } \mathbf{x}_j$;
- (12) **end for**
- (13) **end for**
- (14) **for** $i \leftarrow 1$ to L **do**
- (15) **for** $j \leftarrow 1$ to N **do**
- (16) **if** $\text{dis}(x_{\min_{\text{dis}_i}}, \mathbf{x}_j) \leq \delta_i$, **then**
- (17) Calculate S-KKTPM value at iterate \mathbf{x}_j //Algorithm 1;
- (18) **else**
- (19) Set S-KKTPM (\mathbf{x}_j) equal to 10^6 ;
- (20) **end if**
- (21) **end for**
- (22) **end for**
- (23) **end**

ALGORITHM 3: RS-KKTPM niching strategy.

value of one is given to the points that are closest to the PF. The next-to-smallest S-KKTPM value of two is given to the solutions with the next-to-smallest

S-KKTPM value to the true PF, and so on. Finally, the solutions with the smallest S-KKTPM are preferred to survive and transition to the new population.

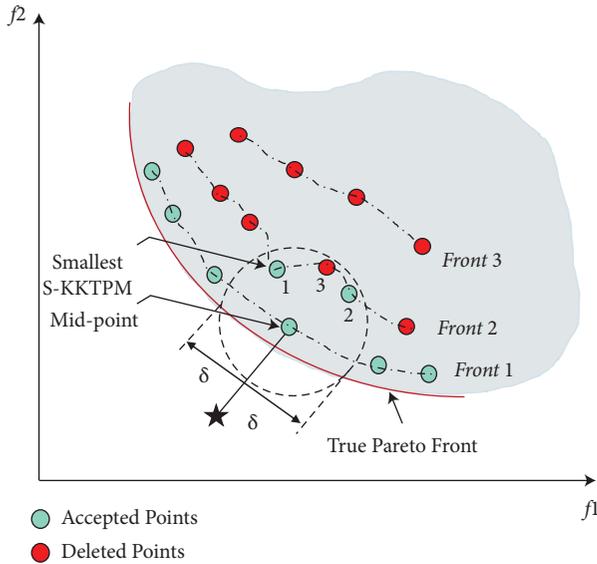


FIGURE 7: S-KKTPM-based metric in R-NSGA-II procedure. Numbers indicate the assigned ranks of points.

Step 3. Good distribution of the obtained solutions. The ϵ -clearing selection strategy, employed in the original R-NSGA-II, is used in RS-KKTPM to control the diversity of chosen solutions near the RPs. A solution is selected randomly from the set of non-dominated solutions to implement this strategy. Then, any solution with a sum of normalized differences in objective values less than ϵ is selected and then given a high-preference distance value to discourage it from remaining in the next generations of the evolution process. The way is then repeated with a new solution picked from the set of efficient points (excluding the one previously selected). The value of ϵ is selected according to the application and can be different for each objective. Thus, it is formed as a parameter provided by the DMs.

Figure 8 depicts how to determine the size of the ROI for each RP using the *mid-point* strategy. As discussed in step 1 above, the *mid-point* is a member of the population that is closest to the RP. As shown in Figure 8, RP can exist anywhere in the objective region (feasible or unfeasible), whereas the *mid-point* can exist anywhere in the feasible objective domain only. The purpose of the proposal of *mid-point* strategy can be summarized as follows: (1) getting PO solutions that are close to the given RP; (2) determining the size of the ROI by calculating the Euclidean distance between each solution and the *mid-point* (each distance value is normalized using zero as the lower bound and one as the upper bound to stay within the interval $[0, 1]$; the solutions that lie within δ value are candidates to be within the ROI); and (3) obtaining a good convergence of solutions towards the ROI. As discussed in step 2 above, the S-KKTPM metric acts as a differentiator in selecting a solution that should remain in the next generations of the optimization process. The solution with the smallest S-KKTPM value is preferentially kept for the next generations because it is the closest to the true PF. This way, the RS-KKTPM can obtain good

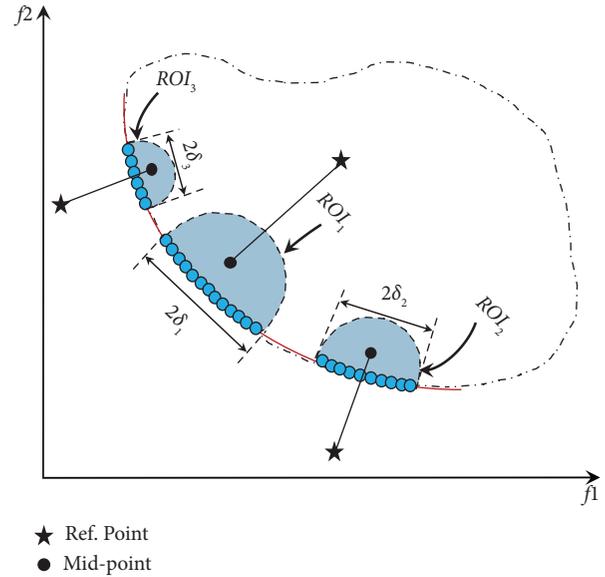


FIGURE 8: ROIs for DM with different ranges.

convergence of solutions towards the ROI. The introduced algorithm can well distribute solutions along the preferred part. RS-KKTPM works well with different RPs (feasible or infeasible) in the objective space, as displayed in Figure 8. In real-world applications, objectives should be normalized when they do not have the same units. Otherwise, δ is not a meaningful parameter.

One of the essential advantages of the introduced method is its ability to control the size of the preferred areas separately by a single simulation run (see Figure 8). This is done using the preferred-region sizing strategy discussed above, based on the S-KKTPM metric. This metric is used as a preference operator to select a subset of solutions close to the PF in order to move to the next population. As the iteration approaches the PF, S-KKTPM value decreases monotonically almost to the final minimum value (zero). This means that the S-KKTPM metric can know the proximity of a point in the search space to the PF. Through this strategy, the introduced algorithm can steer the solutions during the optimization process towards the preferred regions in proportion to the size of each area. In other words, the large ROI gets more PO solutions compared to the smaller preferred region.

On the other hand, the original R-NSGA-II algorithm cannot control the size of the preferred regions separately through a single run. The reason is the preferred-region sizing strategy used in this algorithm, which is based on the Euclidean distance metric. This metric is utilized as a preference operator in the R-NSGA-II algorithm. However, the Euclidean distance metric does not have the unique properties that the S-KKTPM metric does. For example, the final minimal Euclidean distance value is unknown. In other words, the Euclidean distance metric does not have any information about the proximity of a point to the PF. So, the R-NSGA-II algorithm cannot obtain different ranges of ROI in a single run.

TABLE 2: Parameters: columns from left to right represent problem name, number of objectives, number of variables, population size, maximum number of generations, reference points, and size of region of interest.

Problem	nobj M	nvar n	Popsiz N	MaxGen	Ref. point RP	Size of ROI δ
ZDT1	2	30	40	200	$RP_1 = (0.0, 0.8)$	0.05
					$RP_2 = (0.2, 0.6)$	0.05
					$RP_3 = (0.3, 0.3)$	0.05
					$RP_4 = (0.7, 0.25)$	0.05
					$RP_5 = (0.9, 0.0)$	0.05
ZDT2	2	30	40	200	$RP_1 = (0.9, 0.5)$	0.1
					$RP_2 = (0.3, 0.8)$	0.1
ZDT3	2	30	40	200	$RP_1 = (0.1, 0.4)$	0.1
					$RP_2 = (0.5, 0.4)$	0.1
					$RP_3 = (0.7, -0.25)$	0.1
ZDT4	2	10	80	500	$RP_1 = (0.6, 0.6)$	0.15
ZDT6	2	10	40	200	$RP_1 = (0.3, 0.8)$	0.03
					$RP_2 = (0.9, 0.4)$	0.05
					$RP_3 = (0.64, 0.59)$	0.1
BNH	2	2	40	200	$RP_1 = (10, 20)$	0.1
					$RP_2 = (60, 15)$	0.1
					$RP_3 = (90, 5)$	0.1
SRN	2	2	40	200	$RP_1 = (170, -200)$	0.1
					$RP_2 = (100, -50)$	0.1
OSY	2	6	40	200	$RP_1 = (-230, 40)$	0.05
					$RP_2 = (-100, 15)$	0.1
TNK	2	2	20	200	$RP_1 = (0.1, 0.90)$	0.1
					$RP_2 = (0.6, 0.85)$	0.1

TABLE 3: Parameters: columns from left to right represent the problem name, number of objectives, number of variables, population size, maximum number of generations, reference points, and size of region of interest.

Problem	nobj M	nvar n	Popsiz N	MaxGen	Ref. point RP	Size of ROI δ
DTLZ1	3	5	80	300	$RP_1 = (0.0, 0.25, 0.3)$	0.05
					$RP_2 = (0.0, 0.4, 0.15)$	0.05
	10	12	80	300	$RP_3 = (0.0, 0.15, 0.4)$	0.05
					$RP_1 = (0.05, 0.1, 0.15, 0.20, 0.25, 0.30, 0.35, 0.4, 0.45, 0.5)$	0.1
DTLZ2	3	8	60	300	$RP_1 = (0.6, 0.5, 0.8)$	0.05
					$RP_2 = (0.2, 0.2, 0.6)$	0.05
	10	19	60	300	$RP_1 = (0.4, 0.4, 0.4, 0.4, 0.4)$	0.02
					$RP_2 = (0.1, 0.1, 0.1, 0.1, 0.7)$	0.02
DTLZ5	3	12	60	300	$RP_1 = (0.1, 0.2, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.2, 0.1)$	0.05
					$RP_1 = (0.6, 0.6, 0.65)$	0.2
					$RP_2 = (0.2, 0.3, 0.8)$	0.1

5. Experimental Results and Discussion

This section uses a set of benchmark problems and engineering design problems to test our introduced methodology. Specifically, we adopted five two-objective unconstrained problems taken from the ZDT test suite [45], four bi-objective constrained problems (BNH, SRN, OSY, and TNK) taken from [46], and seven test problems having from three to ten objective functions taken from the DTLZ test suite [47]. In addition, we adopted two engineering design problems, the welded beam design problem with two

objective functions (taken from [48]) and the car side impact design problem with three objective functions (taken from [49]). Then, we compare the performance of the RS-KKTPM approach with six EMO preference approaches, including R-NSGA-II, g-NSGA-II [27], r-NSGAI [28], R-NSGA-III [50], WV-MOEA-P [51], and MOEA/D-PRE.

The parameters of the suggested method are set as follows:

- (i) Reproduction operators: as suggested in original study [26], simulated binary crossover (SBX)

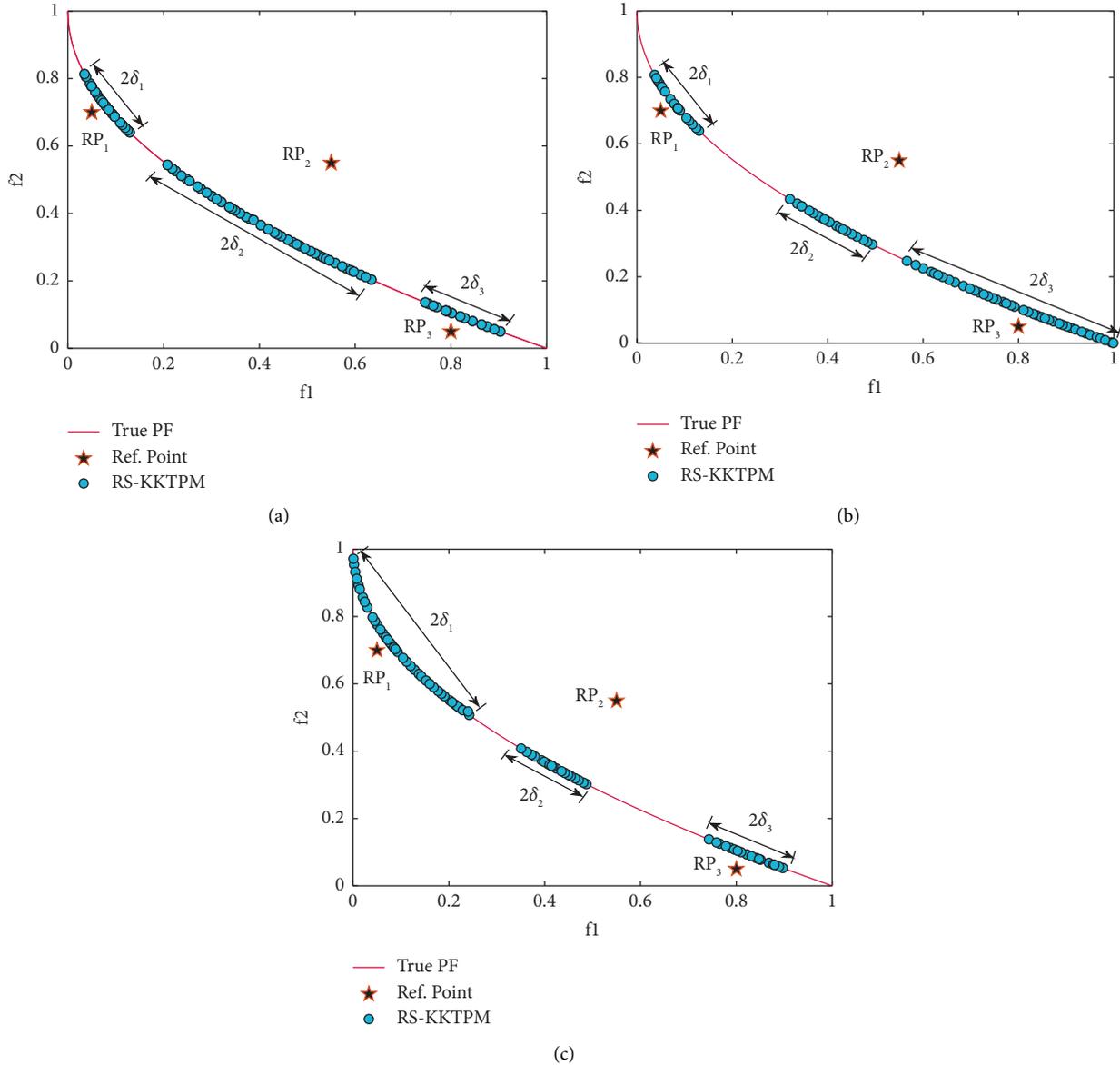


FIGURE 9: Effect of δ in obtaining different sizes of desired regions on ZDT1. (a) $\delta_1 = 0.05$; $\delta_2 = 0.15$; and $\delta_3 = 0.05$. (b) $\delta_1 = 0.05$; $\delta_2 = 0.05$; and $\delta_3 = 0.15$. (c) $\delta_1 = 0.15$; $\delta_2 = 0.05$; and $\delta_3 = 0.05$.

probability and SBX index are set to 0.9 and 10, respectively, and the polynomial mutation probability and mutation index are set to $1/n$ and 20, respectively.

- (ii) Population size, maximum number of generations, RPs, and size of ROI (δ): different parameters for a set of different test instances are displayed in Tables 2 and 3.

For constraint handling in constraint test problems and engineering design problems, we handled it by adding a penalty proportional to the constraint violation to the objective function value as suggested in the original NSGA-II algorithm. In minimization problems, this is a popular approach to deal with constraints in evolutionary algorithms.

The proposed RS-KKTPM algorithm is implemented in the MATLAB R2019a platform. The source codes for the comparison methods are provided by PlatEMO [51] or downloaded from the authors' home page. The suggested and compared methods are simulated on a personal computer with an Intel(R)Core(TM)i7-7500 2.9 GHz Quad-Core Processor and 8 GB RAM.

5.1. Experiments on Two-Objective Unconstrained ZDT Problems. Now, we apply our proposed approach to ZDT1 unconstrained problem (it has a convex PF) with thirty variables. Figure 9 illustrates the influence of different values of δ on the distribution of solutions obtained by RS-KKTPM after 200 generations (i.e., 16000 evaluations, given that RS-KKTPM evaluates 80 offsprings per generation). Three RPs

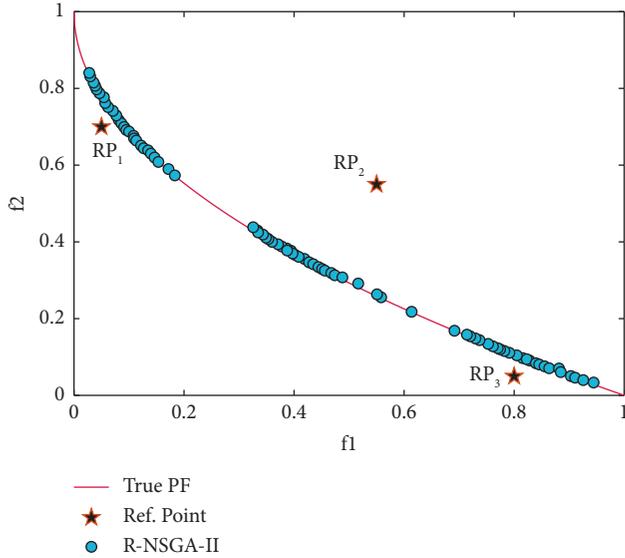


FIGURE 10: Efficient solutions obtained by R-NSGA-II for three RPs on ZDT1.

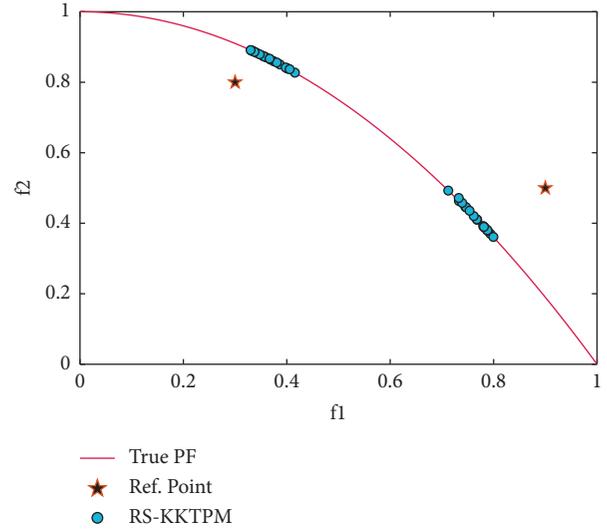


FIGURE 12: Efficient solutions obtained by RS-KKTPM on ZDT2 with two RPs.

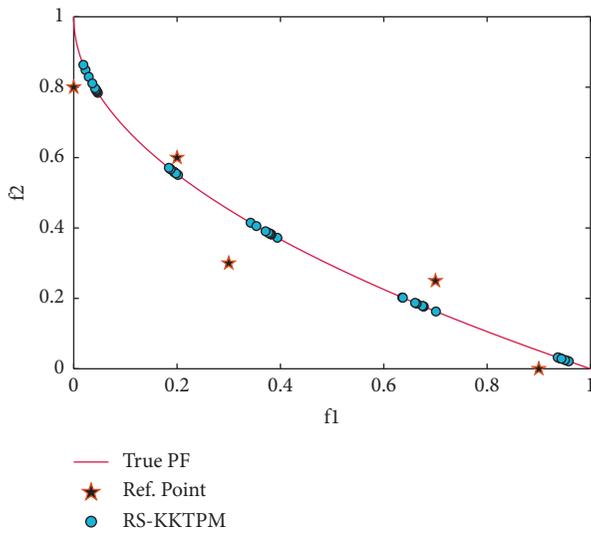


FIGURE 11: Efficient solutions obtained by RS-KKTPM on ZDT1 with five RPs.

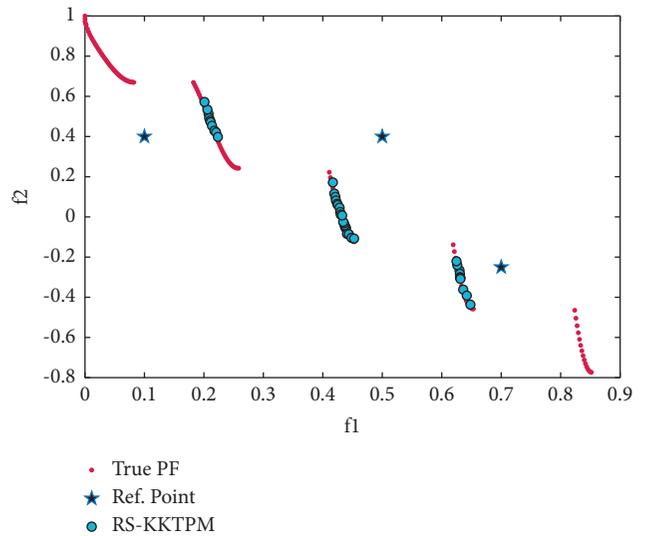


FIGURE 13: Efficient solutions obtained by the RS-KKTPM on ZDT3 with three RPs.

are chosen: $RP_1 = (0.8, 0.05)$, $RP_2 = (0.55, 0.55)$, and $RP_3 = (0.05, 0.7)$. These RPs are shown in the filled stars. RP_1 and RP_3 lie in infeasible search space while RP_2 lies in feasible search space. The different values of δ corresponding to the RPs are detailed in Figures 9(a)–9(c).

Through different values of δ , the proposed algorithm can steer the solutions towards the preferred regions in proportion to the size of each region. Parameter ϵ is still required to ensure that the obtained solutions are well distributed within preferred region. In this problem, the parameter $\epsilon = 0.005$ is chosen. The solutions obtained are clustered near the RPs, as shown in Figures 9(a)–9(c). The distribution of the obtained PO set depends on the range of each desired region. In particular, the range of solutions

obtained is equally vast when the value of δ is large. One of the advantages of RS-KKTPM is that it allows us to adjust the ranges for the desired region in a single run. Thus, if the DM wants to get a set of solutions (near each preferred region) whose number varies depending on the size of each preferred region separately, different values of δ can be chosen. In other words, the DM can control the spread of the generated ROIs by changing the value of parameter δ . If $\delta = 0.5$, the RS-KKTPM provides an approximation of the entire PF. On the contrary, Figure 10 shows the PO set produced utilizing R-NSGA-II for the same three RPs on ZDT1 problem. R-NSGA-II is also performed with $\epsilon = 0.005$ and a population of size 80. It is run until 200 generations. Figure 10 shows that the DM (by RNSGA-II) cannot obtain different regions of desired regions in a single run. Also, R-

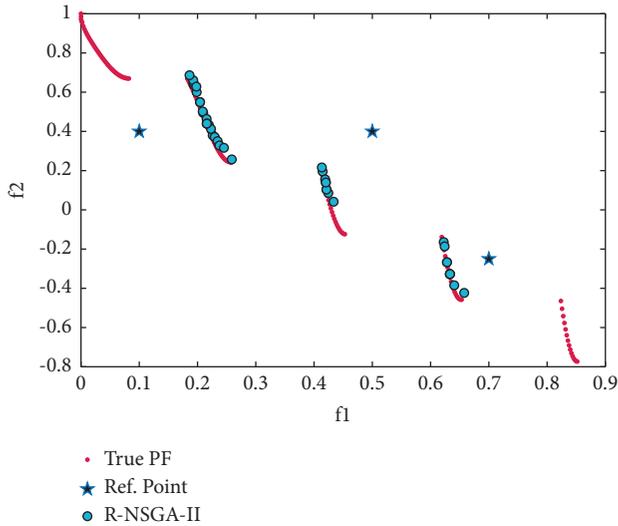


FIGURE 14: Efficient solutions obtained by R-NSGA-II on ZDT3 with three RPs.

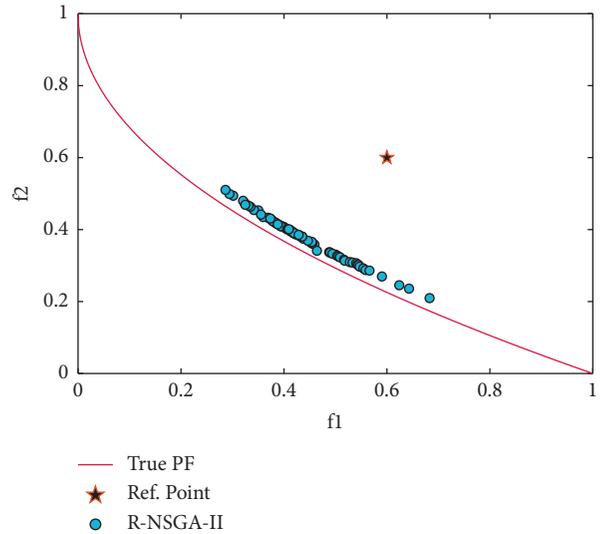


FIGURE 16: Efficient solutions obtained by R-NSGA-II on ZDT4 with one RP.

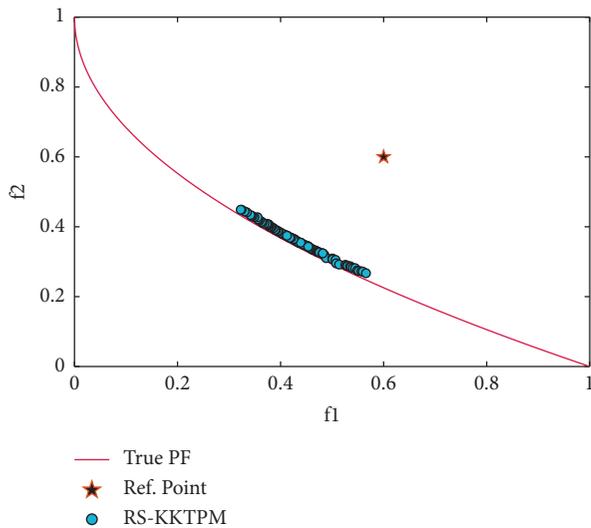


FIGURE 15: Efficient solutions obtained by RS-KKTPM on ZDT4 with one RP.

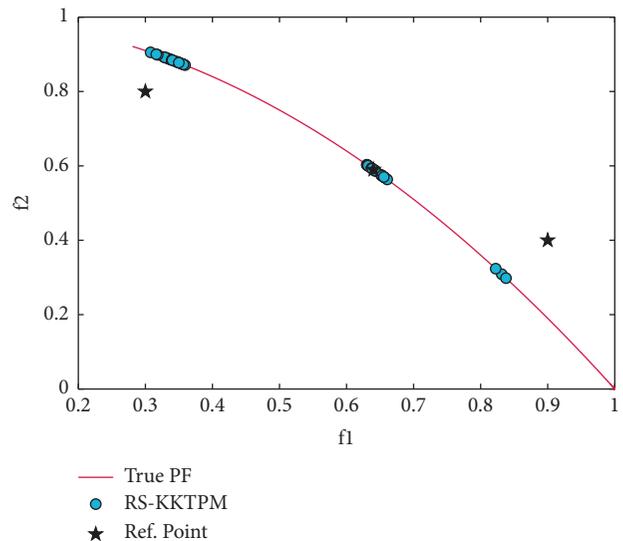


FIGURE 17: Efficient solutions obtained by RS-KKTPM on ZDT6 with three RPs.

NSGA-II cannot steer the solutions toward the preferred regions in proportion to the size of each region.

Henceforth, the parameter $\epsilon=0.001$ is used in all problems. First, we consider ZDT1 test problem with five RPs, of which three are infeasible and two are feasible, as shown in Figure 11. Each RP and corresponding size of ROI are shown in Table 2. RS-KKTPM is utilized for this problem, where the population members and the maximum number of generations are 40 and 200, respectively. The parameter is set to 0.05 for each ROI. Figure 11 also demonstrates how easy the proposed algorithm can be modified to address multiple RPs. As a result, it discovers various ROIs. Well-convergent non-dominated solutions are obtained on PF near all the five RPs.

ZDT2 is the next problem which has a non-convex PF. Two RPs are chosen, of which one is feasible and the other is infeasible, as presented in Table 2. The range of each region corresponding to an RP is also presented in Table 2. The population members and maximum number of generations, respectively, are 40 and 200. Figure 12 displays the convergence and distribution of the solutions near the two chosen RPs. As shown in the figure, RS-KKTPM algorithm can easily deal with feasible and infeasible RPs. The proposed algorithm proves its ability to converge and distribute the solutions obtained within the desired ranges provided by the DMs, as illustrated in Figure 12. RS-KKTPM also showed good distribution on this problem when RP is in the infeasible region.

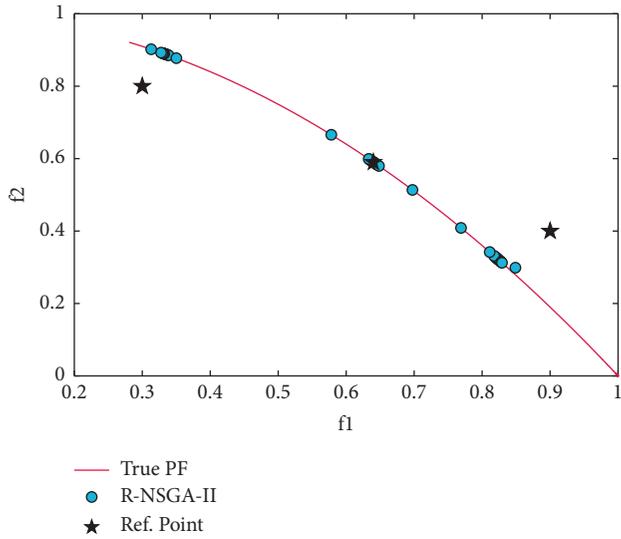


FIGURE 18: Efficient solutions obtained by R-NSGA-II on ZDT6 with three RPs.

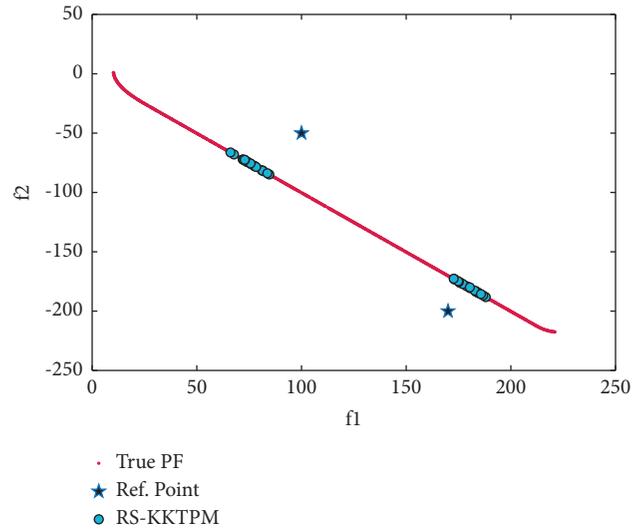


FIGURE 20: Efficient solutions obtained by RS-KKTPM on SRN with two RPs.

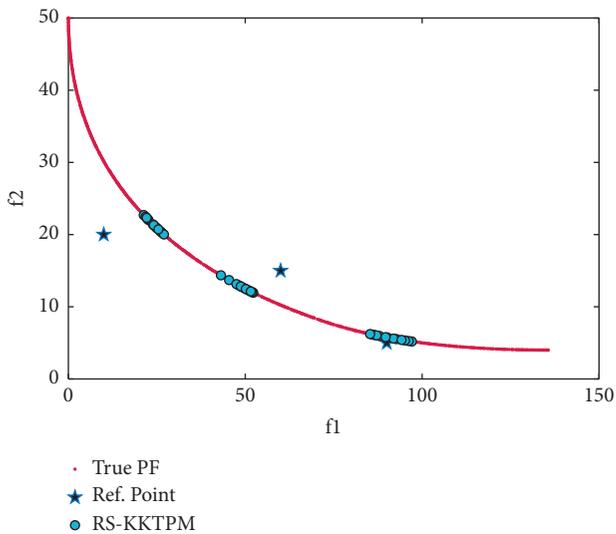


FIGURE 19: Efficient solutions obtained by RS-KKTPM on BNH with three RPs.

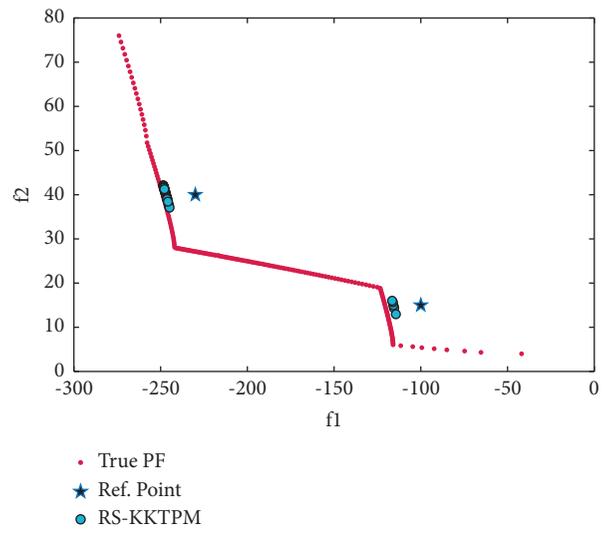


FIGURE 21: Efficient solutions obtained by RS-KKTPM on OSY with two RPs.

The test problem ZDT3, with 30 variables, has a disconnected set of PFs. Three RPs are selected (see Table 2), of which one is infeasible and two are feasible. The desired solutions produced by RS-KKTPM and R-NSGA-II are illustrated in Figures 13 and 14. The population members were 40, and the maximum number of generations was 200. These two figures demonstrate that our approach is able to steer solutions towards the PF in proportion to the size of each ROI, while the R-NSGA-II cannot. As illustrated in Figure 13, our approach does not get stuck in any locally PO part, and all generated solutions are non-nominated and global PO solutions.

Next, the test problem of ZDT4 with 10 variables is solved utilizing RS-KKTPM and R-NSGA-II. This problem has many local PFs. One RP is used with a range of ROI of

0.15, as displayed in Table 2. The RP is (0.6, 0.6), and the generations are 500. The plot of the desired solutions produced by RS-KKTPM and R-NSGA-II is represented in Figures 15 and 16, respectively. As illustrated by the two figures, the performance of RS-KKTPM is much better than that of R-NSGA-II in terms of the distribution and convergence of solutions from the PF. As shown in Figure 15, the selected RP is somewhat far from the PF, which indicates the ability of the introduced approach to work well in the case of distant RPs. Thus, as shown in Figure 15, even though the problem has more than 100 local fronts, the introduced algorithm can converge well to the true PF.

Finally, we apply our proposed method to a ZDT6 problem that has a non-convex PF. Figures 17 and 18 display the obtained PO solutions by RS-KKTPM and R-NSGA-II,

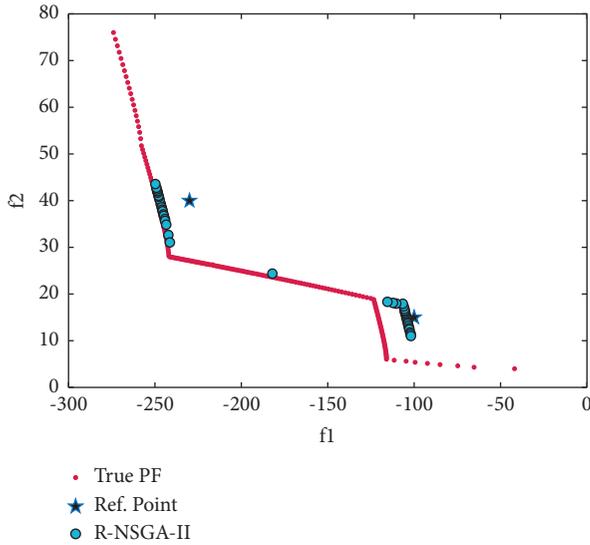


FIGURE 22: Efficient solutions obtained by R-NSGA-II on OSY with two RPs.

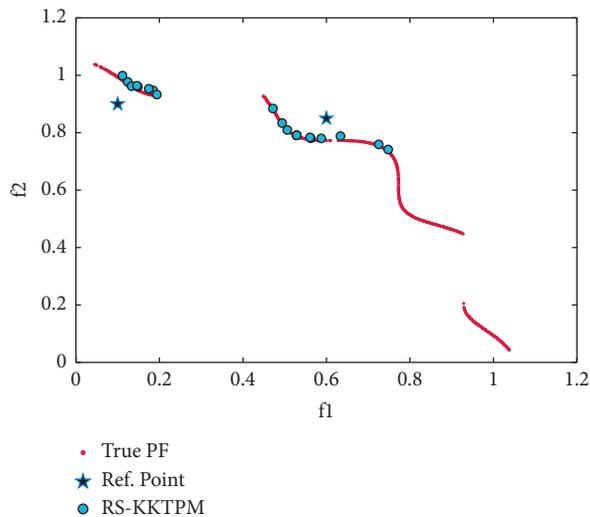


FIGURE 23: Efficient solutions obtained by RS-KKTPM on TNK with two RPs.

respectively. Both techniques used the same RPs, the same number of population members, and the same number of generations (see Table 2). Three RPs are chosen, of which the first lies in feasible search space, the second lies close to/on PF, and the third lies in infeasible search space. For RS-KKTPM, the sizes of ROIs corresponding to RPs (0.9, 0.4), (0.3, 0.8), and (0.64, 0.59) are 0.03, 0.05, and 0.10. For R-NSGA-II, the size of ROIs for all RPs is 0.10. Note that R-NSGA-II cannot adjust the size of each ROI separately, as in RS-KKTPM. As it is clear from Figure 17, the introduced algorithm can seek solutions towards the ROI in proportion to the size of each area separately, whereas R-NSGA-II cannot. The ROI corresponding to RP (0.64, 0.59), with $\delta = 0.1$, contains a large number of points compared to the ROI corresponding to RP (0.3, 0.8) with $\delta = 0.05$, as shown in

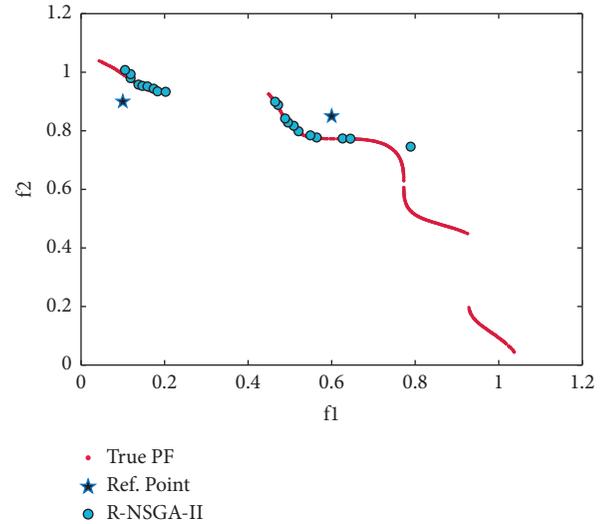


FIGURE 24: Efficient solutions obtained by R-NSGA-II on TNK with two RPs.

Figure 17. Also, the ROI corresponding to RP (0.9, 0.4), with $\delta = 0.03$, contains a few number of solutions compared to the ROIs corresponding to RPs (0.3, 0.8) and (0.64, 0.59). This means that if the DM wants to get PO solutions of different sizes for all regions, the introduced algorithm can do that. In contrast, R-NSGA-II cannot control the number of solutions for each desired area. This is because parameter ϵ takes only one value for all preferred regions corresponding to the given RPs. In other words, when multiple RPs exist, R-NSGA-II cannot give different values for ϵ in a single run. This means that if the DM wants to get PO solutions of different sizes for all regions, the R-NSGA-II algorithm cannot do that.

5.2. Experiments on Two-Objective Constraint Problems.

We now consider two-objective constraint problems: BNH, SRN, OSY, and TNK [46]. RPs and some essential parameters used to solve these problems are shown in Table 2. BNH, TNK, and SRN have only two constraints and two variables. First, the efficient solutions obtained by the RS-KKTPM on BNH with three RPs are illustrated in Figure 19. It is clear from this figure that our approach can find the desired regions near the RPs. Second, the solutions obtained on SRN with two RPs are displayed in Figure 20. The RS-KKTPM algorithm works well when the RP is in the feasible or infeasible domain, as displayed in Figure 20. Next, we consider the OSY test problem, which has six constraints and six variables. Figures 21 and 22 show the obtained solutions by RS-KKTPM and R-NSGA-II on OSY, respectively. Two RPs are chosen with a range of ROIs and a population = 40 (see Table 2). Although RS-KKTPM cannot converge to the true PF, it converges slightly better than R-NSGA-II, as shown in Figures 21 and 22.

Finally, the desired regions obtained by the RS-KKTPM and R-NSGA-II on the TNK problem are illustrated in Figures 23 and 24, respectively. Two RPs and population members are chosen as displayed in Table 2. As shown in

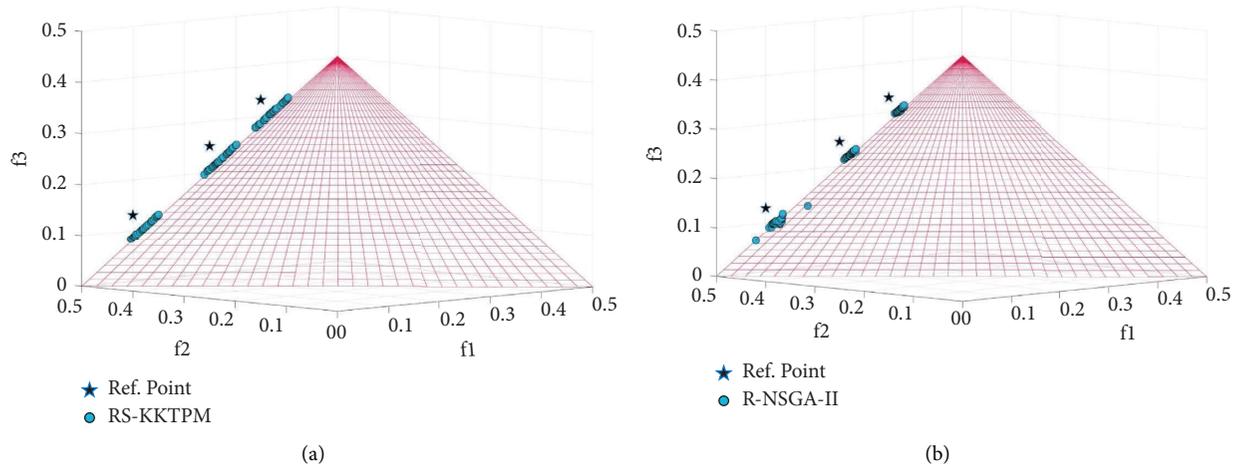


FIGURE 25: Comparison of performances on three-objective DTLZ1 with three RPs: (a) RS-KKTPM; (b) R-NSGA-II.

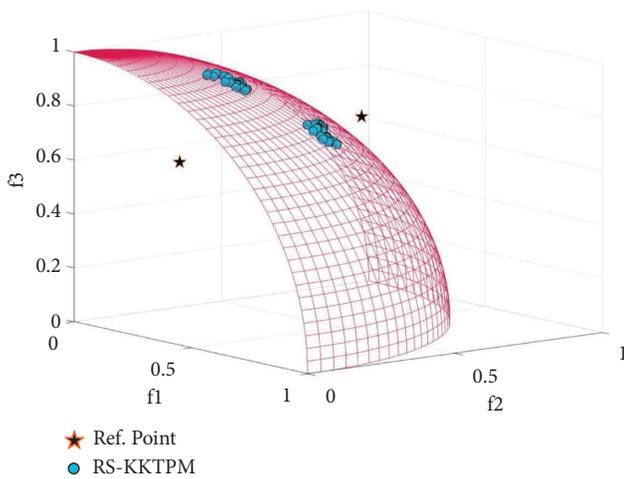


FIGURE 26: Efficient solutions obtained by RS-KKTPM on three-objective DTLZ2 with two RPs.

Figures 23 and 24, the performance of our approach is a little similar to R-NSGA-II. In summary, the introduced approach balances diversity and convergence around ROI for constraint test problems and handles any number of predefined RPs.

5.3. Experiments on Three-Objective Problems. We will select the original DTLZ1, DTLZ2, and DTLZ5 and their scaled versions. Table 3 provides some information about these problems and some parameters required. First, the DTLZ1 problem contains many local PFs, possibly causing some points to stop. This scenario is a relatively complicated problem to address for global optimality. Figure 25 shows the obtained preferred PO solutions using RS-KKTPM and R-NSGA-II algorithms on the three-objective DTLZ1 ($DTLZ1_3$) problem, and the parameter values are presented in Table 3. The three aspiration points are chosen. The distribution and convergence of solutions found by RS-

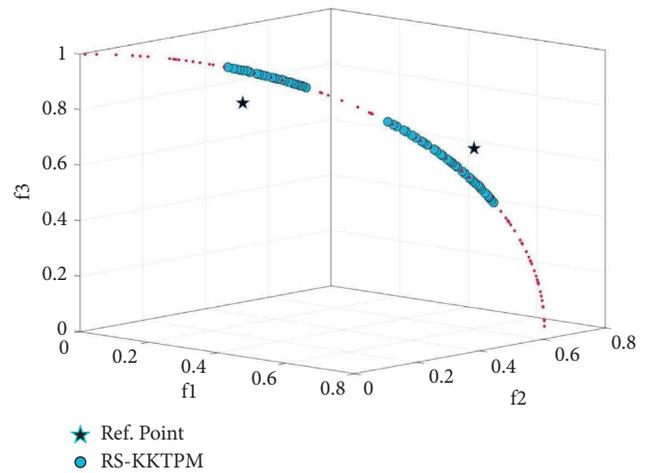


FIGURE 27: Efficient solutions obtained by RS-KKTPM on three-objective DTLZ5 with two RPs.

KKTPM are substantially superior to those by R-NSGA-II, as displayed in Figure 25. Next, RS-KKTPM is utilized to solve the three-objective DTLZ2 problem. Figure 26 shows the obtained solutions with 60 population members with two RPs. Figure 26 clearly illustrates that the RS-KKTPM algorithm can access the efficient region of true PF with few number of population sizes, thereby helping the DM determine the required ROI easily. Finally, the RS-KKTPM is utilized to solve the three-objective DTLZ5 problem. The two RPs, $RP_1 = (0.6, 0.6, 0.65)^T$ and $RP_2 = (0.2, 0.3, 0.8)^T$, are used. The sizes of the preferred areas for RP_1 and RP_2 are 0.2 and 0.1, respectively, as shown in Table 3. Our algorithm is employed to solve this test problem with 60 populations and runs up to 300 generations. The obtained preferred areas of the true PO solutions are displayed in Figure 27. The obtained solutions are distributed according to the size of each preferred area. In a single simulation run, both areas are discovered. Note that the number of solutions generated in the first preferred area, corresponding to RP_1 , is greater than that generated in the second preferred area, corresponding

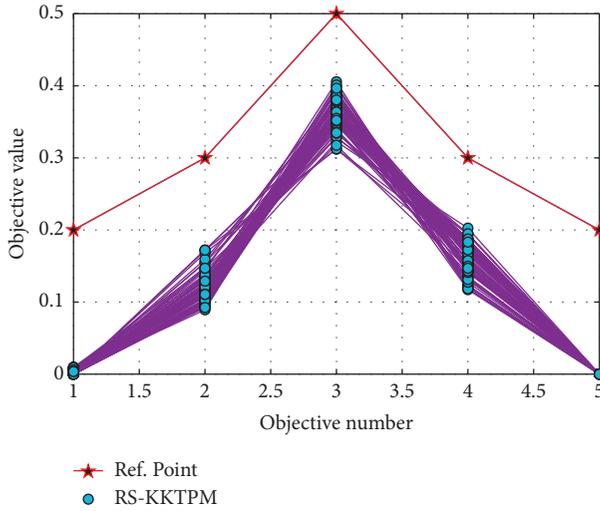


FIGURE 28: Efficient solutions obtained by RS-KKTPM on five-objective DTLZ1 with one RP.

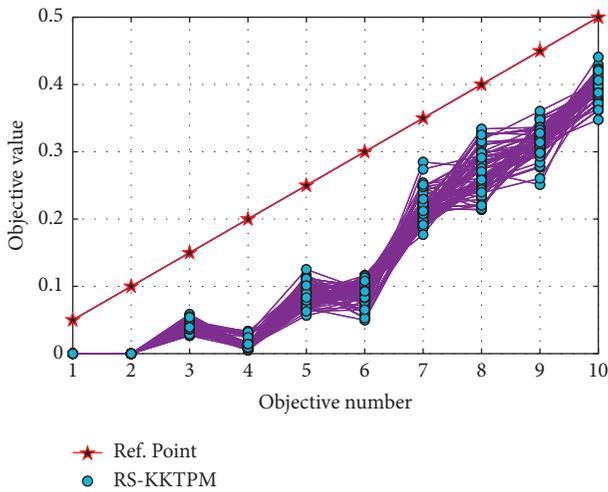


FIGURE 29: Efficient solutions obtained by RS-KKTPM on ten-objective DTLZ1 with one RP.

to RP_2 . Well-convergent and well-distributed solutions are obtained on PF in all two RPs. Thus, the DM can control the size of each efficient region (s) of the true PF separately and in a single simulation run.

5.4. Experiments on Many-Objective Problems. Finally, we test our introduced approach on the many-objective versions of the problems of DTLZ1 and DTLZ2. Table 3 displays all parameters used for 5 and 10-objective problems. First, RS-KKTPM is used for $DTLZ1_5$ and $DTLZ1_{10}$ problems. Population sizes of 80 are used for the two problems. Figures 28 and 29 present the obtained part in a parallel coordinate plot. One RP is used for each problem, as shown in Table 3. The PO solutions of these problems must satisfy $\sum_{i=1}^M f_i = 0.50$. RS-KKTPM can discover the needful regions of the efficient set corresponding to the one predefined RP by the DM.

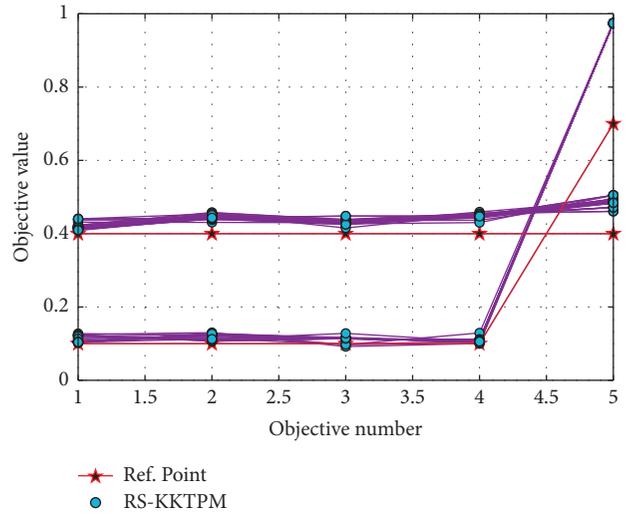


FIGURE 30: Efficient solutions obtained by RS-KKTPM on five-objective DTLZ2 with two RPs.

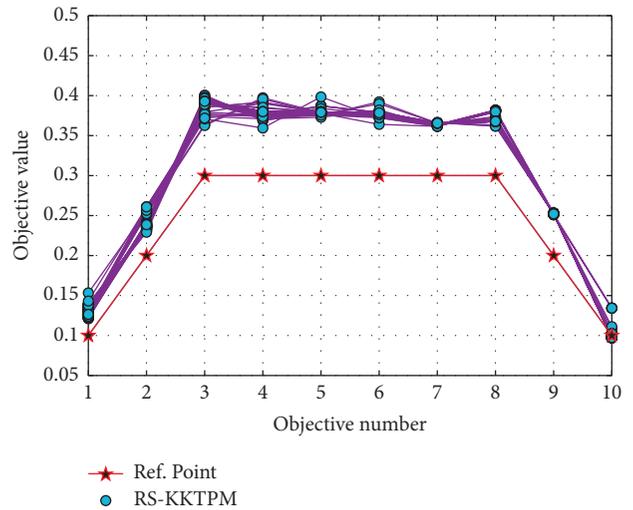


FIGURE 31: Efficient solutions obtained by RS-KKTPM on ten-objective DTLZ2 with one RP.

Finally, the RS-KKTPM algorithm is applied for $DTLZ2_5$ and $DTLZ2_{10}$ with 14 and 19 decision variables. This algorithm is applied for these test problems with 60 populations and runs up to 300 generations. Figures 30 and 31 show the obtained solutions for $DTLZ2_5$ and $DTLZ2_{10}$ with two and one aspiration points, respectively. The PO solutions to these problems must obey the next equation: $\sum_{i=1}^M f_i^2 = 1$. When computing the left side of this equation for all generated PO solutions, all the values lie in the range $[1.000002, 1.000170]$ for $DTLZ2_5$ problem and $[1.000043, 1.002780]$ for $DTLZ2_{10}$ problem, indicating that every solution is very close to the true PF. Thus, RS-KKTPM can converge to the PF corresponding to the chosen aspiration points.

5.5. Experiments on Engineering Problems. We now apply the RS-KKTPM to a couple of engineering design problems. The first test problem has two objectives, while the second test problem has three.

5.5.1. *Welded Beam Design Problem.* First, we now employ a two-objective welded beam design problem [48] as a real-world example. The first objective is to minimize the cost of fabrication, whereas the other objective is to minimize the end deflection of the welded beam. The design of welded beam structure is shown in Figure 32. This problem involves four decision variables, namely, h (weld thickness), l (clip length), t (the height of bar), and b (the thickness of bar). It has also four non-linear constraints. The problem is mathematically formulated as follows [48]:

$$\text{minimize: } \begin{cases} f_1(\mathbf{x}) = 1.10471h^2l + 0.04811\left(\left(\frac{28}{2}\right) + l\right)tb, \\ f_2(\mathbf{x}) = \frac{1372}{(625bt^3)}, \end{cases}$$

subject to

$$g_1(\mathbf{x}) = \tau - 13,600 \leq 0, g_2(\mathbf{x}) = \sigma - 30,000 \leq 0,$$

$$g_3(\mathbf{x}) = h - b \leq 0, g_4(\mathbf{x}) = 6,000 - P_c \leq 0,$$

$$0.125 \leq h, b \leq 5.0, \text{ and } 0.1 \leq l, t \leq 10.0,$$

(11)

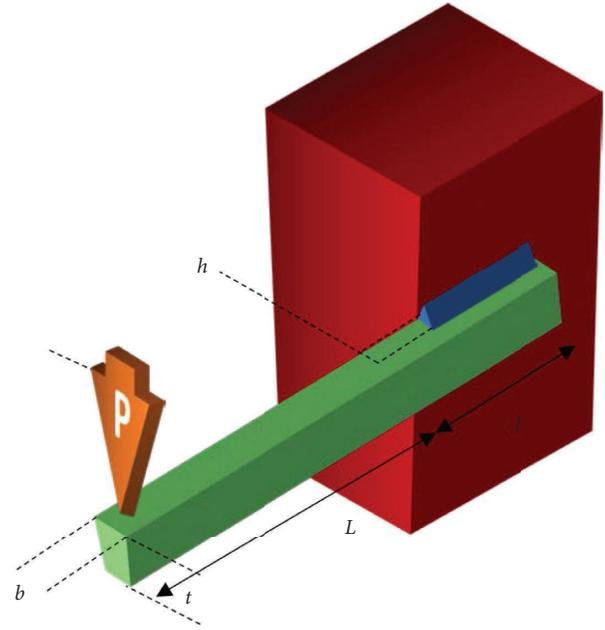


FIGURE 32: Welded beam design [52].

where

$$\left\{ \begin{array}{l} v = \frac{(6 * 10^3)}{(\text{sqrt}(2) * hl)}, \\ \kappa_1 = (6 * 10^3) \left(\left(\frac{28}{2} \right) + \left(\frac{1}{2} \right) l \right) * \sqrt{\left(\left(\frac{1}{4} \right) (l^2 + (t+h)^2) \right)}, \\ \kappa_2 = \left(\frac{707}{1000} \right) hl \left(\frac{l^2}{12} + \left(\frac{1}{4} \right) (l^2 + (t+h)^2) \right), \\ \alpha = \frac{\kappa_1}{2\kappa_2}, \\ \beta = \frac{l v \alpha}{\text{sqrt}((1/4)(l^2 + (t+h)^2))} \\ \tau = \sqrt{(v^2 + \alpha^2 + \beta)}, \\ \sigma = \frac{(504 * 10^3)}{(bt^2)}, \\ P_c = \left(\frac{43 * 514243}{36} \right) * \text{sqrt} \sqrt{\left(\frac{b^6 * t^2}{(144/4)} \right)} \left(1.0 - t \left(\frac{2680}{94919} \right) \right). \end{array} \right. \quad (12)$$

Figure 33 shows the efficient solutions produced by RS-KKTPM, R-NSGA-II, R-NSGA-III, and MOEA/D-PRE [30], respectively, to the welded beam design problem. The relevant parameters of the four comparison algorithms are briefly

presented below. Three RPs are chosen: $RP_1 = (3, 0.005)$, $RP_2 = (15, 0.003)$, and $RP_3 = (25, 0.002)$. The comparison algorithms are used with 100 population members and run for 200 generations. For RS-KKTPM approach, the radius

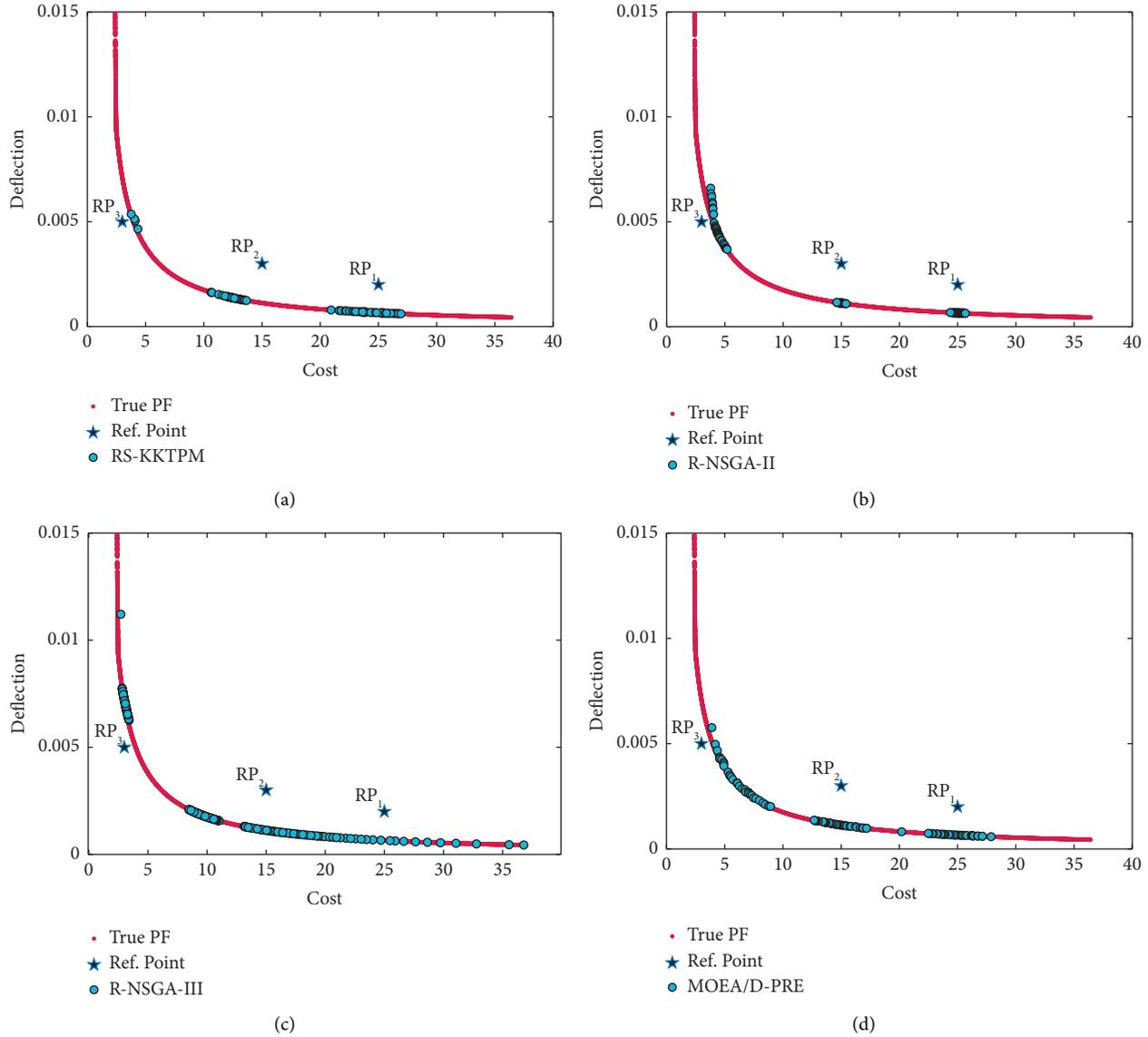


FIGURE 33: Comparison of performances on the welded beam design problem with three RPs: (a) RS-KKTPM; (b) R-NSGA-II; (c) R-NSGA-III; (d) MOEA/D-PRE.

corresponding to each RP is $\delta_1 = 0.2$, $\delta_2 = 0.1$, and $\delta_3 = 0.05$. For the rest of the algorithms, the size of preferred regions is equal to 0.1. Note that the proposed algorithm can control the size of each region separately, while other algorithms cannot. In this problem, all objective function values are normalized using the ideal point as the lower bound and the nadir point estimation as the upper bound to stay within the interval $[0, 1]$. We used $(0, 0)$ as the ideal point and $(36, 0.015)$ as the nadir point.

Figure 33(a) shows that the introduced algorithm outperforms the others by adjusting the size of each preferred region (separately) corresponding to the supplied RP. It has the ability to steer the solutions towards the preferred regions in proportion to the size of each area. As shown in Figure 33(a), the obtained solutions in the preferred region, corresponding to RP_1 , are more compared to the obtained solutions in the other two preferred regions. Indeed, the size of the ROI, corresponding to RP_1 , is greater than the sizes of

the other two preferred regions, i.e., $\delta_1 > \delta_2$ and $\delta_1 > \delta_3$. On the other hand, the preferred region, corresponding to RP_3 , contains a few solutions compared to the other two preferred regions because $\delta_3 < \delta_1$ and $\delta_3 < \delta_2$. Figure 33(a) also shows that the RS-KKTPM can produce well-distributed solutions along the preferred part. The advantages of the RS-KKTPM, discussed above, are mostly not found in R-NSGA-II, R-NSGA-III, and MOEA/D-PRE (see Figures 33(b)–33(d)). In summary, if the DM is interested in finding PO solutions in three main areas (intermediate cost and deflection, minimum cost, and minimum deflection), the introduced algorithm can find solutions near the given RPs, rather than finding solutions on the whole PF, allowing the DM to deal with only a few solutions that lie in parts of her/his interest. Moreover, if the DM is interested in finding these solutions within different sizes for all regions, the proposed algorithm can provide them.

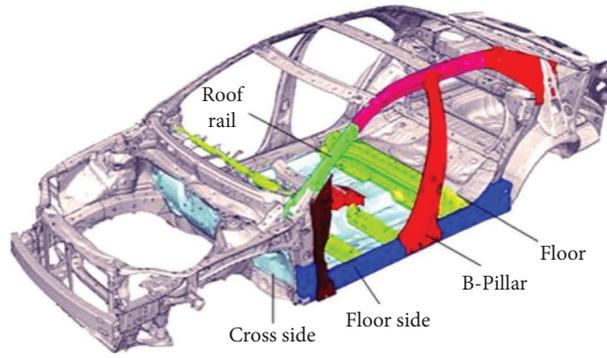


FIGURE 34: Car side impact design [53].

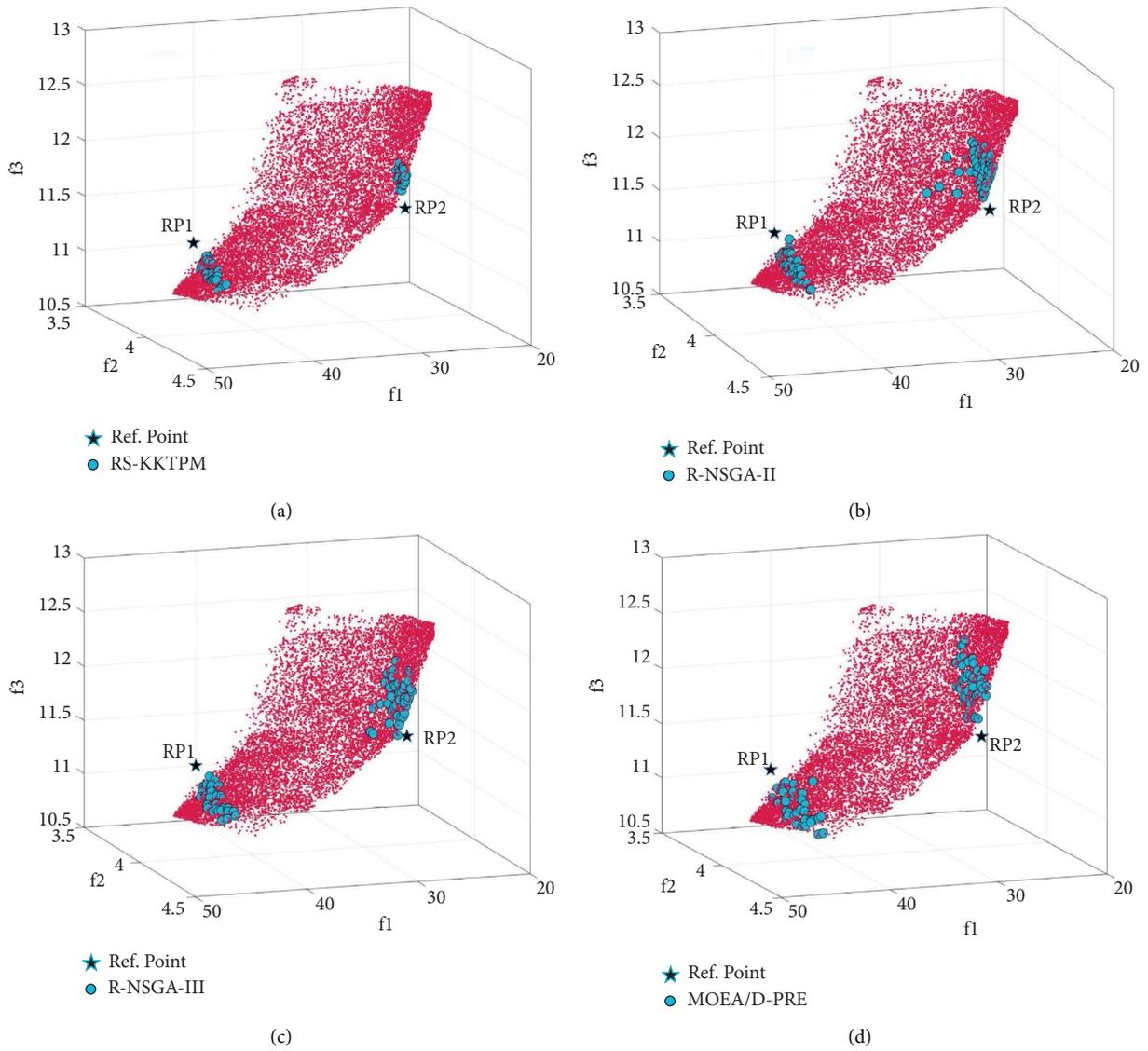


FIGURE 35: Comparison of performances on the car side impact problem with two RPs: (a) RS-KKTPM; (b) R-NSGA-II; (c) R-NSGA-III; (d) MOEA/D-PRE.

5.5.2. *Car Side Impact Design Problem.* Car side impact design is a constrained optimization problem [49]. This problem has three optimization objectives which are described as follows: the first is to reduce the car's weight, the second objective is to minimize the pubic force experienced by a passenger, and the last objective is to minimize the

average velocity of the V-Pillar responsible for withstanding the impact load. It has seven decision variables: B-Pillar, door beam, B-Pillar inner reinforcement, floor side inner, door beltline reinforcement, cross members, and roof rail (see Figure 34). The mathematical model of this problem is as follows [49]:

minimize:

$$\begin{cases} f_1(\mathbf{x}) = 1.98 + 4.9v_1 + 6.67v_2 + 6.98v_3 + 4.01v_4 + 1.78v_5 + 0.00001v_6 + 2.73v_7, \\ f_2(\mathbf{x}) = 4.72 - 0.5v_4 - 0.19v_2v_3, \\ f_3(\mathbf{x}) = 0.5(10.58 - 0.674v_1v_2 - 0.67275v_2 + 16.45 - 0.489v_3v_7 - 0.843v_5v_6), \end{cases}$$

subject to

$$\begin{aligned} g_1(\mathbf{x}) &= 1.160 - 0.3717v_2v_4 - 0.0092928v_3 \leq 1.0, \\ g_2(\mathbf{x}) &= 0.261 - 0.0159v_1v_2 - 0.06486v_1 - 0.019v_2v_7 \\ &\quad + 0.0144v_3v_5 + 0.0154464v_6 \leq 0.32, \\ g_3(\mathbf{x}) &= 0.214 + 0.00817v_5 - 0.045195v_1 - 0.0135168v_1 \\ &\quad + 0.03099v_2v_6 - 0.018v_2v_7 + 0.007176v_3 \\ &\quad + 0.023232v_3 - 0.00364v_5v_6 - 0.018v_2v_2 \leq 0.32, \\ g_4(\mathbf{x}) &= 0.740 - 0.610v_2 - 0.031296v_3 - 0.031872v_7 + 0.227v_2v_2 \leq 0.32, \\ g_5(\mathbf{x}) &= 8.980 + 3.8180v_3 - 4.2v_1v_2 + 1.27296v_6 - 2.68065v_7 \leq 0.32, \\ g_6(\mathbf{x}) &= 33.860 + 2.95v_3 - 5.057v_1v_2 - 3.795v_2 - 3.4431v_7 \\ &\quad + 1.45728 \leq 0.32, \\ g_7(\mathbf{x}) &= 46.360 - 9.90v_2 - 4.4505v_1 \leq 0.32, \\ g_8(\mathbf{x}) &= 4.720 - 0.50v_4 - 0.190v_2v_3 \leq 4.0, \\ g_9(\mathbf{x}) &= 10.58 - 0.674v_1v_2 - 0.67275v_2 \leq 9.9, \\ g_{10}(\mathbf{x}) &= 16.45 - 0.489v_3v_7 - 0.843v_5v_6 \leq 15.7, \\ &0.5 \leq v_{1,3,4} \leq 1.5, \text{ and } 0.45 \leq v_2 \leq 1.35, 0.875 \leq v_5 \leq 2.625, \text{ and } 0.4 \leq v_{6,7} \leq 1.2. \end{aligned} \tag{13}$$

Figure 35 shows the solutions produced by RS-KKTPM, R-NSGA-II, R-NSGA-III, and MOEAD-PRE on the car side impact design problem. In this problem, all relevant parameters of the four comparison approaches are briefly presented below. Two RPs are chosen: $RP_1 = (40, 3.5, 11)$ and $RP_2 = (26, 4, 11.5)$. For this test problem, the comparison algorithms are used with 80 population members and run until 300 generations. For the RS-KKTPM algorithm, the radius corresponding to each RP is $\delta_1 = 0.1$ and $\delta_2 = 0.05$. For the rest of the algorithms, the size of preferred regions is equal to 0.1. In this problem, all objective function values are normalized using the ideal point (15, 3, 10) and nadir point (50, 5, 14). Figure 35(a) shows that the suggested algorithm can control the number of solutions for each desired area in proportion to its size. In other words, a larger desired area gets more solutions, while a smaller preferred region gets fewer solutions. As displayed in Figure 35(a), the number of solutions for the first preferred area, corresponding to the P_1 , is more than that for the second desired area, corresponding to the RP_2 . This is because the size of the first

region is greater than the size of the second region, i.e., $\delta_1 > \delta_2$. Therefore, RS-KKTPM can control the size of each desired area separately, while the rest of its algorithms cannot do that (see Figures 35(b)–35(d)). Thus, if the DM is interested in finding solutions in region of different sizes, the introduced algorithm can find solutions near the RPs and proportional to the size of each preferred area separately.

5.6. *Performance Metrics.* No single performance metric can provide an accurate assessment of an EMO's performance [54]. In our empirical investigations, we use two of the most recognized performance metrics to assess the quality of preferable efficient solutions of preference-based EMO algorithms: R-HV and R-IGD. [55]. Both metrics are utilized to detect the ROI's convergence and the diversity of efficient solutions simultaneously. They are based on two performance metrics, the hypervolume (HV) metric and the inverted generational distance (IGD) metric, which are designed for the entire PF and applicable for partial preferable efficient solutions. The larger the R-HV values are or

TABLE 4: Parameter setting of the test instances utilized in preference-based EMO algorithms.

Problem	Ref. point	Popsiz	Function evaluations
ZDT1	(0.4, 0.5)	40	8000
ZDT2	(0.9, 0.4)	40	8000
ZDT3	(0.1, 0.4)	40	8000
ZDT4	(0.6, 0.6)	80	40000
ZDT6	(0.5, 0.7)	40	8000
DTLZ1 ₃	(0.0, 0.4, 0.15)	80	24000
DTLZ2 ₃	(0.6, 0.5, 0.8)	60	18000
DTLZ5 ₃	(0.2, 0.3, 0.8)	60	18000
DTLZ1 ₅	(0.2, 0.3, 0.5, 0.3, 0.2)	80	24000
DTLZ2 ₅	(0.4, 0.4, 0.4, 0.4, 0.4)	60	18000
DTLZ1 ₁₀	(0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.4, 0.45, 0.5)	80	24000
DTLZ2 ₁₀	(0.1, 0.2, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.2, 0.1)	60	18000
Welded	(15, 0.0030)	100	20000
CAR	(40, 4, 11)	80	24000

TABLE 5: Findings of mean and standard deviation for R-HV values utilizing several preference-based EMO algorithms.

Problem	RS-KKTPM	R-NSGA-II	g-NSGA-II	r-NSGA-II	R-NSGA-III	WV-MOEA-P	MOEA/D-PRE
ZDT1	4.5121 ($4.35e-2$)	4.4507 ($8.75e-2$)	4.2776 ($1.88e-1$)	4.2303 ($9.67e-2$)	4.3990 ($2.66e-2$)	2.1604 ($4.50e-1$)	4.1497 ($2.44e-1$)
ZDT2	4.5090 ($5.57e-2$)	4.4241 ($1.32e-1$)	4.4231 ($1.58e-1$)	4.2033 ($1.08e-1$)	3.8722 ($2.83e-1$)	1.8046 ($9.27e-2$)	4.1300 ($1.96e-1$)
ZDT3	3.7582 ($2.88e-2$)	3.7009 ($2.06e-2$)	3.5993 ($1.2e-1$)	3.2484 ($2.52e-1$)	3.0962 ($2.81e-1$)	2.5306 ($4.11e-1$)	3.4078 ($3.56e-1$)
ZDT4	5.0791 ($1.71e-1$)	5.0525 ($1.69e-1$)	5.2728 ($9.86e-3$)	4.0871 ($2.77e-2$)	5.0714 ($8.40e-3$)	1.5734 ($6.56e-1$)	5.1849 ($2.40e-2$)
ZDT6	4.0303 ($2.57e-2$)	3.9785 ($5.70e-2$)	3.8749 ($1.12e-1$)	3.9565 ($1.17e-1$)	2.7656 ($5.55e-1$)	1.4462 ($1.25e+0$)	3.9088 ($5.65e-1$)
DTLZ1 ₃	8.3033 ($1.10e-2$)	8.0683 ($6.85e-2$)	5.4812 ($0.00e+0$)	8.1306 ($8.30e-1$)	8.1918 ($4.65e-3$)	7.8932 ($6.79e-3$)	8.2689 ($1.31e-1$)
DTLZ2 ₃	9.2300 ($1.59e-1$)	9.0270 ($5.33e-2$)	9.1524 ($7.71e-2$)	9.1859 ($1.18e-1$)	9.6399 ($6.57e-4$)	9.1626 ($8.06e-3$)	9.8589 ($7.14e-2$)
DTLZ5 ₃	7.4794 ($3.86e-2$)	7.1209 ($9.84e-2$)	7.0461 ($6.16e-2$)	6.9673 ($1.12e-1$)	7.3971 ($3.59e-2$)	7.3473 ($1.00e-2$)	7.4601 ($4.93e-1$)
DTLZ1 ₅	51.8760 ($1.50e+0$)	50.5049 ($1.38e+0$)	—	31.9832 ($14.14e+0$)	46.7584 ($1.06e-1$)	44.9438 ($6.13e-2$)	47.9684 ($6.82e-1$)
DTLZ2 ₅	29.4583 ($1.10e+0$)	28.6195 ($5.47e-2$)	4.3340 ($3.19e+0$)	29.2152 ($7.47e-1$)	31.0704 ($1.37e+0$)	29.2568 ($1.41e-1$)	30.5327 ($1.41e+0$)
DTLZ1 ₁₀	1486.5990 ($109.92e+0$)	1412.7149 ($119.32e+0$)	0.0 (0.0)	6.6096 ($11.112e+0$)	1342.6744 ($8.24e+0$)	1260.3203 ($11.03e+0$)	1303.7348 ($25.98e+0$)
DTLZ2 ₁₀	729.1487 ($40.01e+0$)	704.7066 ($2.57e+0$)	571.4168 ($116.75e+0$)	7.6764 ($12.83e+0$)	664.9139 ($5.08e+0$)	594.3837 ($24.46e+0$)	445.8092 ($93.27e+0$)
Welded	14.0481 ($3.769e-01$)	13.9529 ($1.94e-3$)	4.0074 ($8.37e-05$)	4.0077 ($5.07e-08$)	13.9654 ($6.05e-4$)	4.1616 ($5.72e-3$)	9.1082 ($11.32e+0$)
CAR	10.0671 ($8.720e-01$)	8.2522 ($8.28e-1$)	1.0275 ($5.07e+0$)	2.2254 ($1.52e+0$)	9.9254 ($5.05e-2$)	6.5282 ($5.96e+0$)	8.7214 ($4.72e-1$)

the smallest the R-IGD values are, the better the performance of the tested algorithm is. Additional details can be found in Li et al. [55].

5.7. Performance Comparison with Other Preference-Based EMO Algorithms. We compare RS-KKTPM with six EMO preference algorithms, including R-NSGA-II, g-NSGA-II, r-NSGAII, R-NSGA-III, WV-MOEA-P, and MOEA/D-PRE, to verify RS-KKTPM performance. We determine the parameters of the mentioned algorithms in advance to

approximate a similar ROI and make the experimental findings comparable. The parameters utilized in the comparative study are summarized as follows:

- (i) Reproduction operators: in all simulations, crossover probability = 0.9, mutation probability = $1/n$, distribution index for SBX operator = 10, and distribution index for polynomial mutation operator = 20.
- (ii) Number of evaluations, population size, and RP coordinate setting: different parameters for a set of different test instances are displayed in Table 4.

TABLE 6: Findings of mean and standard deviation for R-IGD values utilizing several preference-based EMO algorithms.

Problem	RS-KKTPM	R-NSGA-II	g-NSGA-II	r-NSGA-II	R-NSGA-III	WV-MOEA-P	MOEA/D-PRE
ZDT1	0.0152 (6.88e-3)	0.0275 (1.05e-2)	0.0677 (3.64e-2)	0.0705 (2.05e-2)	0.0339 (6.90e-3)	0.88289 (2.40e-1)	0.0990 (5.68e-2)
ZDT2	0.0159 (1.11e-2)	0.0311 (3.55e-2)	0.0431 (2.91e-2)	0.0776 (1.76e-2)	0.1770 (8.46e-2)	1.0425 (4.92e-2)	0.1082 (5.22e-2)
ZDT3	0.0155 (8.17e-3)	0.0198 (7.79e-3)	0.0357 (3.01e-2)	0.1510 (9.05e-2)	0.2085 (1.22e-1)	0.4436 (1.85e-1)	0.1103 (1.28e-2)
ZDT4	0.0366 (3.62e-2)	0.0409 (3.72e-2)	0.0056 (1.83e-3)	0.2972 (6.56e-3)	0.0236 (1.73e-3)	1.1441 (4.00e-1)	0.0095 (1.95e-3)
ZDT6	0.0305 (6.01e-3)	0.0468 (1.10e-2)	0.0728 (3.27e-2)	0.0555 (2.43e-2)	0.4645 (2.22e-1)	1.4101 (9.86e-1)	0.0986 (1.80e-1)
DTLZ1 ₃	0.0805 (4.54e-3)	0.1003 (5.68e-3)	0.4525 (0.00e+0)	0.0959 (9.95e-2)	0.0814 (1.40e-3)	0.1091 (8.56e-3)	0.0830 (1.56e-2)
DTLZ2 ₃	0.0598 (9.62e-3)	0.0673 (3.50e-3)	0.0318 (6.197e-3)	0.0597 (1.04e-2)	0.0210 (6.89e-5)	0.0532 (3.19e-4)	0.0313 (5.05e-3)
DTLZ5 ₃	0.0834 (1.39e-3)	0.1011 (5.32e-3)	0.1254 (4.12e-3)	0.1144 (8.83e-3)	0.0865 (1.80e-3)	0.0903 (7.70e-4)	0.0857 (3.97e-2)
DTLZ1 ₅	0.1125 (1.61e-2)	0.1507 (1.91e-2)	—	0.6338 (6.84e-1)	0.1425 (1.02e-3)	0.18415 (1.20e-3)	0.16923 (8.06e-3)
DTLZ2 ₅	0.1388 (1.79e-2)	0.1536 (7.06e-4)	1.4854 (3.70e-1)	0.1408 (8.11e-2)	0.0930 (2.94e-2)	0.12086 (3.50e-3)	0.1453 (1.86e-2)
DTLZ1 ₁₀	0.7036 (4.60e-2)	0.7324 (6.14e-2)	447.2251 (222.61e+0)	4.6381 (1.78e+0)	0.7203 (9.92e-3)	0.75856 (4.06e-3)	0.7697 (1.23e-2)
DTLZ2 ₁₀	0.2621 (1.52e-2)	0.3456 (2.76e-3)	1.0850 (1.26e-1)	3.6118 (8.26e-1)	0.30169 (1.30e-2)	0.4210 (5.62e-2)	0.4701 (7.94e-2)
Welded	0.9001 (2.62e-1)	0.9402 (5.63e-1)	4.400 (1.02e+0)	3.2354 (1.26e+0)	0.9358 (3.02e-5)	1.912 (2.60e+0)	1.0105 (3.94e-1)
CAR	1.5211 (5.21e-1)	1.9402 (7.53e-1)	5.2504 (22.02e+0)	3.2356 (26.28e+0)	1.5564 (6.02e-1)	2.0502 (2.92e-1)	1.8502 (5.65e-1)

- (iii) Number of runs: it is 21 for all algorithms on all test problems.
- (iv) Size of the preferred region: it is 0.1 for all algorithms on all test problems.
- (v) Parameters in r-NSGAII: the weight vector w was set as $(1/M, 1/M, \dots)$.

As mentioned earlier, S-KKTPM requires the gradient of all objective and constraint functions. Then, algebraic calculations are performed to compute the theoretical closeness of \mathbf{x} to the true optimal solution. For MOPs, S-KKTPM calculates the closeness metric from a specific PO point. In this article, the introduced RS-KKTPM approach is then compared with six EMO approaches. For a fair comparison between all algorithms, we used equal function evaluations in the comparison.

Compared to the number of function evaluations required for a solution evaluation, the savings reported for the S-KKTPM calculation may not be significant because the evaluation of S-KKTPM for all solutions is an additional computational expense and requires more computation. Thus, once gradients are computed for a real-world problem, the computational time needed for the S-KKTPM optimization procedure would make a small addition to the overall computational time. In the meantime, S-KKTPM helps improve convergence and can differentiate between different non-dominated solutions that are not applicable by using the Euclidean distance or any other evolutionary algorithm.

Tables 5 and 6 display the mean and standard deviation of R-HV and R-IGD values, respectively. The best mean of R-HV and R-IGD metrics is highlighted in bold in Tables 5 and 6.

According to R-HV metric, the ROI is approximated by the introduced RS-KKTPM algorithm in a better way than other algorithms for all the examined problems except the ZDT4, DTLZ2₃, and DTLZ2₅ test cases (see Table 5). We obtain almost the same results according to the R-IGD metric, as shown in Table 6. Based on the R-HV values and the R-IGD values, RS-KKTPM demonstrates better distribution and convergence than other algorithms. The practical findings on the 14 benchmark test problems illustrate that the RS-KKTPM approach outperforms the other approaches used in 11 of 14 comparisons.

6. Conclusions

In this study, the RS-KKTPM preference-based EMO algorithm is proposed. It is an expansion of the R-NSGA-II method, where the Euclidean distance metric is replaced by the S-KKTPM metric. The following are the properties of this new algorithm:

- (i) The RS-KKTPM can obtain the ROI at any specific position of the RP (in the feasible area, on/near the PF, and the infeasible area).
- (ii) The range of each obtained ROI can be controlled by adjusting the interest radius size of each ROI separately and in a single simulation run.

- (iii) The RS-KKTPM algorithm, given herein, improves the quality of the PF approximation and allows a uniform distribution of the approximating objective vectors.
- (iv) The performance of RS-KKTPM is better than that of R-NSGA-II, g-NSGA-II, r-NSGAI, R-NSGA-III, WV-MOEA-P, and MOEA/D-PRE on most multi and many-objective problems.

The direction of future research focuses on using the S-KKTPM metric to improve the performance of other EMO optimization algorithms by reference direction approaches, such as MOEA/D and NSGA-III. These approaches can also be utilized to solve engineering design problems and highly complex problems.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] J. Liang, W. Xu, C. Yue et al., "Multimodal multiobjective optimization with differential evolution," *Swarm and Evolutionary Computation*, vol. 44, pp. 1028–1059, 2019.
- [2] X. S. Yang, *Optimization Techniques and Applications with Examples*, John Wiley & Sons, New York, NY, USA, 2018.
- [3] K. Deb, P. Amrit, A. Sameer, and M. Tamt, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [4] S. Jiang and S. Yang, "A strength pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 329–346, 2017.
- [5] V. Ho-Huu, S. Hartjes, H. G. Visser, and R. Curran, "An improved moea/d algorithm for bi-objective optimization problems with complex pareto fronts and its application to structural optimization," *Expert Systems with Applications*, vol. 92, pp. 430–446, 2018.
- [6] E. Filatovas, O. Kurasova, J. L. Redondo, and J. Fernandez, "A reference point-based evolutionary algorithm for approximating regions of interest in multiobjective problems," *Top*, vol. 28, no. 2, pp. 402–423, 2020.
- [7] Z. Xiong, J. Yang, Z. Hu, Z. Zhao, and X. Wang, "Evolutionary many-objective optimization algorithm based on angle and clustering," *Applied Intelligence*, vol. 51, no. 4, pp. 2045–2062, 2021.
- [8] J. Zou, Y. He, J. Zheng et al., "Hierarchical preference algorithm based on decomposition multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 60, Article ID 100771, 2021.
- [9] J. Zou, Q. Yang, S. Yang, and J. Zheng, "Ra-dominance: a new dominance relationship for preference-based evolutionary multiobjective optimization," *Applied Soft Computing*, vol. 90, Article ID 106192, 2020.
- [10] Q. Li, J. Zou, S. Yang, J. Zheng, and G. Ruan, "A predictive strategy based on special points for evolutionary dynamic multi-objective optimization," *Soft Computing*, vol. 23, no. 11, pp. 3723–3739, 2019.
- [11] M. Caramia and P. Dell'Olmo, "Multi-objective optimization," in *Multi-objective Management in Freight Logistics*, pp. 21–51, Springer, Berlin, Germany, 2020.
- [12] R. Tang, K. Li, W. Ding, Y. Wang, H. Zhou, and G. Fu, "Reference point based multi-objective optimization of reservoir operation: a comparison of three algorithms," *Water Resources Management*, vol. 34, no. 3, pp. 1005–1020, 2020.
- [13] G. Beruvides, F. Castaño, R. Quiza, and R. E. Haber, "Surface roughness modeling and optimization of tungsten-copper alloys in micro-milling processes," *Measurement*, vol. 86, pp. 246–252, 2016.
- [14] Y. J. Cruz, M. Rivas, R. Quiza, A. Villalonga, R. E. Haber, and G. Beruvides, "Ensemble of convolutional neural networks based on an evolutionary algorithm applied to an industrial welding process," *Computers in Industry*, vol. 133, Article ID 103530, 2021.
- [15] R. Landa, G. Lárraga, and G. Toscano, "Use of a goal-constraint-based approach for finding the region of interest in multi-objective problems," *Journal of Heuristics*, vol. 25, no. 1, pp. 107–139, 2019.
- [16] F. Wang, Y. Li, H. Zhang, T. Hu, and X.-L. Shen, "An adaptive weight vector guided evolutionary algorithm for preference-based multi-objective optimization," *Swarm and Evolutionary Computation*, vol. 49, pp. 220–233, 2019.
- [17] J. Hu, G. Yu, J. Zheng, and J. Zou, "A preference-based multi-objective evolutionary algorithm using preference selection radius," *Soft Computing*, vol. 21, no. 17, pp. 5025–5051, 2017.
- [18] A. Villalonga, E. Negri, G. Biscardo et al., "A decision-making framework for dynamic scheduling of cyber-physical production systems based on digital twins," *Annual Reviews in Control*, vol. 51, pp. 357–373, 2021.
- [19] R. Rădulescu, P. Mannion, D. M. Roijers, and A. Nowé, "Multi-objective multi-agent decision making: a utility-based analysis and survey," *Autonomous Agents and Multi-Agent Systems*, vol. 34, no. 1, pp. 10–52, 2020.
- [20] W. Yang, L. Chen, Y. Wang, and M. Zhang, "A reference points and intuitionistic fuzzy dominance based particle swarm algorithm for multi/many-objective optimization," *Applied Intelligence*, vol. 50, no. 4, pp. 1133–1154, 2020.
- [21] R. H. Stewart, T. S. Palmer, and B. DuPont, "A survey of multi-objective optimization methods and their applications for nuclear scientists and engineers," *Progress in Nuclear Energy*, vol. 138, Article ID 103830, 2021.
- [22] B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu, "Interactive multiobjective optimization: a review of the state-of-the-art," *IEEE Access*, vol. 6, pp. 41256–41279, 2018.
- [23] M. K. Tomczyk and M. Kadziński, "Decomposition-based interactive evolutionary algorithm for multiple objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 320–334, 2020.
- [24] A. J. Nebro, A. B. Ruiz, C. Barba-González, J. García-Nieto, M. Luque, and J. F. Aldana-Montes, "Indm2: interactive dynamic multi-objective decision making using evolutionary algorithms," *Swarm and Evolutionary Computation*, vol. 40, pp. 184–195, 2018.
- [25] S. Mirjalili and J. S. Dong, *Multi-objective Optimization Using Artificial Intelligence Techniques*, Springer, Berlin, Germany, 2020.
- [26] K. Deb and J. Sundar, "Reference point based multi-objective optimization using evolutionary algorithms," in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pp. 635–642, New York, USA, July 2006.

- [27] J. Molina, L. V. Santana, A. G. Hernández-Díaz, C. A. Coello Coello, and R. Caballero, “g-dominance: reference point based dominance for multiobjective metaheuristics,” *European Journal of Operational Research*, vol. 197, no. 2, pp. 685–692, 2009.
- [28] L. Ben Said, S. Bechikh, and K. Ghédira, “The r-dominance: a new dominance relation for interactive evolutionary multi-criteria decision making,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 801–818, 2010.
- [29] A. B. Ruiz, R. Saborido, and M. Luque, “A preference-based evolutionary algorithm for multiobjective optimization: the weighting achievement scalarizing function genetic algorithm,” *Journal of Global Optimization*, vol. 62, no. 1, pp. 101–129, 2015.
- [30] G. Yu, J. Zheng, R. Shen, and M. Li, “Decomposing the user-preference in multiobjective optimization,” *Soft Computing*, vol. 20, no. 10, pp. 4005–4021, 2016.
- [31] M. Abouhawwash, “Hybrid evolutionary multi-objective optimization algorithm for helping multi-criterion decision makers,” *International Journal of Management Science and Engineering Management*, vol. 16, no. 2, pp. 94–106, 2021.
- [32] M. Abouhawwash and K. Deb, “Reference point based evolutionary multi-objective optimization algorithms with convergence properties using kktpm and asf metrics,” *Journal of Heuristics*, vol. 27, no. 4, pp. 575–614, 2021.
- [33] E. Filatovas, A. Lančinskas, O. Kurasova, and J. Žilinskas, “A preference-based multi-objective evolutionary algorithm r-nsga-ii with stochastic local search,” *Central European Journal of Operations Research*, vol. 25, no. 4, pp. 859–878, 2017.
- [34] Z. Hou, S. Yang, J. Zou, J. Zheng, G. Yu, and G. Ruan, “A performance indicator for reference-point-based multi-objective evolutionary optimization,” in *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1571–1578, IEEE, Bangalore, India, November 2018.
- [35] H. Wang, M. Olhofer, and Y. Jin, “A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges,” *Complex & Intelligent Systems*, vol. 3, no. 4, pp. 233–245, 2017.
- [36] M. Abouhawwash, M. Jameel, and K. Deb, “A smooth proximity measure for optimality in multi-objective optimization using benson’s method,” *Computers & Operations Research*, vol. 117, Article ID 104900, 2020.
- [37] K. Deb and M. Abouhawwash, “An optimality theory-based proximity measure for set-based multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 515–528, 2016.
- [38] B. Qu, C. Li, J. Liang, L. Yan, K. Yu, and Y. Zhu, “A self-organized speciation based multi-objective particle swarm optimizer for multimodal multi-objective problems,” *Applied Soft Computing*, vol. 86, Article ID 105886, 2020.
- [39] G. Giorgi, B. Jiménez, and V. Novo, “Approximate karush–kuhn–tucker condition in multiobjective optimization,” *Journal of Optimization Theory and Applications*, vol. 171, no. 1, pp. 70–89, 2016.
- [40] Y. Zhou-Kangas and K. Miettinen, “Decision making in multiobjective optimization problems under uncertainty: balancing between robustness and quality,” *Spectrum*, vol. 41, no. 2, pp. 391–413, 2019.
- [41] D. Ghosh, A. Singh, K. Shukla, and K. Manchanda, “Extended karush-kuhn-tucker condition for constrained interval optimization problems and its application in support vector machines,” *Information Sciences*, vol. 504, pp. 276–292, 2019.
- [42] J. Dutta, K. Deb, R. Tulshyan, and R. Arora, “Approximate kkt points and a proximity measure for termination,” *Journal of Global Optimization*, vol. 56, no. 4, pp. 1463–1499, 2013.
- [43] M. Abouhawwash and M. A. Jameel, “Evolutionary multi-objective optimization using benson’s karush-kuhn-tucker proximity measure,” in *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*, pp. 27–38, Springer, Berlin, Germany, March 2019.
- [44] G. Eichfelder and L. Warnow, “Proximity measures based on kkt points for constrained multi-objective optimization,” *Journal of Global Optimization*, vol. 80, no. 1, pp. 63–86, 2021.
- [45] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multi-objective evolutionary algorithms: empirical results,” *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [46] A. Tharwat, E. H. Houssein, M. M. Ahmed, A. E. Hassanien, and T. Gabel, “Mogoa algorithm for constrained and unconstrained multi-objective optimization problems,” *Applied Intelligence*, vol. 48, no. 8, pp. 2268–2283, 2018.
- [47] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multiobjective optimization,” in *Evolutionary Multiobjective Optimization*, pp. 105–145, Springer, Berlin, Germany, 2005.
- [48] K. Zhong, G. Zhou, W. Deng, Y. Zhou, and Q. Luo, “Momp: multi-objective marine predator algorithm,” *Computer Methods in Applied Mechanics and Engineering*, vol. 385, Article ID 114029, 2021.
- [49] G. Dhiman, M. Soni, H. M. Pandey, A. Slowik, and H. Kaur, “A novel hybrid hypervolume indicator and reference vector adaptation strategies based evolutionary algorithm for many-objective optimization,” *Engineering with Computers*, vol. 37, no. 4, pp. 3017–3035, 2021.
- [50] Y. Vesikar, K. Deb, and J. Blank, “Reference point based nsga-iii for preferred solutions,” in *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1587–1594, IEEE, Bangalore, India, November 2018.
- [51] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, “Platemo: a matlab platform for evolutionary multi-objective optimization [educational forum],” *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.
- [52] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, “The arithmetic optimization algorithm,” *Computer Methods in Applied Mechanics and Engineering*, vol. 376, Article ID 113609, 2021.
- [53] V. R. d. Carvalho, K. Larson, A. A. F. Brandão, and J. S. Sichman, “Applying social choice theory to solve engineering multi-objective optimization problems,” *Journal of Control, Automation and Electrical Systems*, vol. 31, no. 1, pp. 119–128, 2020.
- [54] A. H. Halim, I. Ismail, and S. Das, “Performance assessment of the metaheuristic optimization algorithms: an exhaustive review,” *Artificial Intelligence Review*, vol. 54, no. 3, pp. 2323–2409, 2021.
- [55] K. Li, K. Deb, and X. Yao, “R-metric: evaluating the performance of preference-based evolutionary multiobjective optimization using reference points,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 6, pp. 821–835, 2018.