A Territory Defining Multiobjective Evolutionary Algorithms and Preference Incorporation

İbrahim Karahan and Murat Köksalan

***Abstract*—We have developed a steady-state elitist evolutionary algorithm to approximate the Pareto-optimal frontiers of multiobjective decision making problems. The algorithms define a territory around each individual to prevent crowding in any region. This maintains diversity while facilitating the fast execution of the algorithm. We conducted extensive experiments on a variety of test problems and demonstrated that our algorithm performs well against the leading multiobjective evolutionary algorithms. We also developed a mechanism to incorporate preference information in order to focus on the regions that are appealing to the decision maker. Our experiments show that the algorithm approximates the Pareto-optimal solutions in the desired region very well when we incorporate the preference information.**

***Index Terms*—Crowding prevention, evolutionary algorithms, guidance, multiobjective optimization, preference incorporation.**

I. Introduction

MANY DECISION making situations require considering several objectives simultaneously. Many mathematical programming-based approaches have been developed to handle multiple objectives [30]. Although they are effective in finding nondominated solutions, they usually do not generate multiple solutions in a single run. In addition, they often have significant computational requirements, especially when the problems are combinatorial. Multiobjective evolutionary algorithms (MOEAs) overcome some of these shortcomings. These algorithms imitate the genetic evolution in nature and apply it to optimization. They work with populations of solutions and progress with their genetic evolution. Consequently, they find multiple approximately nondominated solutions in a reasonable time in a single run.

These qualifications have made MOEAs popular among researchers in the past two decades. Many successful MOEAs, such as Pareto-archived evolution strategy (PAES) [24], Pareto envelope-based selection algorithm (PESA)-II [7], nondominated sorting genetic algorithm (NSGA)-II [15], strength Pareto evolutionary algorithm (SPEA2) [35], indicator-based evolutionary algorithm (IBEA) [34], ϵ-MOEA [14], S metric selection evolutionary multiobjective optimization algorithm [1], and favorable weight-based evolutionary algorithm (FWEA) [29] are proposed in the literature. The review papers by [19], [31], [4], [9], and [33] examine various MOEA approaches and point out their increasing popularity among researchers. The books by [10] and [6] cover many aspects of evolutionary algorithms in multiobjective optimization. Coello maintains a list of references to publications in the multiobjective evolutionary optimization area at http://www.lania.mx/~ccoello/EMOO/EMOObib.html.

One of the goals of MOEAs is to minimize the proximity of solutions to the true Pareto-optimal frontier. While achieving this, they have to maintain the diversity among the solutions in the population so that a good approximation of the Pareto-optimal frontier can be obtained with a limited number of solutions. Hence, the performance of an algorithm not only depends on its convergence but also on its diversity. The computational burden of achieving these is another concern in designing the algorithm.

Although most of the MOEAs try to generate solutions that approximate the entire efficient frontier, this does not help decision making as [5] claims. Typically, the decision maker (DM) is interested in only a few of the solutions. If the preferences of the DM can be elicited before or during optimization, this information can be exploited to guide the algorithm to those regions that are appealing to the DM. Consequently, much of the computational effort can be saved and the desired regions can be explored in more detail.

This third goal of converging to the interesting regions of the Pareto-optimal frontier has become popular among MOEA researchers in recent years. The review papers of [5], [8], and [28] stress the importance of this new goal and examine the methods from this perspective. There are several studies in the literature that try to guide the search in MOEAs to the regions that are desirable to the DM.

Fonseca and Fleming [20] propose the preferability operator. In this method, the DM sets goals and priority levels for each objective. Taking this preference information into account, ranks of the individuals are assigned using a modified dominance scheme based on the preferability operator.

Branke *et al*. [3] propose a method that asks the decision maker to specify his/her trade-offs between each pair of objectives. Then they construct *minimal* and *maximal* utility functions. These utility functions are used for modifying the dominance scheme based on the DM's preferences.

Phelps and Köksalan [27] develop an interactive MOEA for multiobjective combinatorial optimization problems (MOCO).

Manuscript received December 15, 2008; revised July 22, 2009. Date of publication April 22, 2010; date of current version July 30, 2010.

İ. Karahan is with the University of Illinois, Urbana-Champaign, IL 61801 USA (e-mail: karahan1@illinois.edu).

M. Köksalan is with the Department of Industrial Engineering, Middle East Technical University, Ankara 06531, Turkey (e-mail: koksalan@ie.metu.edu.tr).

Digital Object Identifier 10.1109/TEVC.2009.2033586

This algorithm interacts with the decision maker during the run to guide the search toward the preferred regions.

Tan *et al*. [32] propose a modified domination scheme that incorporates flexible goal and priority information on each objective component, as well as hard and soft priority and constraint specifications. Their method allows the use of logical connectives such as "AND" and "OR" to join different preference specifications and guides the search toward multiple regions of interest to the DM.

The method of [16] seeks to find a set of solutions close to one or more reference points that are specified by the DM. The idea is implemented by modifying the crowding distance operator of NSGA-II.

Deb and Kumar [13] incorporate the ideas from the interactive approach of [25] into NSGA-II, and make use of *achievement scalarizing functions* to guide the search toward preferred regions.

Köksalan and Phelps [23] develop an evolutionary metaheuristic for approximating preference-nondominated solutions for the MOCO problems to find solutions that are appealing to the DM. The algorithm uses partial preference information provided by the DM, which is gathered through qualitative statements. These statements are then used for restricting the weight space.

We propose a new MOEA, the territory defining evolutionary algorithm (TDEA), that aims to converge to the true Pareto-optimal frontier while maintaining a uniform diversity among solutions. The algorithm utilizes its territory defining property to keep the solutions in the population well-dispersed. This eliminates the need for an explicit diversity preservation operator and provides computational advantage.

In TDEA, we design a mechanism that can be adapted to situations where some preference information is available. We develop a preference-based modification of TDEA based on the territory defining property, the preference-based TDEA (prTDEA), to achieve this goal. prTDEA utilizes preference information to guide the search to the interesting portions of the Pareto-optimal frontier by modifying the territory sizes of the population members in those regions. Hence, it allows a better approximation of such regions while reducing computational effort.

In Section II, we present the details of the algorithm and then compare the performance of the algorithm against well-known MOEAs in Section III. In Section IV, we describe preference incorporation. After giving the results of computational experiments in Section V, we conclude in Section VI.

II. Development of the Algorithm

TDEA is a steady-state algorithm that maintains two populations. The archive population consists of individuals that are nondominated relative to the population at hand and the regular population contains both dominated and nondominated individuals. When updating the archive population, we define a *territory* around the individual closest to the offspring and reject the offspring if it violates the territory. The territory defining property of TDEA eliminates the need for an explicit diversity operator, resulting in a fast operation while always keeping a diverse set of individuals in the archive population.

Territory preservation has a similarity to the idea of [14], where they divide the objective space into hyperboxes of equal size. Each hyberbox can contain at most a single individual and hence prevents crowding. However, unlike territory preservation, the locations of hyperboxes are independent of the locations of the solutions and are fixed before the optimization. This makes ϵ-MOEA more prone to losing individuals toward the extremes of the Pareto-optimal frontier. These differences allow TDEA to reach the extreme regions better. We illustrate this in Fig. 1 using identical hyperbox and territory sizes. In ϵ-MOEA, the individuals close to the extremes ϵ-dominate the remaining extreme regions of the objectives. Hence, no other individuals located in such regions can enter the archive and they are lost. On the other hand, territories in TDEA only disallow other individuals in a limited region around the individual. Thus, individuals toward the extremes do not experience a handicap as in ϵ-MOEA.

In another paper, Goh and Tan [21] propose an NP-dominance scheme, where some of the nondominated space is also considered as dominated. Here, the authors propose the opposite of territory definition. The space outside the territory of an individual is considered as dominated, but another individual is allowed inside the territory.

A. Definitions

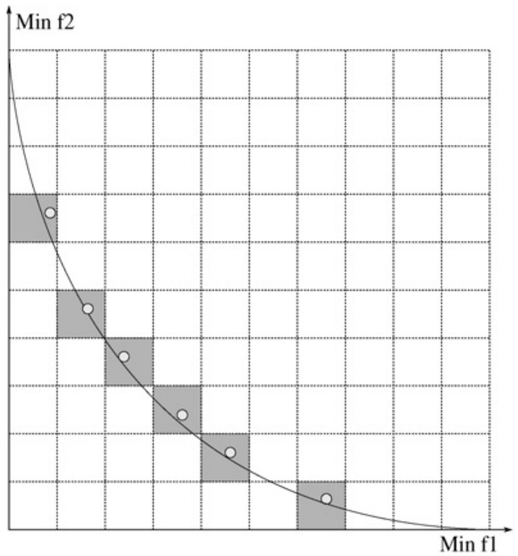
Consider a problem with m objectives to be maximized. Let $X_p = \left(f_{p1}, f_{p2}, \ldots, f_{pm}\right)$ denote solution p having an objective function value of f_{pj} in objective j. Solution X_i is said to dominate solution X_p if $f_{ij} \geq f_{pj}$ for $j = 1, 2, \ldots, m$ and $f_{ij} > f_{pj}$ for at least one j. If there exists no X_i dominating X_p, then X_p is said to be nondominated or Pareto-optimal.

We define the territory of X_p as the region within a distance τ of X_p in each objective among the regions that neither dominate nor is dominated by X_p. Mathematically, the territory of X_p is defined as the hypervolume $V_p = \{x \in \Re^m \mid |f_{pj} - f_j| < \tau$ for $j = 1, 2, \ldots, m$ AND $f_{pj} - f_j < 0$ for at least one j AND $f_{pj} - f_j > 0$ for at least one $j\}$. The territory of an individual in a 2-D space is illustrated in Fig. 2.

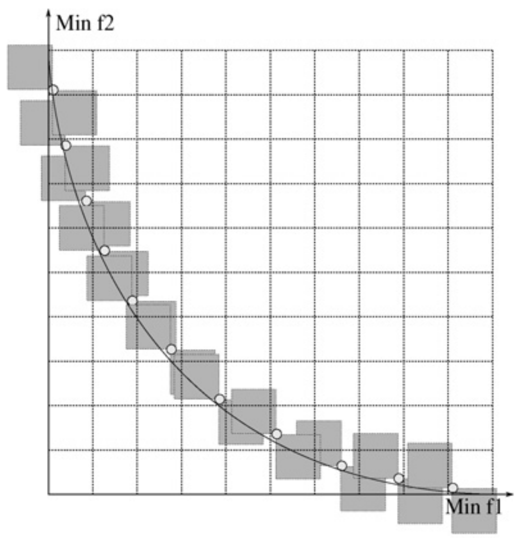
B. Overview

The *regular population*, P, is formed by randomly created individuals at the beginning of the algorithm until its maximum size $\bar{N}$ is reached. It may contain both nondominated and dominated individuals. An offspring is accepted to the regular population if it is nondominated relative to the individuals in P.

The *archive*, A, consists of only nondominated individuals. It is created from the copies of the nondominated individuals of P. Although its size is not explicitly restricted by a fixed number, it depends on τ that defines the size of the territory of an individual. Only the individuals that are accepted to the regular population are eligible to be evaluated to enter the archive. The evaluation is done using the territory defining property of the algorithm, which will be described later.



(a)



(b)

Fig. 1. Hyperbox versus territory. (a) ϵ-MOEA. (b) TDEA.

We should note that separating two populations is a design choice. It is also possible to implement the algorithm combining the two populations into a single population.

We first give a general outline of the algorithm.

1) Ask the user to specify $\bar{N}$, τ and maximum number of iterations T. Set iteration count $t = 0$. Create $\bar{N}$ random individuals to form the initial regular population $P(0)$. Copy the nondominated individuals of $P(0)$ into $A(0)$ to form the initial archive population.
2) Set $t \leftarrow t + 1$. Choose a parent from each of the populations $P(t)$ and $A(t)$. Recombine parents to create a new offspring and apply mutation.
3) Check whether the offspring satisfies the acceptance condition into $P(t)$. If it does, insert it into $P(t)$ and go to the next step. Otherwise, go to Step 5.
4) Check whether the offspring satisfies the acceptance condition into $A(t)$. If it does, insert into $A(t)$.

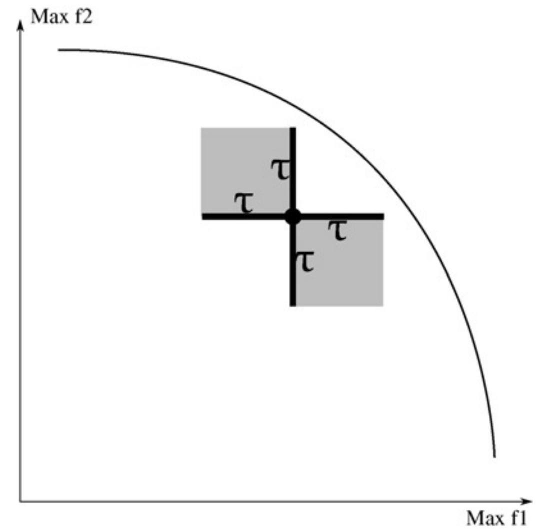


Fig. 2. Illustration of territory in 2-D space.

5) If $t = T$, stop and report the archive population. Otherwise go to Step 2.

C. Parent Selection

In this paper, the experiments are performed by choosing one individual from each population for recombination. The selection scheme is different for the two populations. In the regular population, the binary tournament selection is used. Two individuals s_1 and s_2 are randomly picked from the regular population and parent p_1 is determined according to the following procedure.

1) Test for dominance between s_1 and s_2. If one dominates the other, denote the dominating individual as the first parent p_1.
2) If there is no dominance relation, then select randomly among s_1 and s_2.

The second parent p_2 is selected from the archive population. Since all individuals in the archive are nondominated relative to each other, we randomly choose one individual as p_2.

It should be stressed that the parent selection scheme used in this paper is not the only selection scheme suitable for TDEA. Any other selection scheme suitable for the needs can be employed to select the parents to reproduce. In addition, tournament size is not strict. If a larger selection pressure is desired, a larger tournament size can be set. Also, parents that compete in the tournament selection can be chosen from any population freely.

D. Scaling

In multiobjective optimization problems, the range and the scale of objectives may vary considerably. These differences may cause MOEAs to be biased. To address this issue, scaling techniques have been proposed in the literature (see [30, pp. 200–202, Ch. 8] for details). The algorithms that require some sort of cohesion of the objectives need to properly scale different objectives. TDEA implicitly requires a cohesion of the objectives through its territory mechanism. In order to

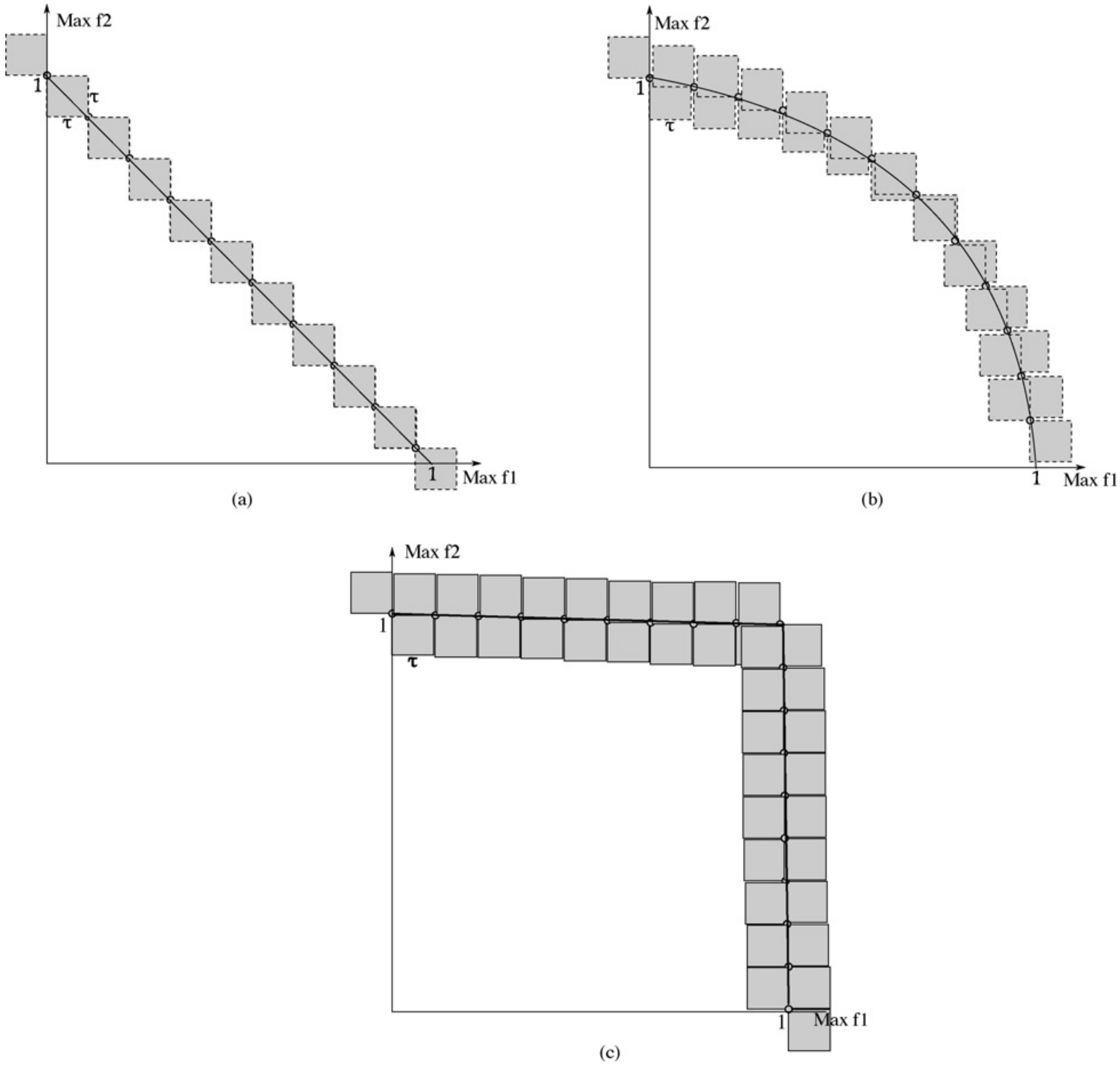


Fig. 3. Possible placement of individuals on different Pareto-optimal frontiers. (a) Linear. (b) Curve. (c) Nearly rectangular.

define a meaningful τ value applicable to all objectives, a scaling mechanism is needed.

In TDEA, we use the idea of [29] for scaling the objectives. They scale all objective values into [0, 1] interval. They treat the objective values between the ideal (f^*) and the nadir point (f^n) differently from those beyond the nadir point. While they use linear scaling for the former interval, they scale the values beyond the nadir point using a sigmoid function. This way, the values in the nondominated range are scaled into a large portion of [0, 1]. The remaining values are scaled into a narrow interval since they are not as important as the nondominated range. If only the nondominated range were scaled in the [0, 1] interval, the territory defining property might not work well for values beyond the nadir point, because the territory sizes are set based on the nondominated range. If the nadir point is known exactly, then the scaling can be accurately done. Otherwise, the nadir can be approximately estimated.

E. Population Updates

An offspring, c, is first evaluated for acceptance into the regular population. The evaluation procedure is as follows.

1) Test c against each individual $s_i \in P(t)$ for dominance. Mark the individuals dominated by c. If c is dominated by at least one s_i, reject c. Otherwise, go to the next step.
2) Remove one of the marked individuals randomly from $P(t)$. If no individuals are marked, choose and remove an individual randomly from $P(t)$.
3) Insert c into $P(t)$ and test for acceptance in $A(t)$.

If c is rejected in the above procedure, then it is not evaluated for archive acceptance at all. The archive evaluation process consists of two stages. First, the offspring c is checked for dominance against the members of the archive. If it is not dominated by any individual in the archive, then we proceed to the second stage. The second stage begins with the removal of individuals dominated by c from the archive. After that, we determine the closest individual, s_{i^*}, to c in terms of the scaled rectilinear distance. Then, we check whether c is in the territory of s_{i^*}. If it is, we reject c. Otherwise, c is accepted.

Defining $\hat{f}_{ij}$ as the scaled value of individual i in objective j, we present the details of the procedure below.

1) Test c against each individual $s_i \in A(t)$ for dominance. Mark the individuals dominated by c. If c is dominated by at least one s_i, reject c. Otherwise, go to the next step.
2) Remove all marked individuals from $A(t)$.
3) If $A(t)$ is empty, accept and insert c into $A(t)$ and stop. Otherwise, continue to next step.
4) Calculate the rectilinear distance $d_{ci} = \sum_{j=1}^{m} \left| \hat{f}_{cj} - \hat{f}_{ij} \right|$ of c to each individual $s_i \in A(t)$.
5) Find $i^* = \text{argmin}_i(d_{ci})$, that is, the individual s_{i^*} closest to c.
6) Find the maximum scaled absolute objective difference between c and s_{i^*}. That is, find
$$\delta = \max_{j=1,2,\ldots,m} \left| \hat{f}_{cj} - \hat{f}_{i^*j} \right|. \quad (1)$$
7) If $\delta \geq \tau$, accept and insert c into $A(t)$. Otherwise, reject c.

F. Determination of τ

In TDEA, τ defines the territory size by creating a hypervolume around each individual in the objective space. Since the total nondominated hypervolume is limited, the territory size bounds the maximum number of individuals in the archive population. Reducing the τ value leads to an increased population size in general. The actual size of the nondominated hypervolume, on the other hand, depends on the number of objectives and the shape of the Pareto-optimal frontier, and it differs from problem to problem.

We illustrate nondominated hypervolumes for several hypothetical Pareto-frontier shapes in Fig. 3. We try to locate the maximum number of individuals on the given Pareto-optimal frontiers using the same τ value. It can be observed that although the same τ value is used, population sizes are different between the Pareto-frontiers of different shapes. As we go from the linear to nearly rectangular Pareto-optimal frontier, the population size increases. The linear Pareto-optimal frontier has a length of $l_l = \sqrt{2}$, and the distance between individuals can be at least $\tau\sqrt{2}$. Then, the maximum number of individuals can be found roughly by $\frac{l_l}{\tau\sqrt{2}} = \frac{1}{\tau}$. On the other hand, the length l_r of the nearly rectangular Pareto-optimal frontier is approximately 2, and the individuals are separated by a distance of almost τ. Then, the maximum number of individuals is approximately $\frac{l_r}{\tau} = \frac{2}{\tau}$, which is two times the first case. Any other curve-shaped Pareto-optimal frontier as shown in Fig. 3(b) has a population between $\frac{1}{\tau}$ and $\frac{2}{\tau}$. For two-objective problems, such bounds are useful to set the value of τ. For higher dimensions, the derivations are more involved and they are left for a future research. As the number of objectives increases, the nondominated hypervolume increases. For this reason, in general τ should be increased for problems with higher number of objectives. Territory preservation has similarities to the t-dominance idea [26] for which the authors provide upper bounds for population size for a given value of t. Those bounds may be modified to provide similar results for our approach.

G. Computational Complexity

TDEA is a steady-state multiobjective evolutionary algorithm. In each generation, only one offspring is created and evaluated to enter the populations. For an offspring c that is accepted into the regular population P, it takes $m\bar{N}$ comparisons to check that it is nondominated with respect to all individuals in P. Similarly, it takes $m\hat{N}$ comparisons to check that c is nondominated with respect to all $\hat{N}$ members of the archive population. It requires an additional $m\hat{N}$ comparisons to determine the closest individual to c in A. At the end, $m\bar{N}+2m\hat{N}$ operations are made in each generation. We neglect the operations made for crossover and mutation operators since they are not specific to the algorithm. Since the size of A is not fixed, the exact number of comparisons needed for dominance check in A is not known exactly. However, it is kept in a reasonable number by setting τ suitably. If $\hat{N} \gg \bar{N}$, then the computational complexity of TDEA is dominated by the size of the archive population. In this case, the computational complexity is $\mathcal{O}\left(m\hat{N}\right)$.

Among the MOEAs that we compare TDEA with, ϵ-MOEA has the lowest computational complexity with $\mathcal{O}\left(m\hat{N}\right)$. NSGA-II's diversity preservation mechanism requires $\mathcal{O}(N \log N)$ operations but its computational complexity is determined by the fast nondominated sorting with $\mathcal{O}\left(mN^2\right)$. The fitness assignment procedure in SPEA2 requires $\mathcal{O}\left(M^2 \log M\right)$ operations, where M is the sum of the sizes of regular and archive populations. SPEA2's run-time is dominated by its truncation operator that requires $\mathcal{O}\left(M^3\right)$ operations. Finally, IBEA's run-time complexity is $\mathcal{O}\left(N^2\right)$.

III. Computational Experiments with TDEA

In this section, we evaluate TDEA on some test problems and compare its performance against some of the well-known MOEAs, namely SPEA2, NSGA-II, IBEA, and ϵ-MOEA. We use the test problem families ZDT [10] and DTLZ [17]. We also made comparisons with PAES and PESA-II, but we do not present those since they were outperformed in all problems. We present the results for only a set of problems. For more details, see [22].

We use the real-valued representation of the decision variables. That is, each gene in a chromosome represents the value of the corresponding decision variable. For crossover, we use the simulated binary crossover operator [11] with $p_c = 1.0$ and crossover parameter $\eta_c = 20$ for all metaheuristics. The mutation operator is the polynomial mutation operator [12] with $p_{mut} = \frac{1}{\text{\# of variables}}$ and mutation parameter $\eta_{mut} = 20$.

For each problem, we run each algorithm 50 times with different seeds for the random number generator and limit each run to a prespecified maximum number of function evaluations. Initial populations are formed from randomly created individuals and population sizes are set based on

TABLE I
TEST PARAMETERS

	ZDT4	ZDT6	DTLZ1	DTLZ2
Number of objectives	2	2	3	3
Regular population size	200	200	400	400
Function evaluations	40 000	40 000	160 000	160 000
Replications	50	50	50	50
τ	0.0075	0.006	0.04	0.065
	DTLZ3	DTLZ7	DTLZ1-5D	DTLZ2-5D
Number of objectives	3	3	5	5
Regular population size	400	400	400	400
Function evaluations	160 000	160 000	320 000	320 000
Replications	50	50	50	50
τ	0.06	0.05	0.14	0.22

TABLE II
INDICATOR RESULTS FOR ZDT4

	Hypervolume		Inv. Gen. Dist.		Additive Epsilon		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	$\bar{x}_I$	s_I	Time (s)
TDEA	0.6591	0.0011	0.000016	0.000001	0.0104	0.0015	0.66
ϵ-MOEA	0.6590	0.0013	0.000018	0.000004	0.0091	0.0041	0.34
IBEA	0.0000	0.0000	0.007720	0.011070	2.6630	3.5110	2.41
NSGA2	0.6582	0.0013	0.000018	0.000001	0.0129	0.0019	5.26
SPEA2	0.6592	0.0016	0.000015	0.000002	0.0086	0.0016	17.17
True Pareto	0.6660	–	0	–	0	–	–

the number of objectives. For metaheuristics without constant population sizes (ϵ-MOEA and TDEA), we set ϵ and τ so that their final populations are approximately of the same size as the other algorithms. It is important to have approximately equal final population sizes in order to make fair comparisons between the algorithms. Further parameter settings are given in Table I.

We implement our algorithm in C++. For ϵ-MOEA, we use the source code available from the website of Kanpur Genetic Algorithms Laboratory (http://www.iitk.ac.in/kangal/codes.shtml). Both programs are built with GNU C/C++ Compiler 4.2.3. For other metaheuristics, we use jMetal framework [18] with Sun Java JDK/JRE 1.6.0.3. All computational tests in this section are made on a Pentium IV 2.8 GHz, 1 GB RAM computer running Kubuntu Linux 8.04.

For every problem, we calculate the performance of each run of each algorithm in terms of the hypervolume (H) [36], the inverted generational distance (D) [2], and additive ϵ-indicator (I) metrics [34]. H measures the total dominated hypervolume by the individuals generated by the algorithm and an inferior reference point. Hence, larger H values are more desirable. On the other hand, D measures the average distance between true efficient solutions and their closest generated individual in the population. Smaller D values are desirable. I is the minimum distance by which the individuals reported by the algorithm need to be improved in each dimension of the objective space so that the true Pareto-optimal set is weakly dominated. A smaller I value is better.

For calculating the D and I metrics, we generate approximately 100 000 well-dispersed true nondominated solutions for each problem. For calculating the H metric, we use the nadir point as the reference point. Then, we compute the sample means $\bar{x}_H, \bar{x}_D, \bar{x}_I$ and the sample standard deviations s_H, s_D, s_I of these metrics. By the central limit theorem, we assume that the sample means are normally distributed. Then, we test the following hypothesis at a 99% significance level to check whether there exists a statistically significant difference between the performances of TDEA (denoted as T) and other algorithms (denoted as C) in both metrics

$$H_0 : \mu^T_{pm} = \mu^C_{pm} \tag{2}$$

$$H_1 : \mu^T_{pm} \neq \mu^C_{pm} \tag{3}$$

where pm stands for performance metric. For every problem, we report the difference $\Delta_{pm} = \bar{x}^T_{pm} - \bar{x}^C_{pm}$ between average values obtained for TDEA and its contender for both metrics. We also present the p-value of the statistical test. The metric results of the true nondominated sets are also given as benchmarks. For visualization, we provide the best run result (as

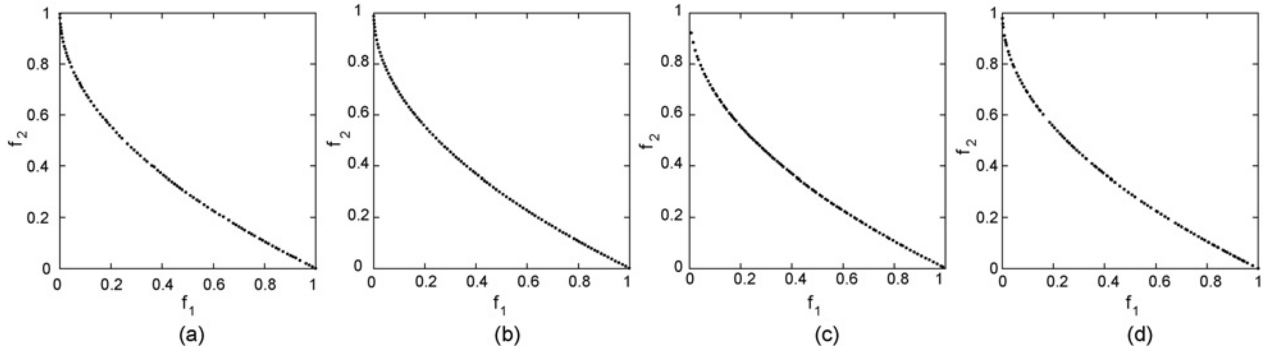


Fig. 4. ZDT4 plots. (a) TDEA. (b) SPEA2. (c) ϵ-MOEA. (d) NSGA2.

TABLE III
TEST RESULTS FOR ZDT4

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ versus $H_1 : \mu_{TDEA} \neq \mu_{Contender}$								
	Hypervolume			Inv. Gen. Dist.			Additive Epsilon		
Contender	Δ_H	p-Value	Winner	Δ_D	p-Value	Winner	Δ_I	p-Value	Winner
ϵ-MOEA	0.0001	0.625	None	−0.000002	0	TDEA	0.0013	0.035	None
IBEA	N/A	N/A	TDEA	N/A	N/A	TDEA	−2.6520	0	TDEA
NSGA2	0.0008	0.001	TDEA	−0.000002	0	TDEA	−0.0025	0	TDEA
SPEA2	−0.0001	0.661	None	0.000001	0.035	None	0.0018	0	SPEA2

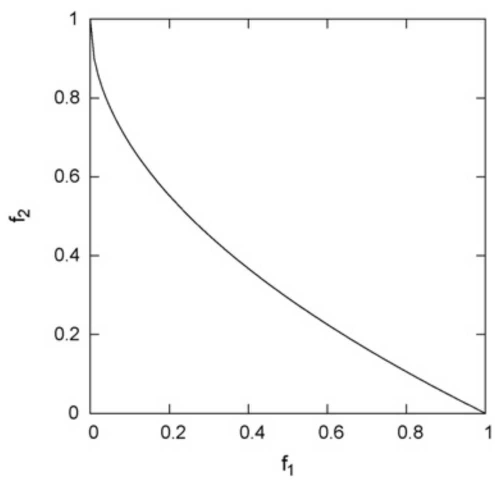


Fig. 5. Pareto-optimal frontier of ZDT4.

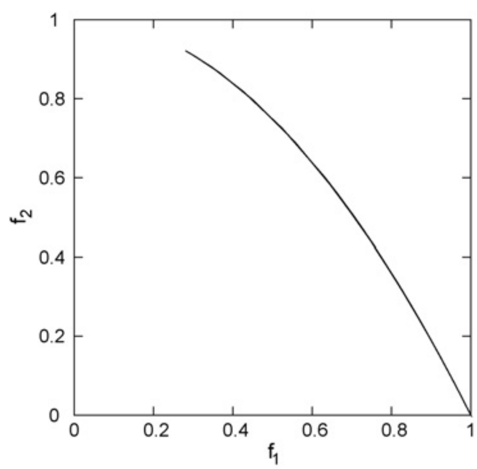


Fig. 6. Pareto-optimal frontier of ZDT6.

measured by the hypervolume metric) of each algorithm in each problem as a sample plot.

A. Problems

We report computational results on a number of problems from the literature. For our results on a more comprehensive set of test problems, see [22].

1) *ZDT4:* This $n = 10$-variable problem has the Pareto-optimal frontier shown in Fig. 5. There are 100 local optimal frontiers. Each of these local optimal frontiers can trap an algorithm and cause it to get stuck. Table II shows that IBEA is unable to find any individuals in the nondominated range while others find good approximation sets. We present the test results in Table III. SPEA2 performs the best, while TDEA and ϵ-MOEA follow it, though their performances are very similar. Fig. 4 displays sample distributions in the final populations of these four algorithms. Note that ϵ-MOEA loses individuals toward the best values of the first objective.

2) *ZDT6:* ZDT6 is an $n = 10$-variable problem with a nonconvex Pareto-optimal frontier (Fig. 6). In ZDT6, the feasible objective space becomes less dense as it approaches the Pareto-optimal frontier. In addition, the density of solutions among the Pareto-optimal frontier is not uniform. These cause difficulty in convergence and maintaining a diverse population.

Table IV shows that all metaheuristics are able to converge to the Pareto-optimal frontier with good population diversity.

TABLE IV
INDICATOR RESULTS FOR ZDT6

Algorithm	Hypervolume $\bar{x}_H$	s_H	Inv. Gen. Dist. $\bar{x}_D$	s_D	Additive Epsilon $\bar{x}_I$	s_I	Time (s)
TDEA	0.3995	0.0002	0.000015	0.000002	0.0078	0.0009	0.78
ϵ-MOEA	0.3943	0.0001	0.000062	0.000000	0.1077	0.0004	0.38
IBEA	0.3290	0.0220	0.000225	0.000074	0.0830	0.0258	2.29
NSGA2	0.3991	0.0003	0.000015	0.000002	0.0109	0.0016	4.97
SPEA2	0.3982	0.0006	0.000014	0.000001	0.0075	0.0005	18.94
True Pareto	0.4064	–	0	–	0	–	–

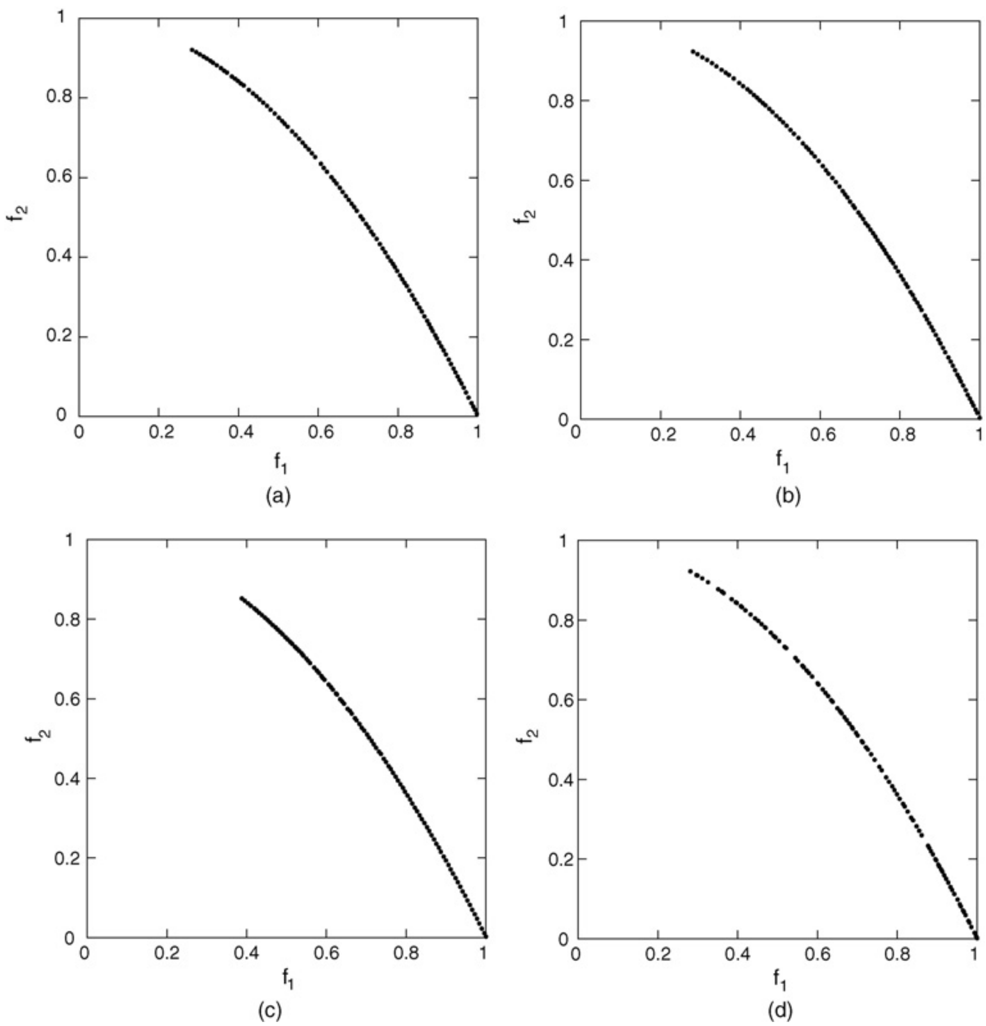


Fig. 7. ZDT6 plots. (a) TDEA. (b) SPEA2. (c) ϵ-MOEA. (d) NSGA2.

TABLE V
TEST RESULTS FOR ZDT6

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ versus $H_1 : \mu_{TDEA} \neq \mu_{Contender}$								
	Hypervolume			Inv. Gen. Dist.			Additive Epsilon		
Contender	Δ_H	p-Value	Winner	Δ_D	p-Value	Winner	Δ_I	p-Value	Winner
ϵ-MOEA	0.0051	0	TDEA	−0.000046	0	TDEA	−0.1000	0	TDEA
IBEA	0.0705	0	TDEA	−0.000210	0	TDEA	−0.0753	0	TDEA
NSGA2	0.0004	0	TDEA	0.000001	0.065	None	−0.0031	0	TDEA
SPEA2	0.0012	0	TDEA	0.000002	0	SPEA2	0.0003	0.034	None

TABLE VI
INDICATOR RESULTS FOR DTLZ1

	Hypervolume		Inv. Gen. Dist.		Additive Epsilon		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	$\bar{x}_I$	s_I	Time (s)
TDEA	0.8023	0.0007	0.000091	0.000001	0.0221	0.0009	4.46
ϵ-MOEA	0.7569	0.0040	0.000101	0.000006	0.0232	0.0005	2.37
IBEA	0.4467	0.0633	0.000802	0.000331	0.1857	0.0552	22.09
NSGA2	0.7843	0.0030	0.000130	0.000006	0.0413	0.0044	47.49
SPEA2	0.8024	0.0005	0.000095	0.000001	0.0241	0.0010	353.18
True Pareto	0.8305	–	0	–	0	–	–

In statistical tests (Table V), we see that TDEA outperforms all algorithms in the hypervolume metric, whereas SPEA2 has the best inverted generational distance score. There are no statistically significant differences between them in additive ϵ-indicator metric. We again observe in Fig. 7 that ϵ-MOEA misses individuals toward the smaller values of objective 1. SPEA2 and TDEA have well-distributed individuals along the frontier.

3) *DTLZ1:* DTLZ1 is a three-objective problem that has a linear Pareto optimal frontier (see Fig. 9). Performance metric scores and test results are given in Tables VI and VII, respectively. Test results show that TDEA is better than other metaheuristics in both metrics, except that it is at par with SPEA2 in the hypervolume metric.

Fig. 8 shows the plots of the populations from sample runs found by TDEA, SPEA2, and ϵ-MOEA. Although ϵ-MOEA produces finely spaced individuals around the center of the objective space, it loses individuals toward the extremes, which causes it to have low scores. On the other hand, TDEA and SPEA2 maintain diversity throughout the entire frontier.

4) *DTLZ2:* DTLZ2 is a three-objective problem that has a spherical Pareto-optimal frontier (Fig. 10). We present indicator results in Table VIII. In this problem, TDEA performs better than all other metaheuristics in all but additive ϵ-indicator metric (see Table IX). Interestingly, IBEA's performance is noticeably better than that in DTLZ1.

Fig. 11 shows results of sample runs for different algorithms. It can be seen that TDEA and SPEA2 maintain the diversity well, whereas others have gaps at different parts of the frontier. The individuals of ϵ-MOEA are not as finely-spaced as in DTLZ1. We see that there are gaps toward the ends and between edges and the center.

5) *DTLZ3:* DTLZ3 is a modified version of DTLZ2 (see Fig. 13 for Pareto-optimal frontier). The change does not affect the Pareto-optimality conditions but adds multimodality. That is, it has many local frontiers that are not Pareto-optimal. It tests the ability of an MOEA to converge the true Pareto-optimal frontier. As can be seen from Table X, SPEA2 has the best values. TDEA outperforms the remaining algorithms, except ϵ-MOEA in additive ϵ-indicator metric (Table XI). IBEA performs poorly and remains far from the Pareto-optimal front. We present sample plots for TDEA, SPEA2, ϵ-MOEA, and NSGA2 in Fig. 12. We observe similar patterns to those in DTLZ2.

6) *DTLZ7:* DTLZ7's Pareto-optimal frontier consists of 2^{m-1} disconnected parts. We present its frontier for the 3-objective case in Fig. 14. The density of solutions is uniform throughout the frontier. It tests an algorithm's ability to maintain populations in all disconnected regions. As can be seen from Table XII, all algorithms converge successfully to the Pareto-optimal frontier. Table XIII shows the statistical comparisons of TDEA with other algorithms. The results indicate that all algorithms perform equally well in this problem.

We present the sample plots of TDEA, SPEA2, ϵ-MOEA, IBEA, and NSGA2 in Fig. 15. ϵ-MOEA displays the best diversity with TDEA and SPEA2 following it. NSGA2 produces an irregular pattern, while IBEA has difficulty in finding individuals toward the center.

7) *DTLZ1-5D:* As in the 3-objective DTLZ1, DTLZ1-5D has a linear Pareto-optimal frontier in five objectives. It is a multimodal problem with many local Pareto-optimal frontiers. These local frontiers attract MOEAs and cause difficulties for them to converge the true frontier. Table XIV shows that TDEA and ϵ-MOEA are the most successful algorithms

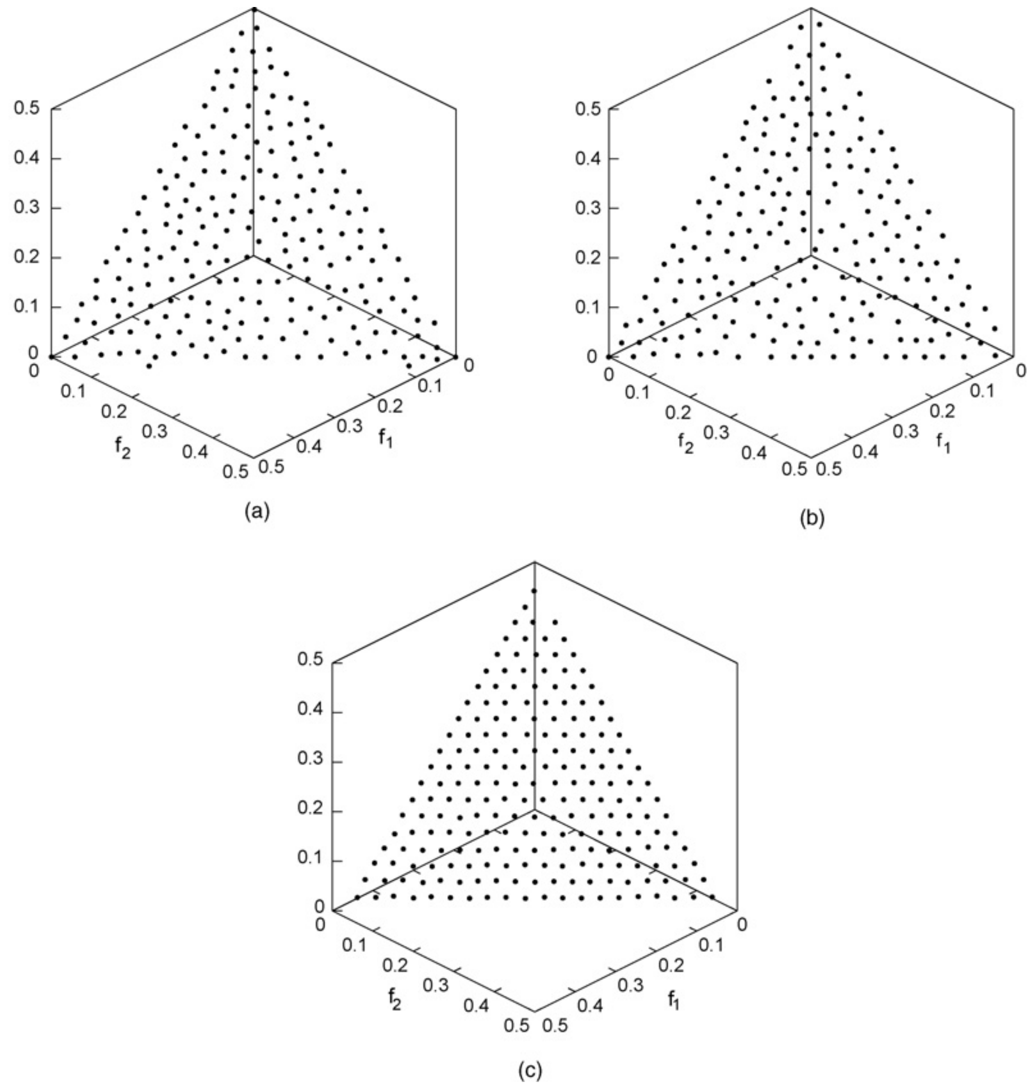


Fig. 8. DTLZ1 plots. (a) TDEA. (b) SPEA2. (c) ϵ-MOEA.

TABLE VII
TEST RESULTS FOR DTLZ1

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ versus $H_1 : \mu_{TDEA} \neq \mu_{Contender}$								
	Hypervolume			Inv. Gen. Dist.			Additive Epsilon		
Contender	Δ_H	p-Value	Winner	Δ_D	p-Value	Winner	Δ_I	p-Value	Winner
ϵ-MOEA	0.0454	0	TDEA	−0.000010	0	TDEA	−0.0011	0	TDEA
IBEA	0.3556	0	TDEA	−0.000711	0	TDEA	−0.1636	0	TDEA
NSGA2	0.0180	0	TDEA	−0.000039	0	TDEA	−0.0192	0	TDEA
SPEA2	−0.0001	0.277	None	−0.000003	0	TDEA	−0.0020	0	TDEA

TABLE VIII
INDICATOR RESULTS FOR DTLZ2

Algorithm	Hypervolume $\bar{x}_H$	s_H	Inv. Gen. Dist. $\bar{x}_D$	s_D	Additive Epsilon $\bar{x}_I$	s_I	Time (s)
TDEA	0.4334	0.0011	0.000122	0.000002	0.0608	0.0044	7.51
ϵ-MOEA	0.4306	0.0005	0.000162	0.000002	0.0435	0.0008	4.37
IBEA	0.4325	0.0005	0.000272	0.000006	0.0641	0.0045	20.39
NSGA2	0.4036	0.0036	0.000173	0.000006	0.0997	0.0140	37.25
SPEA2	0.4259	0.0010	0.000130	0.000002	0.0630	0.0069	683.86
True Pareto	0.4740	–	0	–	0	–	–

TABLE IX
TEST RESULTS FOR DTLZ2

$H_0 : \mu_{TDEA} = \mu_{Contender}$ versus $H_1 : \mu_{TDEA} \neq \mu_{Contender}$

Contender	Hypervolume Δ_H	p-Value	Winner	Inv. Gen. Dist. Δ_D	p-Value	Winner	Additive Epsilon Δ_I	p-Value	Winner
ϵ-MOEA	0.0028	0	TDEA	−0.000040	0	TDEA	0.0173	0	ϵ-MOEA
IBEA	0.0009	0	TDEA	−0.000150	0	TDEA	−0.0033	0	TDEA
NSGA2	0.0298	0	TDEA	−0.000051	0	TDEA	−0.0390	0	TDEA
SPEA2	0.0075	0	TDEA	−0.000008	0	TDEA	−0.0022	0	None

TABLE X
INDICATOR RESULTS FOR DTLZ3

Algorithm	Hypervolume $\bar{x}_H$	s_H	Inv. Gen. Dist. $\bar{x}_D$	s_D	Additive Epsilon $\bar{x}_I$	s_I	Time (s)
TDEA	0.4284	0.0025	0.000124	0.000003	0.0622	0.0036	4.34
ϵ-MOEA	0.4204	0.0047	0.000182	0.000007	0.0494	0.0031	2.48
IBEA	0.0000	0.0000	0.046090	0.027260	15.0300	8.6500	20.81
NSGA2	0.4078	0.0041	0.000169	0.000005	0.1004	0.0139	49.41
SPEA2	0.4318	0.0020	0.000126	0.000002	0.0581	0.0057	251.61
True Pareto	0.4740	–	0	–	0	–	–

TABLE XI
TEST RESULTS FOR DTLZ3

$H_0 : \mu_{TDEA} = \mu_{Contender}$ versus $H_1 : \mu_{TDEA} \neq \mu_{Contender}$

Contender	Hypervolume Δ_H	p-Value	Winner	Inv. Gen. Dist. Δ_D	p-Value	Winner	Additive Epsilon Δ_I	p-Value	Winner
ϵ-MOEA	0.0080	0	TDEA	−0.000058	0	TDEA	0.0128	0	ϵ-MOEA
IBEA	N/A	N/A	TDEA	N/A	N/A	TDEA	−14.9700	0	TDEA
NSGA2	0.0207	0	TDEA	−0.000045	0	TDEA	−0.0382	0	TDEA
SPEA2	−0.0034	0	SPEA2	−0.000002	0.004	None	0.0041	0	SPEA2

TABLE XII
INDICATOR RESULTS FOR DTLZ7

Algorithm	Hypervolume $\bar{x}_H$	s_H	Inv. Gen. Dist. $\bar{x}_D$	s_D	Additive Epsilon $\bar{x}_I$	s_I	Time (s)
TDEA	0.3206	0.0140	0.000162	0.000304	0.0370	0.0404	7.58
ϵ-MOEA	0.3163	0.0014	0.000080	0.000001	0.0289	0.0018	4.84
IBEA	0.3016	0.0374	0.000686	0.000576	0.0842	0.1922	20.76
NSGA2	0.3108	0.0023	0.000127	0.000007	0.0347	0.0103	39.11
SPEA2	0.3192	0.0010	0.000103	0.000003	0.0382	0.0106	551.75
True Pareto	0.3444	–	0	–	0	–	–

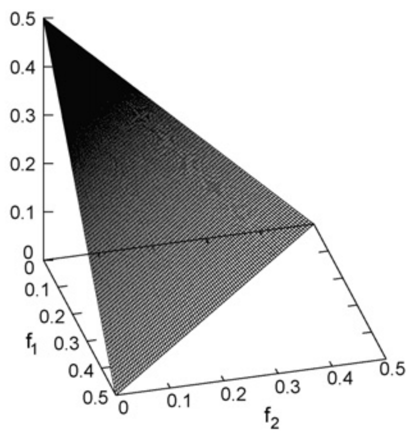


Fig. 9. Pareto-optimal frontier of DTLZ1.

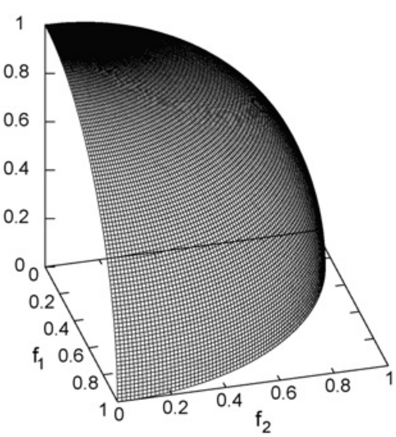


Fig. 10. Pareto-optimal frontier of DTLZ2.

in converging the Pareto-optimal frontier. While IBEA and NSGA2 partially converge, SPEA2 is unable to find any individuals within the nondominated range. In test results (Table XV), TDEA outperforms all algorithms except ϵ-MOEA in additive ϵ-indicator metric. Note that the standard deviation values of TDEA are also the smallest, which means that it has the least difficulty in finding good final populations compared to other algorithms. Furthermore, the average times to complete their runs for ϵ-MOEA and TDEA are much smaller than the other algorithms.

8) *DTLZ2-5D:* DTLZ2-5D is the 5-objective version of DTLZ2, sharing the same properties. Indicator results in Table XVI show that all algorithms converge the Pareto-optimal frontier, although there are significant differences in metric values. This is also confirmed in test results (Table XVII). TDEA and IBEA display better hypervolume performances than other algorithms, indicating better convergence. However, TDEA scores better in the inverted generational distance compared to IBEA. However, in additive ϵ-indicator metric ϵ-MOEA outperforms TDEA and IBEA's performance is similar. The durations of ϵ-MOEA and TDEA are much smaller than those of the other algorithms.

B. *Effects of Changing τ*

In TDEA, τ controls the size of the territory of an individual. Its value changes the hypervolume that an individual occupies in the objective space. Since the total nondominated hypervolume is limited, the size of the archive depends on τ. The population size is expected to be smaller with a larger τ compared to that with a smaller τ. TDEA scales all objectives into the $[0, 1]$ range, hence τ is always between 0 and 1. However, a τ value set for a problem may yield a larger or smaller population for another problem, because the space between individuals changes with the shape of the Pareto-optimal frontier. Also the number of objectives affects the value of τ. For a large number of objectives, τ needs to be large in order to keep the population size reasonable.

A smaller τ yields a better approximation of the Pareto-optimal frontier than a larger τ. However, it requires more computational effort since the size of the archive population increases. To observe the effect of different τ values, we make test runs on problems ZDT4, DTLZ1, and DTLZ2 with various τ values and compare their results. For each problem, we use three different τ values and replicate each run 50 times. For each problem and τ value, we find the average metric values, average duration, and average final archive size.

In Table XVIII, we observe that hypervolume values slightly increase as τ decreases. Average duration of runs also increases when τ is decreased. Fig. 16 illustrates the difference in the details of final populations. Tables XIX and XX give results for the 3-objective problems DTLZ1 and DTLZ2. It can be seen that decreasing τ gives more detail in return for increased computational cost. In all problems, decreasing τ increases the size of final populations. However, using the same τ for different problems does not necessarily lead to the same number of individuals in final populations. Although $\tau = 0.03$

TABLE XIII
TEST RESULTS FOR DTLZ7

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ versus $H_1 : \mu_{TDEA} \neq \mu_{Contender}$								
	Hypervolume			Inv. Gen. Dist.			Additive Epsilon		
Contender	Δ_H	p-Value	Winner	Δ_D	p-Value	Winner	Δ_I	p-Value	Winner
ϵ-MOEA	0.0043	0.037	None	0.000082	0.062	None	0.0080	0.168	None
IBEA	0.0189	0.001	TDEA	−0.000524	0	TDEA	−0.0473	0.094	None
NSGA2	0.0097	0	TDEA	0.000035	0.414	None	0.0022	0.705	None
SPEA2	0.0013	0.501	None	0.000059	0.174	None	−0.0013	0.829	None

TABLE XIV
INDICATOR RESULTS FOR DTLZ1-5D

	Hypervolume		Inv. Gen. Dist.		Additive Epsilon		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	$\bar{x}_I$	s_I	Time (s)
TDEA	0.9520	0.0016	0.000352	0.000020	0.0673	0.0040	25.745
ϵ-MOEA	0.9150	0.0084	0.000392	0.000031	0.0425	0.0009	13.3264
IBEA	0.5667	0.0518	0.000915	0.000106	0.1663	0.0372	114.295
NSGA2	0.1539	0.2943	0.012060	0.013180	1.8000	1.8530	163.369
SPEA2	0.0000	0.0000	0.118330	0.030570	14.8010	3.6520	2140

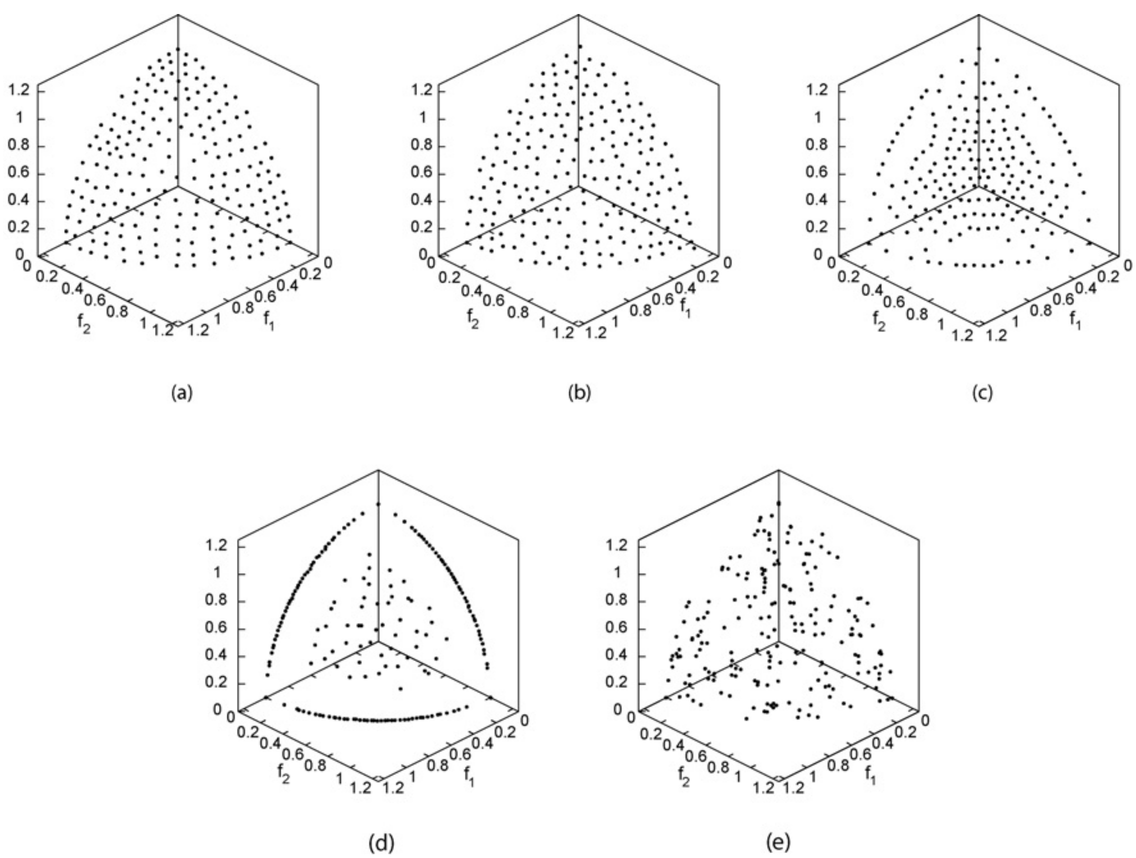


Fig. 11. DTLZ2 plots. (a) TDEA. (b) SPEA2. (c) ϵ-MOEA. (d) IBEA. (e) NSGA2.

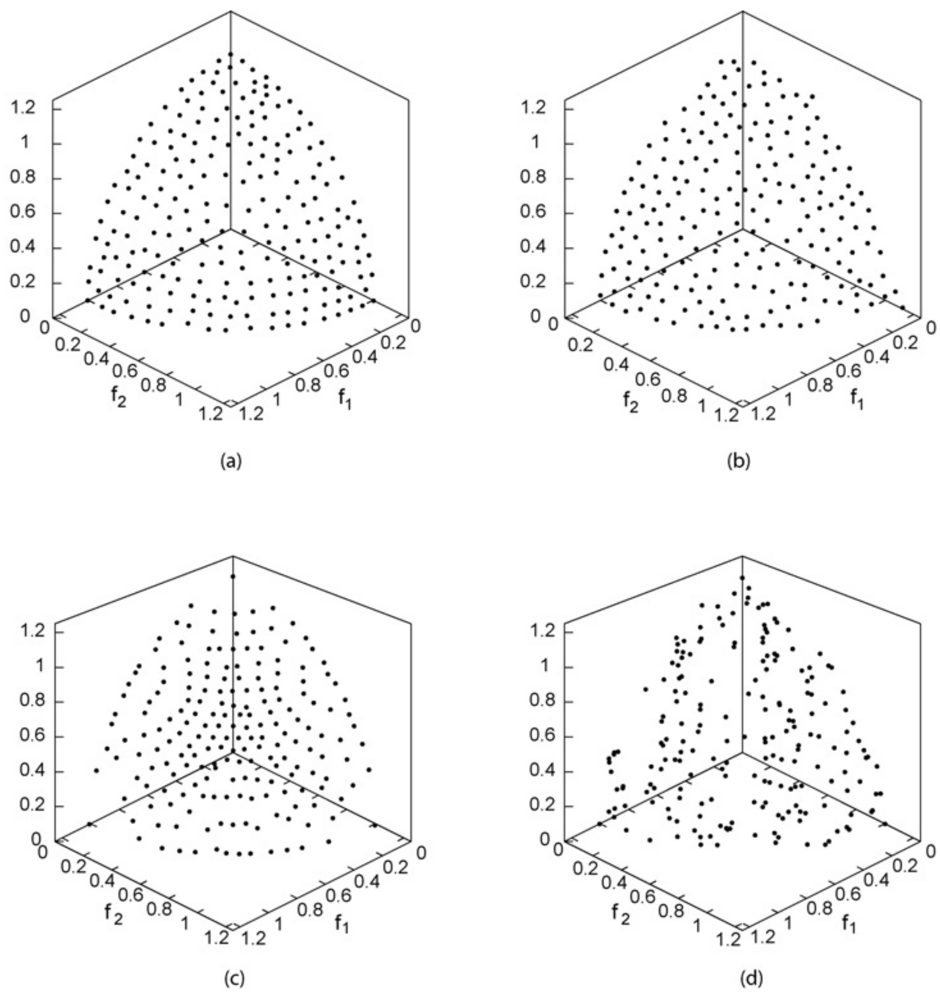


Fig. 12. DTLZ3 plots. (a) TDEA. (b) SPEA2. (c) ϵ-MOEA. (d) NSGA2.

TABLE XV
TEST RESULTS FOR DTLZ1-5D

$H_0 : \mu_{TDEA} = \mu_{Contender}$ versus $H_1 : \mu_{TDEA} \neq \mu_{Contender}$									
	Hypervolume			Inv. Gen. Dist.			Additive Epsilon		
Contender	Δ_H	p-Value	Winner	Δ_D	p-Value	Winner	Δ_I	p-Value	Winner
ϵ-MOEA	0.0370	0	TDEA	−0.000040	0	TDEA	0.0248	0	ϵ-MOEA
IBEA	0.3853	0	TDEA	−0.000562	0	TDEA	−0.0990	0	TDEA
NSGA2	0.7981	0	TDEA	−0.011709	0	TDEA	−1.7330	0	TDEA
SPEA2	N/A	N/A	TDEA	N/A	N/A	TDEA	−14.7340	0	TDEA

TABLE XVI
INDICATOR RESULTS FOR DTLZ2-5D

	Hypervolume		Inv. Gen. Dist.		Additive Epsilon		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	$\bar{x}_I$	s_I	Time (s)
TDEA	0.7121	0.0019	0.000434	0.000009	0.1526	0.0110	41.1262
ϵ-MOEA	0.6935	0.0019	0.000486	0.000006	0.1348	0.0028	22.4666
IBEA	0.7234	0.0005	0.000633	0.000016	0.1544	0.0047	106.964
NSGA2	0.4116	0.0304	0.000794	0.000069	0.3363	0.0410	165.432
SPEA2	0.5926	0.0101	0.000595	0.000036	0.2270	0.0203	4692.72

TABLE XVII
TEST RESULTS FOR DTLZ2-5D

$H_0 : \mu_{TDEA} = \mu_{Contender}$ versus $H_1 : \mu_{TDEA} \neq \mu_{Contender}$

	Hypervolume			Inv. Gen. Dist.			Additive Epsilon		
Contender	Δ_H	p-Value	Winner	Δ_D	p-Value	Winner	Δ_I	p-Value	Winner
ϵ-MOEA	0.0186	0	TDEA	−0.000052	0	TDEA	0.0177	0	ϵ-MOEA
IBEA	−0.0113	0	IBEA	−0.000199	0	TDEA	−0.0019	0.268	None
NSGA2	0.3005	0	TDEA	−0.000360	0	TDEA	−0.1837	0	TDEA
SPEA2	0.1195	0	TDEA	−0.000161	0	TDEA	−0.0744	0	TDEA

TABLE XVIII
INDICATOR RESULTS FOR ZDT4 WITH DIFFERENT τ VALUES

	Hypervolume		Inv. Gen. Dist.		Additive Epsilon			Average
τ	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	$\bar{x}_I$	s_I	Time (s)	Archive Size
0.0005	0.6646	0.0005	0.000003	0.000003	0.0022	0.0004	3.7	~1400
0.001	0.6644	0.0005	0.000003	0.000001	0.0024	0.0035	2.7	~750
0.005	0.6626	0.0004	0.000009	0.000000	0.0055	0.0004	1.7	~170
True Pareto	0.6660	–	0	–	0	–	–	

TABLE XIX
INDICATOR RESULTS FOR DTLZ1 WITH DIFFERENT τ VALUES

	Hypervolume		Inv. Gen. Dist.		Additive Epsilon			Average
τ	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	$\bar{x}_I$	s_I	Time (s)	Archive Size
0.01	0.8231	0.0008	0.000025	0.000001	0.0072	0.0005	86.1	~3000
0.02	0.8176	0.0004	0.000046	0.000000	0.0116	0.0008	22.4	~800
0.03	0.8110	0.0003	0.000066	0.000001	0.0162	0.0009	15.4	~400
True Pareto	0.8305	–	0	–	0	–	–	

results in a population size of approximately 400 in DTLZ1, the population size in DTLZ2 is around 1000 with the same τ value. To obtain the same population size in DTLZ2, it is necessary to set $\tau = 0.05$. This occurs because the Pareto-optimal frontiers of these problems have different shapes. Since DTLZ2 has a larger nondominated hypervolume than DTLZ1, the same τ yields larger populations in DTLZ2. In addition, when the number of objectives increases, the same τ results in larger populations. The plots in Figs. 17 and 18 clearly display the difference in the densities of the final populations when τ is changed. Note that in all of the problems and for all τ values, TDEA successfully maintains the diversity among the entire population.

C. Discussions

The tests show that TDEA is able to converge the true Pareto-optimal frontier in all of the problems. It maintains a uniform diversity of individuals over the entire Pareto-optimal frontier and obtains a good representation of the true frontier. We observe that it performs better than other metaheuristics in most of the problems with respect to the hypervolume, inverted generational distance, and additive ϵ-indicator metrics. Thanks to its territory defining property, it maintains diversity while obtaining the fastest execution time after ϵ-MOEA. As shown in the plots, this mechanism also helps the algorithm to obtain the best spread of individuals together with SPEA2.

The number of function evaluations we use in our experiments is larger than those used in the original experiments reported for each of the algorithms we tested. Since IBEA performed poorly in many of the problems, we repeated some of its runs with much longer function evaluations to see if there is a potential for improvement. We did not observe improvements. Other algorithms performed fairly well in two and three-objective problems. The performances of NSGA2 and SPEA2 deteriorated substantially for the five-objective problems. It is unlikely to obtain sizeable improvements with reasonable increases in the number of function evaluations. Furthermore, the execution times of SPEA2 are already very large for these problems. The performance of ϵ-MOEA is fairly good in all problems. As we showed on a hypothetical case, as well as in our experiments, it seems to miss the solutions in the extreme regions for each objective. Increasing the number of function evaluations is unlikely to improve the performance in those regions since those solutions are ϵ-dominated.

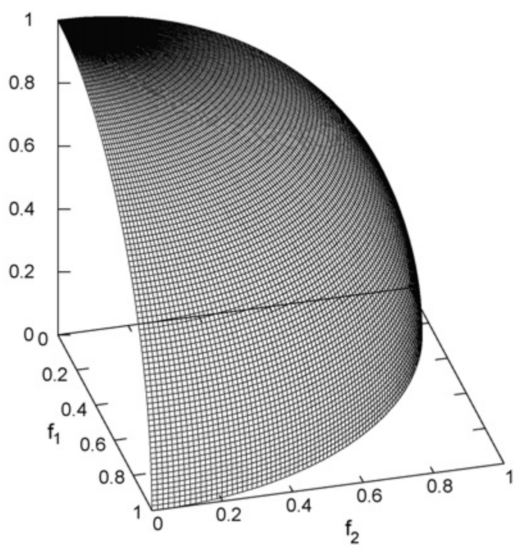


Fig. 13. Pareto-optimal frontier of DTLZ3.

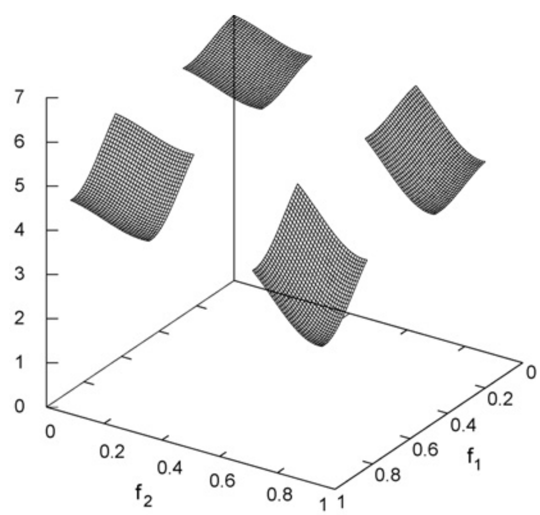


Fig. 14. Pareto-optimal frontier of DTLZ7.

IV. Preference Incorporation

In this section, we present a method to incorporate preference information to guide the search to those regions that are more desirable for the decision maker.

A. Variable Territory Sizes

The results in Section III-B show that a smaller τ brings better details, though with extra computational cost. However, as mentioned before, approximating the entire Pareto-optimal frontier is not very useful. Instead, focusing on a particular region helps to avoid much of the additional effort. This is only possible if preferences of the DM can be integrated into the solution procedure. TDEA's territory defining property can be effectively utilized for this purpose. If the DM's preferred region is defined, then the algorithm can concentrate by shrinking the territories of the individuals falling in this region. This can be achieved by simply using a smaller τ for such offspring in the archive evaluation stage. This maintains more individuals in this region in the archive population, leading to a higher resolution and better approximation. Meanwhile, individuals located elsewhere can be evaluated using a larger τ. This leads to less dense populations in the regions that are less desirable to the DM. An illustration of this idea is shown in Fig. 19. Consequently, the search in the less desirable regions is discouraged as the individuals in those regions will have fewer opportunities to participate in the genetic operations. In this way, the number of individuals in the focused region is kept large without enlarging the size of the entire population. Note that the individuals outside the preferred region are not lost, but only a few of them allowed to survive in the archive. Hence, the final population of the algorithm still gives an approximation of the entire Pareto-optimal frontier, providing a sense of locations of possible solutions.

B. Favorable Weights

To make variable territory sizes operational, we have to define the region to be focused on and determine whether individuals fall within this region. In TDEA, we use the idea of favorable weights (see [29]) for this purpose. The favorable weights $\mathbf{w^s} = \{w_1^s, w_2^s, \ldots, w_m^s\}$ of individual s are a set of weights that minimizes its weighted Tchebycheff distance from the ideal point. They are computed as shown in (4). Note that we dropped s from the superscript to simplify the notation.

$$w_i = \begin{cases} \dfrac{1}{f_i^* - f_i}\left[\displaystyle\sum_{j=1}^{m} \dfrac{1}{f_j^* - f_j}\right]^{-1}, & \text{if } f_j \neq f_j^* \text{for all} \\ & j = 1, 2, \ldots, m \\ 1, & \text{if } f_i = f_i^* \\ 0, & \text{if } f_i \neq f_i^* \text{ but } \exists j \\ & \text{such that } f_j = f_j^* \end{cases} \tag{4}$$

where f_i is the ith objective value and f_i^* is the ith element of the ideal objective vector (see [30, p. 425]).

These weights determine the direction in which an individual contributes the highest to the convergence. More specifically, the corresponding individual is in the direction $\frac{1}{w_i}, i = 1, 2, \ldots, m$, from the ideal point. We define a *preferred region* R_P by a set of Tchebycheff weights $\mathbf{W}^P$. Then, we say that an individual s is in R_P if its weights, $\mathbf{w}^s$, satisfy $\mathbf{w}^s \in \mathbf{W}^P$. Otherwise, s is not in R_P. In other words, an individual is favorable if its favorable weights are covered by the weight ranges of a preferred region.

Restricting the weight space is one of the common approaches in the multiple criteria decision making literature to reach the solutions preferred by the DM (see [30, pp. 361–425, Ch. 13]). There may be different ways of determining the preferred weight sets in our case. For example, a preliminary run can be made and several well-dispersed solutions can be presented to the DM. A rough approximation is sufficient at this stage. Such solutions may be obtained by a short run of TDEA or by problem-specific approximation techniques. The obtained several well-dispersed solutions can be presented to the DM and those preferred by the DM can be used to define the desirable weight regions. The Tchebycheff weight vectors

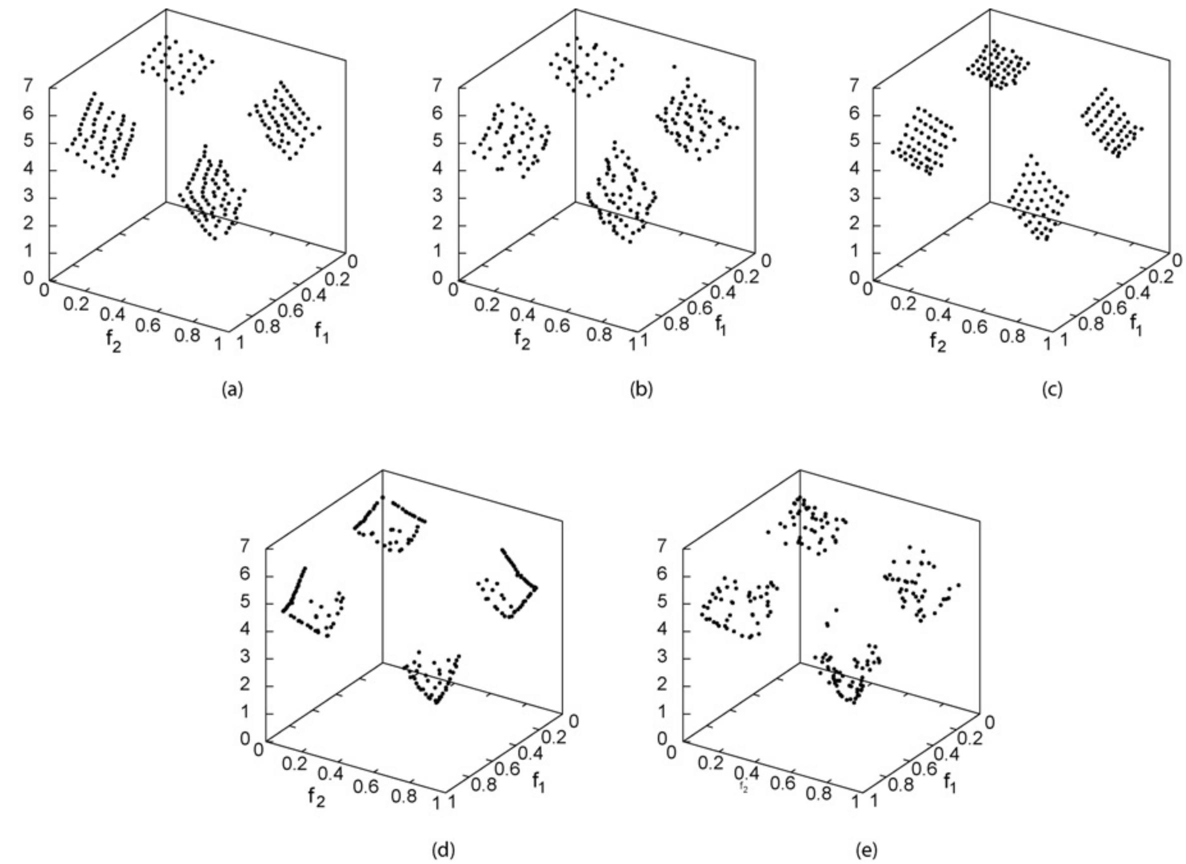


Fig. 15. DTLZ7 plots. (a) TDEA. (b) SPEA2. (c) ϵ-MOEA. (d) IBEA. (e) NSGA2.

in the direction of each preferred solution can be found using (4). Then, a weight region can be constructed around each such weight vector (see [30, pp. 446–450, Ch. 14] for a variation of this procedure).

C. Details of the Algorithm

Preferred-region territory defining evolutionary algorithm (prTDEA) is an application of the preference incorporation mechanism in TDEA. In prTDEA, the DM states his/her preference information before optimization. This is done by explicitly specifying a preference region R_P by its weight set. Then, this region and the remaining space R_U are assigned their own τ values. An offspring is evaluated using the τ of the region to which it corresponds. An outline of the algorithm is given below.

1) Set $\bar{N}$ and T. Ask the user to specify his/her preferred region R_P and set the remaining space as R_U. Assign τ_P to R_P and τ_U to R_U. Set iteration count $t = 0$.
2) Create $\bar{N}$ random individuals to form the initial regular population $P(0)$. Copy the nondominated individuals in $P(0)$ into $A(0)$ to form the initial archive population.
3) Set $t \leftarrow t + 1$. Choose a parent from each of the populations $P(t)$ and $A(t)$. Recombine parents to create new offspring and apply mutation.
4) Check whether the offspring satisfies the acceptance condition into $P(t)$. If it does, insert it into $P(t)$ and go to the next step. Otherwise, go to Step 6.
5) Check whether the offspring satisfies the acceptance condition into $A(t)$. If it does, insert into $A(t)$.
6) If $t = T$, stop and report the archive population. Otherwise go to Step 3.

The presence of a preference region in prTDEA requires a change in the archive acceptance procedure. The process starts with a dominance check. If any individual in the archive dominates offspring c, then c is rejected and the process is terminated. Otherwise, all individuals dominated by c are removed from the archive. If the archive is empty, then c is accepted and the process is terminated. Otherwise, we determine whether the preference region R_P contains c. For this purpose, we compute the favorable weights, $\mathbf{w}^c$, of c and check if the weight set $\mathbf{W}^P$ of R_P contains $\mathbf{w}^c$. If it does, we set $\tau = \tau_P$. Otherwise, we set $\tau = \tau_U$. The rest is the same as TDEA. First, the closest individual s_{i^*} to c in terms of scaled rectilinear distance is determined. Then, we check whether c is in the territory of s_{i^*}. c is rejected if the maximum scaled objective distance between s_{i^*} and c is less than the τ value. Otherwise, it is accepted. We give the details of the procedure below.

1) Test c against each individual $s_i \in A(t)$ for dominance. Mark the individuals dominated by c. If c is dominated by at least one s_i, reject c. Otherwise, go to the next step.
2) Remove all marked individuals from $A(t)$.
3) If $A(t)$ is empty, accept c, insert it into $A(t)$, and stop. Otherwise, go to the next step.

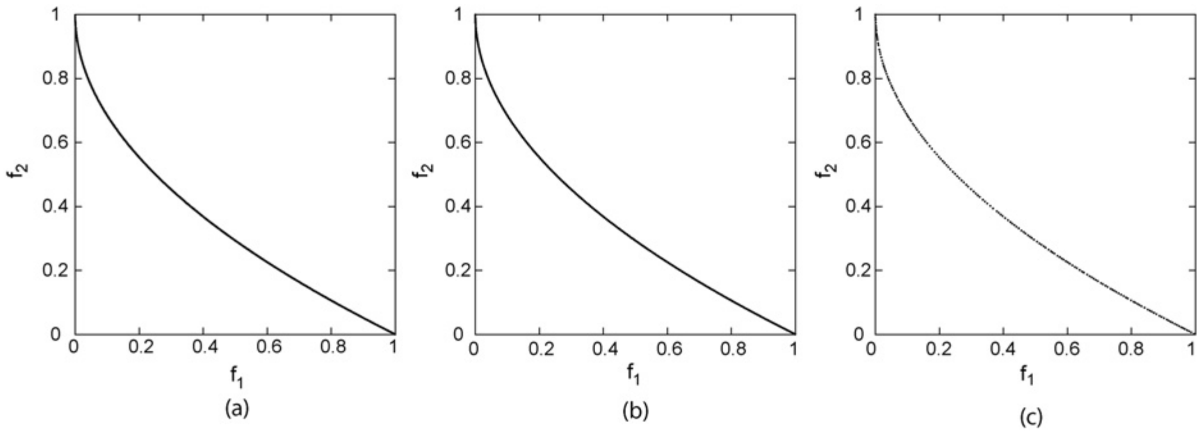


Fig. 16. ZDT4 plots with different τ values. (a) τ = 0.0005. (b) τ = 0.001. (c) τ = 0.005.

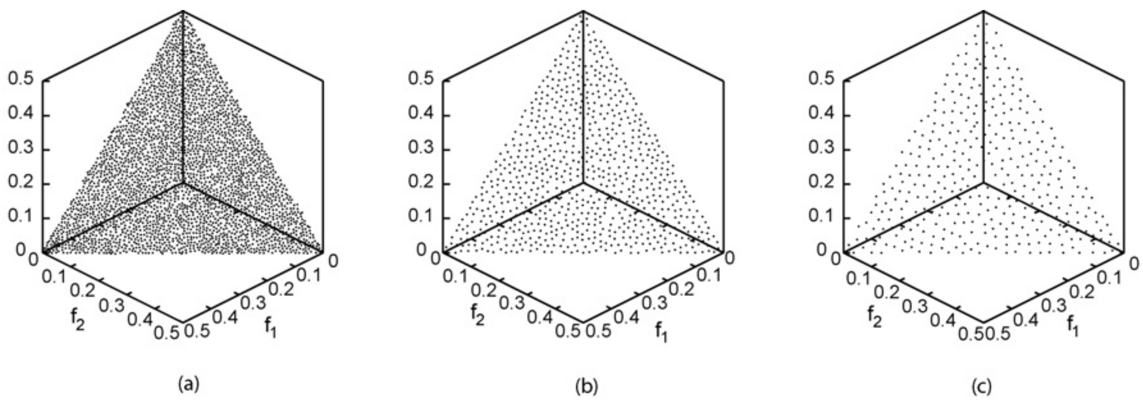


Fig. 17. DTLZ1 plots with different τ values. (a) τ = 0.01. (b) τ = 0.03. (c) τ = 0.05.

4) Calculate the favorable weights $\mathbf{w}^c$ of c with (4) using scaled objective values $\hat{f}_i,\ i = 1, 2, \ldots, m$.
5) Set $\tau = \tau_P$ if the weight set $\mathbf{W}^P$ of R_P contains the favorable weights $\mathbf{w}^c$ of c. Otherwise, set $\tau = \tau_U$. That is, set τ as follows:

$$\tau = \begin{cases} \tau_P, & \text{if } \mathbf{w^c} \in \mathbf{W}^P \\ \tau_U, & \text{otherwise.} \end{cases} \tag{5}$$

6) Calculate the rectilinear distance $d_{ci} = \sum_{j=1}^{m} \left| \hat{f}_{cj} - \hat{f}_{ij} \right|$ of c to each individual $s_i \in A(t)$.
7) Find $i^* = \text{argmin}i(d_{ci})$, that is, the individual s_{i^*} closest to c.
8) Find the maximum scaled absolute objective difference between c and s_{i^*}. That is, find

$$\delta = \max_{j=1,2,\ldots,m} \left| \hat{f}_{cj} - \hat{f_{i^*j}} \right|. \tag{6}$$

9) If $\delta \geq \tau$, accept c and insert it into $A(t)$. Otherwise, reject c.

V. Computational Experiments with prTDEA

Since prTDEA focuses on the region specified by the DM, we expect it to find better approximations of this preferred region than does TDEA under the same test conditions. To test this claim, we make simulation runs on 2-objective ZDT4 [10], and 3 and 5-objective DTLZ1 and DTLZ2 [17] problems. We create four tests for 2 and 3-objective problems, and two tests for 5-objective problems. Each test has a different preference region, R_P, defined by a set of Tchebycheff weight ranges $\mathbf{W}^P = \left[\mathbf{l^P}, \mathbf{u^P}\right] = \left\{\left[l_1^P, u_1^P\right], \left[l_2^P, u_2^P\right], \ldots, \left[l_m^P, u_m^P\right]\right\}$. In prTDEA, a small τ_P value is used for these preference regions, whereas a larger τ_U value is used for the remaining regions. TDEA uses the same small τ_P for the entire Pareto-optimal frontier. Each algorithm is allowed to run for the same number of function evaluations. For statistical analysis, we replicate each test 50 times. After each run, we filter out the individuals of the two algorithms corresponding to the preference region and compute their hypervolume (H) [36], inverted generational distance (D) [2] and additive ϵ-indicator (I) [34] values. D and I metrics are computed against the true Pareto-optimal frontier filtered with respect to the preference region of the test. For visualization, we provide the best run results (as measured by the hypervolume metric) of filtered-TDEA and prTDEA together with the corresponding Pareto-frontier in each problem as sample plots. The details of each test are given in Table XXI. Both algorithms are implemented on C++ and tests are run on a computer with Intel Core 2 Duo T8300 CPU, 4 GB RAM running Ubuntu Linux 9.04.

The $\mathbf{W}^P$ regions in Table XXI are chosen just to demonstrate how prTDEA focuses on different regions. As discussed before, in an application we would need a DM indicate his/her preferred region. We may present several representative

TABLE XX
INDICATOR RESULTS FOR DTLZ2 WITH DIFFERENT τ VALUES

	Hypervolume		Inv. Gen. Dist.		Additive Epsilon			Average
τ	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	$\bar{x}_I$	s_I	Time (s)	Archive Size
0.02	0.4614	0.0002	0.000039	0.000000	0.0221	0.0012	87.1	~2000
0.03	0.4567	0.0002	0.000055	0.000000	0.0303	0.0018	45.1	~1000
0.05	0.4454	0.0006	0.000088	0.000001	0.0462	0.0031	25.7	~400
True Pareto	0.4740	–	0	–	0	–	–	

TABLE XXI
PREFERENCE TEST PARAMETERS

	ZDT4	DTLZ1	DTLZ2
Test 1 $\mathbf{W}^P$	[0.4, 0.6], [0.4, 0.6]	[0.25, 0.5], [0.25, 0.5], [0.25, 0.5]	[0.25, 0.5], [0.25, 0.5], [0.25, 0.5]
Test 2 $\mathbf{W}^P$	[0.3, 0.7], [0.3, 0.7]	[0.2, 0.6], [0.2, 0.6], [0.2, 0.6]	[0.2, 0.6], [0.2, 0.6], [0.2, 0.6]
Test 3 $\mathbf{W}^P$	[0.1, 0.3], [0.7, 0.9]	[0.1, 0.4], [0.3, 0.6], [0.1, 0.6]	[0.1, 0.4], [0.3, 0.6], [0.1, 0.6]
Test 4 $\mathbf{W}^P$	[0.75, 1], [0, 0.25]	[0.4, 0.7], [0.1, 0.3], [0.2, 0.5]	[0.4, 0.7], [0.1, 0.3], [0.2, 0.5]
Function evaluations	80 000	160 000	160 000
Regular population size	200	400	400
τ_P	0.00001	0.005	0.01
τ_U	0.01	0.05	0.1
Replications	50	50	50

	DTLZ1-5D	DTLZ2-5D
Test 1$\mathbf{W}^P$	[0.1, 0.4], [0.1, 0.4], [0.1, 0.4], [0.1, 0.4], [0.1, 0.4]	[0.1, 0.4], [0.1, 0.4], [0.1, 0.4], [0.1, 0.4], [0.1, 0.4]
Test 2$\mathbf{W}^P$	[0.2, 0.6], [0.1, 0.45], [0.1, 0.35],[0.1, 0.35], [0.1, 0.35]	[0.2, 0.6], [0.1, 0.45], [0.1, 0.35], [0.1, 0.35], [0.1, 0.35]
Function evaluations	320 000	320 000
Regular population size	400	400
τ_P	0.03	0.07
τ_U	0.2	0.3
Replications	50	50

TABLE XXII
A SET OF REPRESENTATIVE SOLUTIONS

Obj.	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5
1	0.103	0.186	0.327	0.137	0.195
2	0.155	0.031	0.164	0.412	0.195
3	0.207	0.186	0.041	0.206	0.292
4	0.310	0.124	0.327	0.041	0.146
5	0.124	0.372	0.041	0.103	0.073

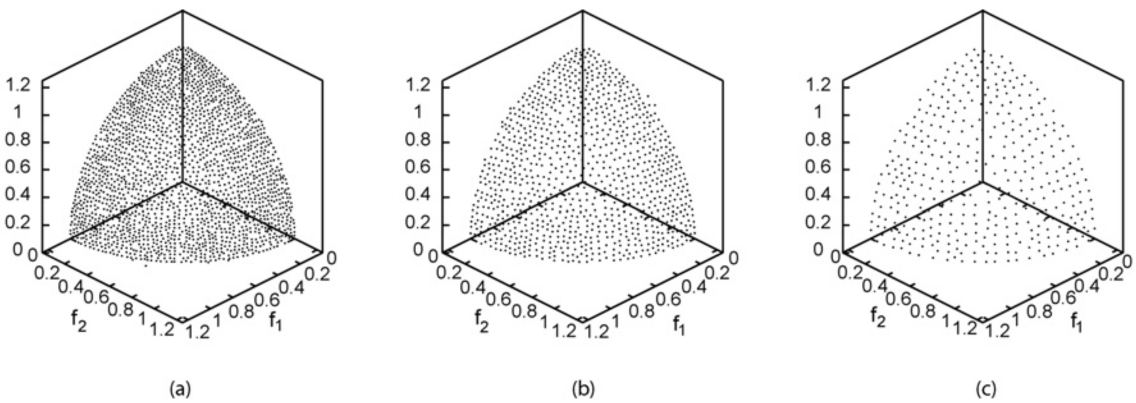


Fig. 18. DTLZ2 plots with different τ values. (a) τ = 0.02. (b) τ = 0.03. (c) τ = 0.05.

solutions to the DM from a population obtained in a preliminary run. We may then find the favorable weights of the solution preferred by the DM, using (4). We may then create an interval around the favorable weights to restrict the weight space. For example, assume we select the five representative solutions given in Table XXII from our preliminary run.

Since the criterion values of these solutions are scaled, we would transform them back to the original scale relevant to the DM. Assume that the DM selects the first solution corresponding to original criterion values $(0.057, 0.086, 0.115, 0.172, 0.069)$. Then using (4) we obtain the favorable weight set $(0.30, 0.20, 0.15, 0.10, 0.25)$. Then we may, for example, create the weight ranges $[0.2, 0.4]$, $[0.1, 0.3]$, $[0.05, 0.25]$, $[0.00, 0.20]$ and $[0.15, 0.35]$ for w_1, w_2, w_3, w_4, and w_5, respectively.

A. 2-Objective Tests

We use problem ZDT4 to conduct 2-objective preference tests.

1) *Test 1:* The first test covers a region in the middle of the Pareto-optimal frontier [Fig. 20(a)]. As it can be observed from Table XXIII, prTDEA outperforms TDEA in all metrics. Table XXVII shows that the differences are statistically significant. Furthermore, the average run times for prTDEA are shorter. Sample plots of the results of the two algorithms are given in Fig. 20.

2) *Test 2:* The region in this test is an enlarged version of the region in Test 1 as seen in Fig. 21(a). Although prTDEA still gives better results than TDEA (Tables XXIV and XXVII), the gap between the two algorithms decreases. This is because the focused region is larger. We present sample plots of the results of the two algorithms in Fig. 21.

3) *Test 3:* The preference region of this test is toward the better values of the second objective as shown in Fig. 22(a). In Table XXV, we observe that prTDEA finds better results in all metrics. However, the differences between the inverted generational distance and additive ϵ-indicator metric values are not statistically significant (Table XXVII), although the estimated difference is the largest among all tests. This is due to the high standard deviation in TDEA's results. Fig. 21 shows the sample plots of the results of the two algorithms.

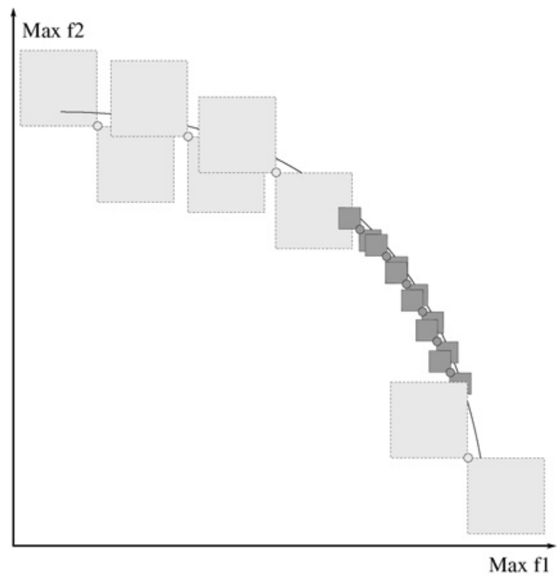


Fig. 19. Different territory sizes.

4) *Test 4:* The preference region in Test 4 covers the best values of the first objective [Fig. 23(a)]. Similar to the previous results, Table XXVI indicates that prTDEA performs better than TDEA in all metrics, as well as in the average time required to complete the runs. The differences are statistically significant as seen in Table XXVII. Sample plots of the final populations are presented in Fig. 23.

B. 3-Objective Tests

We conduct 3-objective tests on DTLZ1 and DTLZ2 problems using the same desirable regions for both problems. The reason behind using two problems is to test whether the shape of the Pareto-optimal frontier affects the performance.

1) *Test 1:* The first preference region [Fig. 24(a) and (d)] is set as the middle of the Pareto-optimal frontier. We see from Table XXVIII that prTDEA outperforms TDEA as in ZDT4 tests and the difference between computation times is substantially lower for prTDEA. The differences between sample plots (Fig. 24) are also clearly observable.

TABLE XXIII
INDICATOR RESULTS FOR ZDT4 PREFERENCE TEST 1

Algorithm	Hypervolume $\bar{x}_H$	s_H	Inv. Gen. Dist. $\bar{x}_D$	s_D	Additive Epsilon $\bar{x}_I$	s_I	Time (s)
prTDEA	0.5139	0.0024	0.000023	0.000012	0.0006	0.0002	2.05
TDEA	0.5072	0.0046	0.000053	0.000023	0.0015	0.0005	2.97
True Pareto	0.5185	–	0	–	0	–	–

TABLE XXIV
INDICATOR RESULTS FOR ZDT4 PREFERENCE TEST 2

Algorithm	Hypervolume $\bar{x}_H$	s_H	Inv. Gen. Dist. $\bar{x}_D$	s_D	Additive Epsilon $\bar{x}_I$	s_I	Time (s)
prTDEA	0.5348	0.0015	0.000011	0.000005	0.0009	0.0003	2.36
TDEA	0.5325	0.0022	0.000018	0.000008	0.0016	0.0005	2.97
True Pareto	0.5380	–	0	–	0	–	–

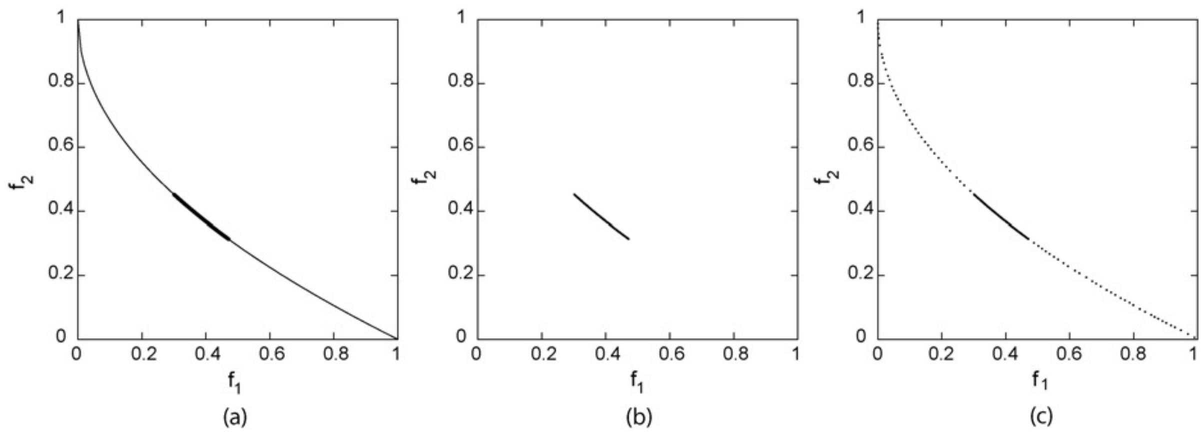


Fig. 20. ZDT4 test 1 plots. (a) Preference region 1. (b) Filtered TDEA. (c) prTDEA.

TABLE XXV
INDICATOR RESULTS FOR ZDT4 PREFERENCE TEST 3

Algorithm	Hypervolume $\bar{x}_H$	s_H	Inv. Gen. Dist. $\bar{x}_D$	s_D	Additive Epsilon $\bar{x}_I$	s_I	Time (s)
prTDEA	0.5117	0.0019	0.000014	0.000008	0.0008	0.0004	2.23
TDEA	0.5031	0.0173	0.000083	0.000250	0.0042	0.0112	2.97
True Pareto	0.5153	–	0	–	0	–	–

TABLE XXVI
INDICATOR RESULTS FOR ZDT4 PREFERENCE TEST 4

Algorithm	Hypervolume $\bar{x}_H$	s_H	Inv. Gen. Dist. $\bar{x}_D$	s_D	Additive Epsilon $\bar{x}_I$	s_I	Time (s)
prTDEA	0.6623	0.0014	0.000014	0.000007	0.0010	0.0004	2.37
TDEA	0.6607	0.0018	0.000021	0.000009	0.0015	0.0006	2.97
True Pareto	0.6651	–	0	–	0	–	–

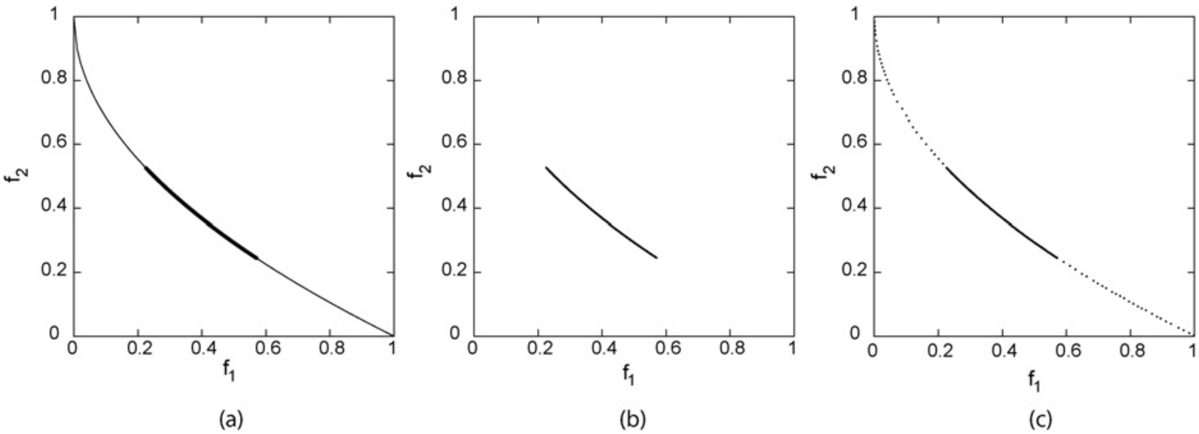


Fig. 21. ZDT4 test 2 plots. (a) Preference region 2. (b) Filtered TDEA. (c) prTDEA.

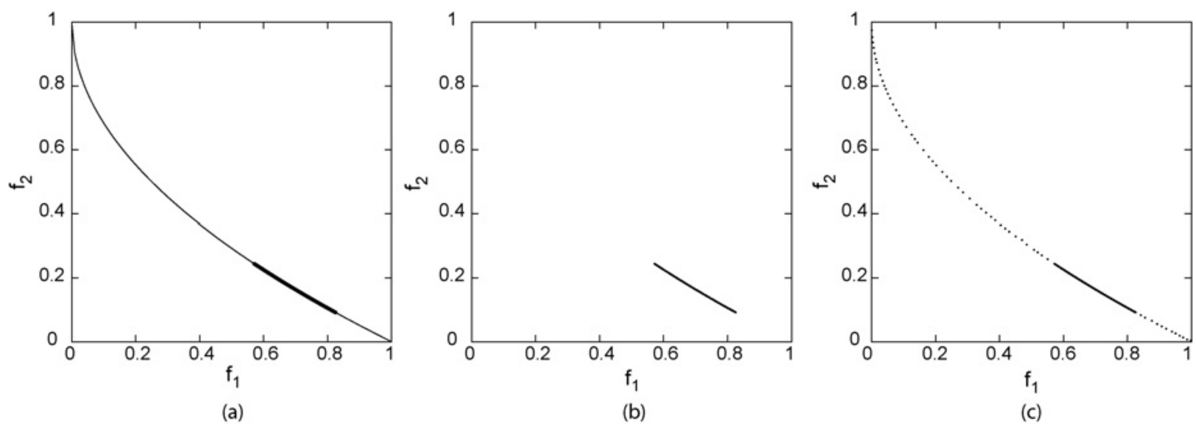


Fig. 22. ZDT4 test 3 plots. (a) Preference region 3. (b) Filtered TDEA. (c) prTDEA.

TABLE XXVII

STATISTICAL TESTS FOR ZDT4 PREFERENCE TESTS

$H_0 : \mu_{prTDEA} = \mu_{TDEA}$ versus $H_1 : \mu_{prTDEA} \neq \mu_{TDEA}$									
	Hypervolume			Inv. Gen. Dist.			Additive Epsilon		
Test	Δ_H	p-Value	Winner	Δ_D	p-Value	Winner	Δ_I	p-Value	Winner
ZDT4-1	0.0067	0	prTDEA	−0.000031	0	prTDEA	−0.0009	0	prTDEA
ZDT4-2	0.0022	0	prTDEA	−0.000007	0	prTDEA	−0.0006	0	prTDEA
ZDT4-3	0.0086	0.001	prTDEA	−0.000069	0.058	None	−0.0034	0.035	None
ZDT4-4	0.0016	0	prTDEA	−0.000008	0	prTDEA	−0.0005	0	prTDEA

2) *Test 2:* We expand the preference region in Test 1 to form this test [Fig. 25(a) and (d)]. We observe that the differences between their metric values are smaller compared to those in Test 1, since the preference region is larger. However, prTDEA's superiority over TDEA persists as seen in Table XXIX and XXXII. We present sample plots of the results in Fig. 25 for both algorithms.

3) *Test 3:* Here we assign each objective different weight ranges and test whether this creates difficulties in focusing on a preferred region. prTDEA has no problems in finding better approximations than TDEA, with a noticeably shorter average computational time (Table XXX). The statistical tests in Table XXXII indicate significance. The difference can also be seen from the sample plots given in Fig. 26.

4) *Test 4:* The last test is similar to Test 3, but it features a smaller preference region as shown in Fig. 27(a) and (d). We expect the relative performance of prTDEA to further improve in both metrics, as well as in the average computational time, since the region is smaller than that of Test 3. The indicator results given in Table XXXI confirm our expectation. We present sample plots in Fig. 27 to illustrate the difference.

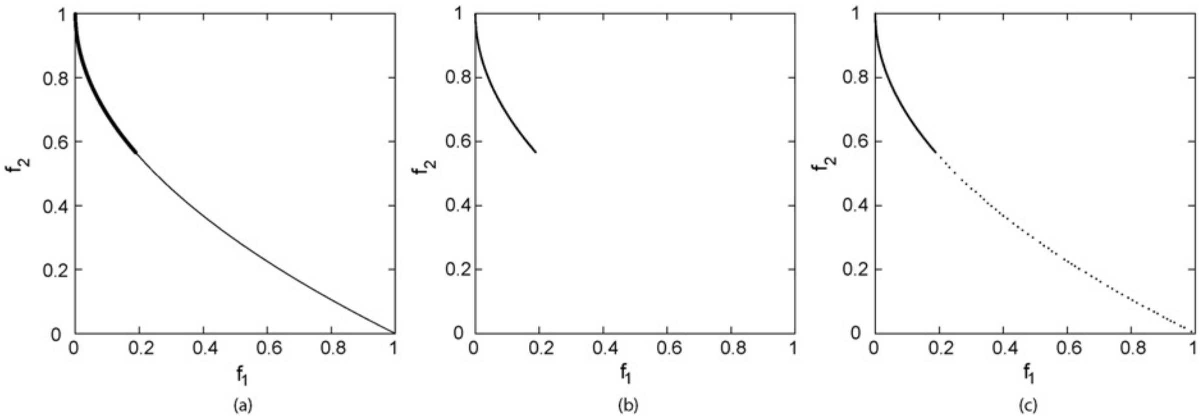


Fig. 23. ZDT4 Test 4 plots. (a) Preference region 4. (b) Filtered TDEA. (c) prTDEA.

TABLE XXVIII
INDICATOR RESULTS FOR DTLZ1 AND DTLZ2 PREFERENCE TEST 1

Test	Algorithm	Hypervolume $\bar{x}_H$	s_H	Inv. Gen. Dist. $\bar{x}_D$	s_D	Additive Epsilon $\bar{x}_I$	s_I	Time (s)
DTLZ1-1	prTDEA	0.2955	0.0024	0.000257	0.000011	0.0035	0.0004	6.03
	TDEA	0.2775	0.0063	0.000407	0.000053	0.0058	0.0007	33.01
	True Pareto	0.3119	–	0	–	0	–	–
DTLZ2-1	prTDEA	0.2671	0.0005	0.000204	0.000002	0.0125	0.0010	9.86
	TDEA	0.2559	0.0009	0.000252	0.000004	0.0178	0.0014	40.24
	True Pareto	0.2840	–	0	–	0	–	–

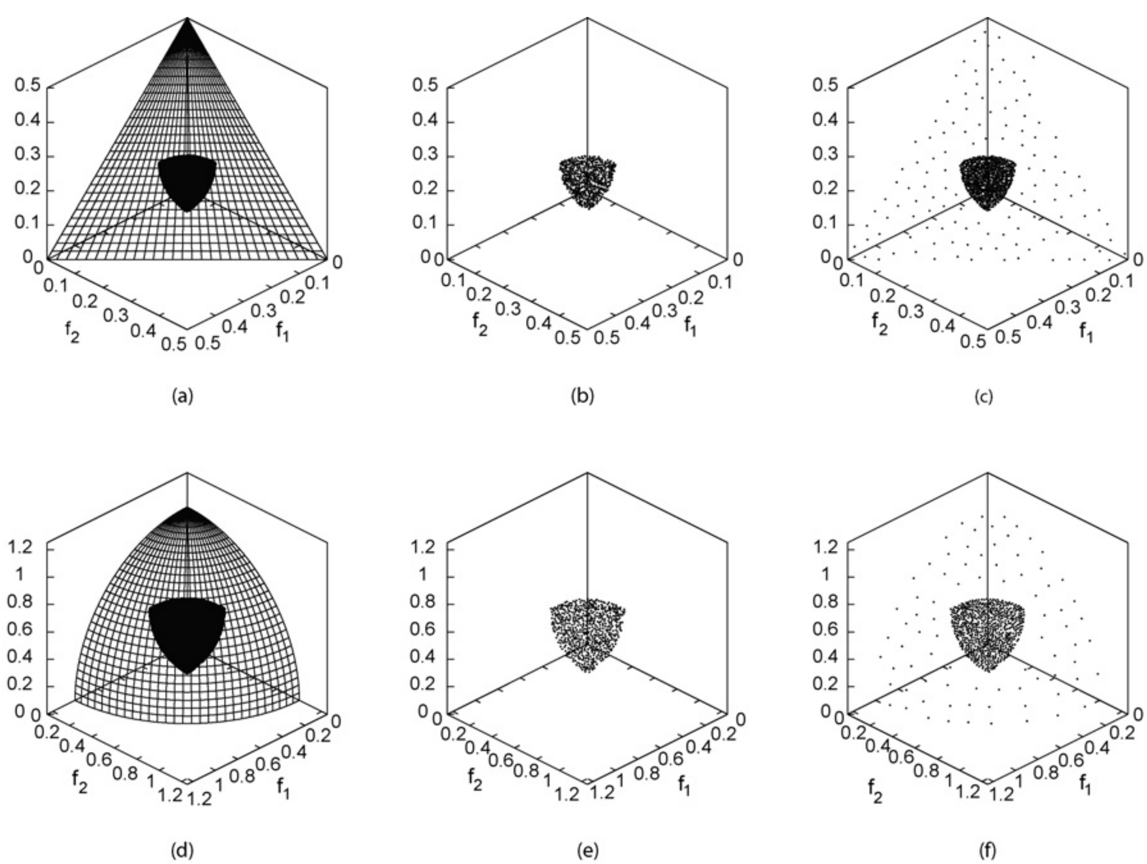


Fig. 24. DTLZ1 and DTLZ2 test 1 plots. (a) DTLZ1 preference region 1. (b) DTLZ1: filtered TDEA. (c) DTLZ1-prTDEA. (d) DTLZ2 preference region 1. (e) DTLZ2-filtered TDEA. (f) DTLZ2-prTDEA.

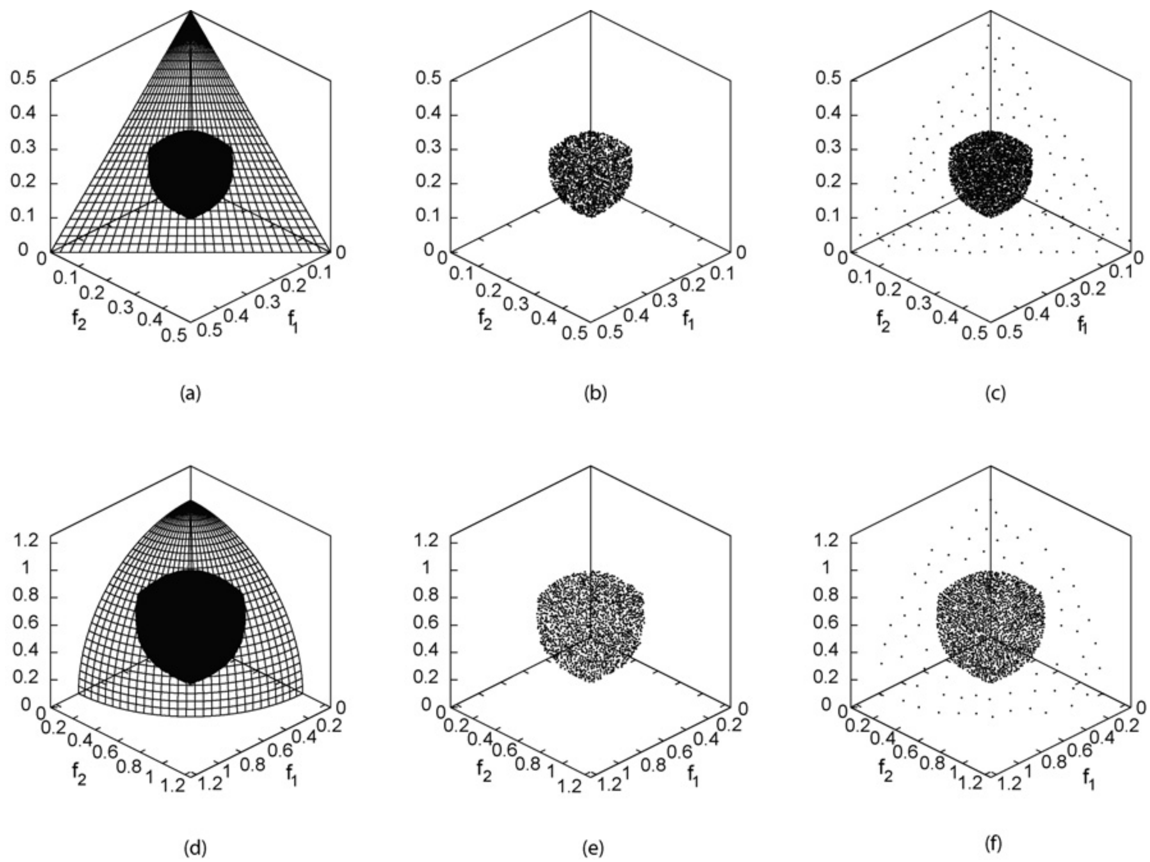


Fig. 25. DTLZ1 and DTLZ2 test 2 plots. (a) DTLZ1 preference region 2. (b) DTLZ1-filtered TDEA. (c) DTLZ1-prTDEA. (d) DTLZ2 preference region 2. (e) DTLZ2-filtered TDEA. (f) DTLZ2-prTDEA.

TABLE XXIX
INDICATOR RESULTS FOR DTLZ1 AND DTLZ2 PREFERENCE TEST 2

Test	Algorithm	Hypervolume $\bar{x}_H$	Hypervolume s_H	Inv. Gen. Dist. $\bar{x}_D$	Inv. Gen. Dist. s_D	Additive Epsilon $\bar{x}_I$	Additive Epsilon s_I	Time (s)
DTLZ1-2	prTDEA	0.3982	0.0023	0.000115	0.000009	0.0041	0.0003	10.46
	TDEA	0.3893	0.0045	0.000160	0.000021	0.0058	0.0007	33.01
	True Pareto	0.4139	–	0	–	0	–	–
DTLZ2-2	prTDEA	0.3453	0.0006	0.000091	0.000001	0.0143	0.0013	17.09
	TDEA	0.3395	0.0008	0.000106	0.000001	0.0180	0.0016	40.24
	True Pareto	0.3607	–	0	–	0	–	–

TABLE XXX
INDICATOR RESULTS FOR DTLZ1 AND DTLZ2 PREFERENCE TEST 3

Test	Algorithm	Hypervolume $\bar{x}_H$	Hypervolume s_H	Inv. Gen. Dist. $\bar{x}_D$	Inv. Gen. Dist. s_D	Additive Epsilon $\bar{x}_I$	Additive Epsilon s_I	Time (s)
DTLZ1-3	prTDEA	0.5549	0.0031	0.000096	0.000009	0.0043	0.0006	10.36
	TDEA	0.5463	0.0040	0.000129	0.000014	0.0058	0.0006	33.01
	True Pareto	0.5698	–	0	–	0	–	–
DTLZ2-3	prTDEA	0.3559	0.0004	0.000085	0.000001	0.0139	0.0012	16.04
	TDEA	0.3493	0.0006	0.000100	0.000001	0.0186	0.0029	40.24
	True Pareto	0.3684	–	0	–	0	–	–

TABLE XXXI
INDICATOR RESULTS FOR DTLZ1 AND DTLZ2 PREFERENCE TEST 4

Test	Algorithm	Hypervolume $\bar{x}_H$	Hypervolume s_H	Inv. Gen. Dist. $\bar{x}_D$	Inv. Gen. Dist. s_D	Additive Epsilon $\bar{x}_I$	Additive Epsilon s_I	Time (s)
DTLZ1-4	prTDEA	0.4451	0.0024	0.000198	0.000007	0.0035	0.0003	6.70
	TDEA	0.4304	0.0059	0.000285	0.000034	0.0056	0.0006	33.01
	True Pareto	0.4600	–	0	–	0	–	–
DTLZ2-4	prTDEA	0.3366	0.0008	0.000180	0.000001	0.0118	0.0012	10.35
	TDEA	0.3242	0.0011	0.000231	0.000003	0.0182	0.0028	40.24
	True Pareto	0.3491	–	0	–	0	–	–

TABLE XXXII
STATISTICAL TESTS FOR DTLZ1 AND DTLZ2 PREFERENCE TESTS

$H_0 : \mu_{prTDEA} = \mu_{TDEA}$ versus $H_1 : \mu_{prTDEA} \neq \mu_{TDEA}$

Test	Hypervolume Δ_H	Hypervolume p-Value	Hypervolume Winner	Inv. Gen. Dist. Δ_D	Inv. Gen. Dist. p-Value	Inv. Gen. Dist. Winner	Additive Epsilon Δ_I	Additive Epsilon p-Value	Additive Epsilon Winner
DTLZ1-1	0.0180	0	prTDEA	−0.000150	0	prTDEA	−0.0022	0	prTDEA
DTLZ1-2	0.0089	0	prTDEA	−0.000045	0	prTDEA	−0.0017	0	prTDEA
DTLZ1-3	0.0086	0	prTDEA	−0.000032	0	prTDEA	−0.0015	0	prTDEA
DTLZ1-4	0.0147	0	prTDEA	−0.000087	0	prTDEA	−0.0021	0	prTDEA
DTLZ2-1	0.0112	0	prTDEA	−0.000048	0	prTDEA	−0.0053	0	prTDEA
DTLZ2-2	0.0058	0	prTDEA	−0.000015	0	prTDEA	−0.0037	0	prTDEA
DTLZ2-3	0.0066	0	prTDEA	−0.000015	0	prTDEA	−0.0047	0	prTDEA
DTLZ2−4	0.0124	0	prTDEA	−0.000051	0	prTDEA	−0.0064	0	prTDEA

TABLE XXXIII
INDICATOR RESULTS FOR DTLZ1-5D AND DTLZ2-5D PREFERENCE TEST 1

Test	Algorithm	Hypervolume $\bar{x}_H$	Hypervolume s_H	Inv. Gen. Dist. $\bar{x}_D$	Inv. Gen. Dist. s_D	Additive Epsilon $\bar{x}_I$	Additive Epsilon s_I	Time (s)
DTLZ1-5D-1	prTDEA	0.4422	0.0077	0.002621	0.000056	0.0147	0.0008	30.94
	TDEA	0.2607	0.0515	0.005721	0.001370	0.0334	0.0084	196.62
	True Pareto	0.4529	–	0	–	0	–	–
DTLZ2-5D-1	prTDEA	0.4217	0.0025	0.001449	0.000010	0.0691	0.0035	60.91
	TDEA	0.3549	0.0040	0.001840	0.000022	0.0863	0.0040	502.86
	True Pareto	0.4739	–	0	–	0	–	–

TABLE XXXIV
INDICATOR RESULTS FOR DTLZ1-5D AND DTLZ2-5D PREFERENCE TEST 2

Test	Algorithm	Hypervolume $\bar{x}_H$	Hypervolume s_H	Inv. Gen. Dist. $\bar{x}_D$	Inv. Gen. Dist. s_D	Additive Epsilon $\bar{x}_I$	Additive Epsilon s_I	Time (s)
DTLZ1-5D-2	prTDEA	0.2752	0.0069	0.004306	0.000098	0.0145	0.0012	18.72
	TDEA	0.1248	0.0326	0.009845	0.003102	0.0368	0.0133	196.62
	True Pareto	0.2840	–	0	–	0	–	–
DTLZ2-5D-2	prTDEA	0.2817	0.0020	0.002187	0.000015	0.0657	0.0046	36.16
	TDEA	0.2112	0.0029	0.002862	0.000031	0.0883	0.0054	502.86
	True Pareto	0.3195	–	0	–	0	–	–

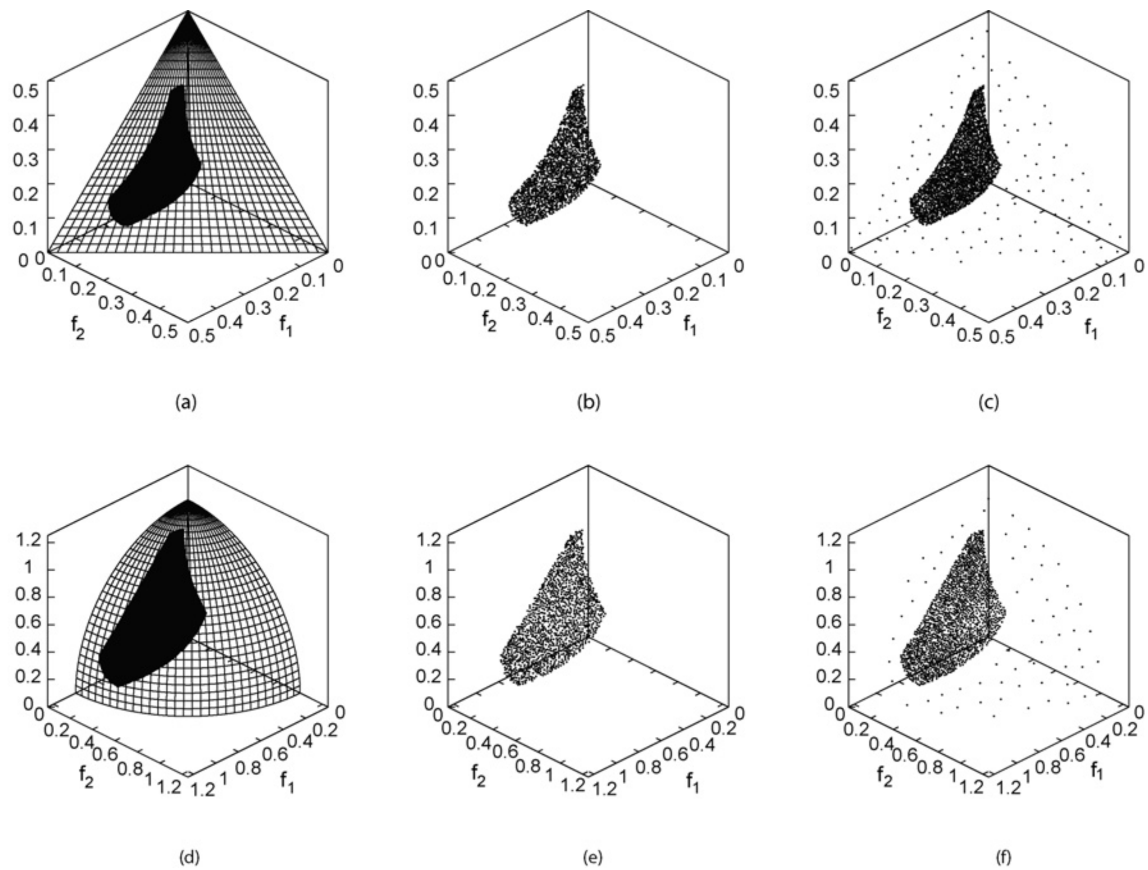


Fig. 26. DTLZ1 and DTLZ2 Test 3 plots. (a) DTLZ1 preference region 3. (b) DTLZ1-filtered TDEA. (c) DTLZ1-prTDEA. (d) DTLZ2 preference region 3. (e) DTLZ2-filtered TDEA. (f) DTLZ2-prTDEA.

TABLE XXXV
STATISTICAL TESTS FOR DTLZ1-5D AND DTLZ2-5D PREFERENCE TESTS

$H_0 : \mu_{prTDEA} = \mu_{TDEA}$ versus $H_1 : \mu_{prTDEA} \neq \mu_{TDEA}$									
	Hypervolume			Inv. Gen. Dist.			Additive Epsilon		
Test	Δ_H	p-Value	Winner	Δ_D	p-Value	Winner	Δ_I	p-Value	Winner
DTLZ1-5D-1	0.1815	0	prTDEA	−0.003099	0	prTDEA	−0.0187	0	prTDEA
DTLZ1-5D-2	0.1504	0	prTDEA	−0.005539	0	prTDEA	−0.0223	0	prTDEA
DTLZ2-5D-1	0.0667	0	prTDEA	−0.000391	0	prTDEA	−0.0172	0	prTDEA
DTLZ2-5D-2	0.0704	0	prTDEA	−0.000675	0	prTDEA	−0.0226	0	prTDEA

C. 5-Objective Tests

The 5-objective tests are conducted on DTLZ1 and DTLZ2 problems. As in 3-objective tests, the same desirable regions are used for both problems.

Test 1. The preference region of the first test is in the middle of the Pareto-optimal frontier. Table XXXIII shows that prTDEA clearly outperforms TDEA in all three metrics. prTDEA's hypervolume values are very close to the true Pareto-optimal regions' values. On the other hand, TDEA's values are substantially lower. Table XXXV shows that the differences are statistically significant. We also observe that the average time to complete the runs is significantly lower in prTDEA.

Test 2. In this test, the preference region is slightly biased toward the first objective. The metric results in Table XXXIV are similar to those observed in the previous test. prTDEA scores much better in the hypervolume metric than TDEA. The differences are statistically significant as shown in Table XXXV. We observe that the difference between the two algorithms increases as the number of objectives increases.

D. Discussions

The test results show that focusing on a preference region provides both a better approximation and a substantial computational time advantage in all settings. The positive effects of focusing are clearer when the region is smaller. In addition, the metric and computation time differences get larger when the number of objectives increases. This is because the number of individuals needed to properly approximate the Pareto-

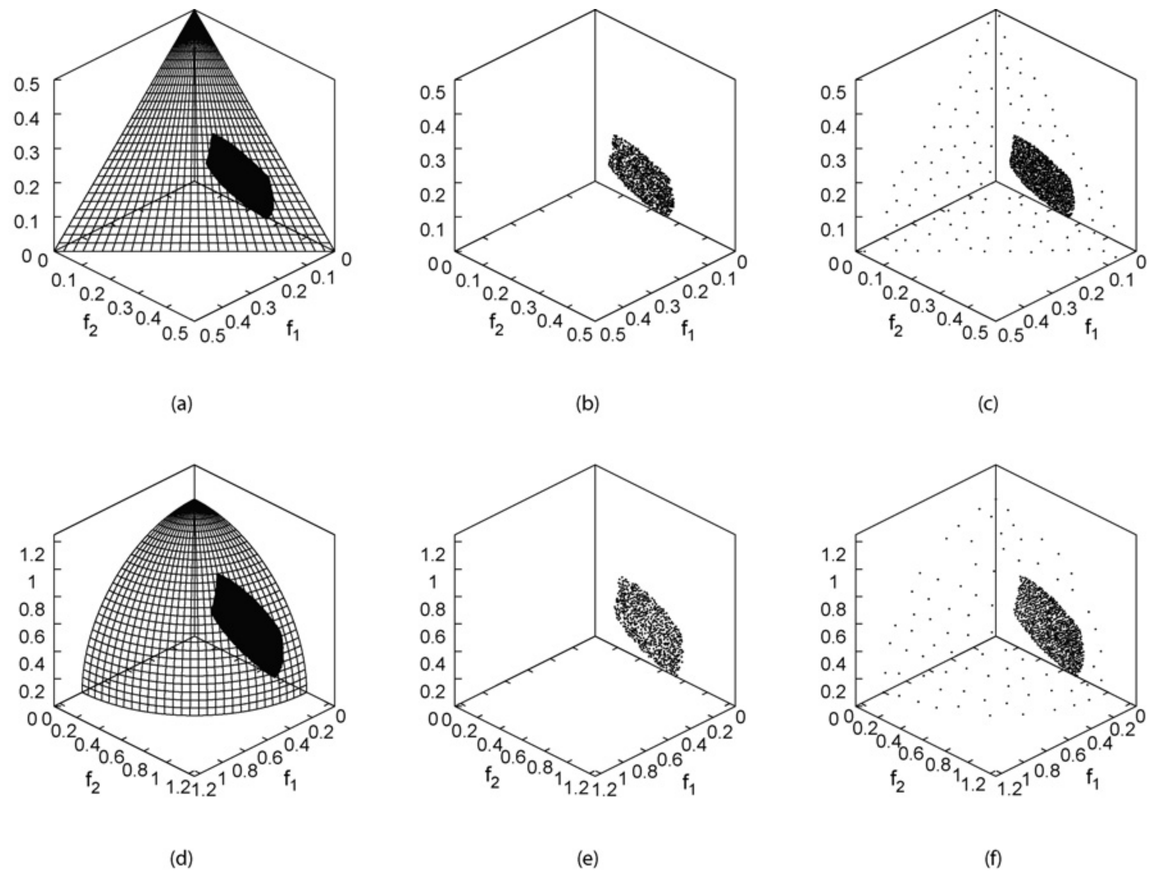


Fig. 27. DTLZ1 and DTLZ2 Test 4 plots. (a) DTLZ1 preference region 4. (b) DTLZ1-filtered TDEA. (c) DTLZ1-prTDEA. (d) DTLZ2 preference region 4. (e) DTLZ2-filtered TDEA. (f) DTLZ2-prTDEA.

optimal frontier increases with the number of objectives. We can conclude that DM preferences should be exploited whenever possible. prTDEA's preference incorporation mechanism effectively serves this purpose. Note that it does not lose the rest of the Pareto-optimal frontier while concentrating on these regions as shown in the sample plots of the tests. It displays an overview of the entire Pareto-optimal frontier, while presenting finely detailed individuals in the regions of interest. Thus, the DM will be able to get information about the extent of the whole Pareto-optimal frontier, while getting a much finer detail of the preferred regions.

VI. Conclusion

In this paper, we proposed a new multiobjective evolutionary algorithm, the Territory Defining Evolutionary Algorithm and tested its performance against well-known MOEAs in the literature. We discuss how the territory defining property of the algorithm helps preserve diversity and allows fast execution.

We tested the algorithm on 2, 3, and 5-objective problems, each having different characteristics. We observe that TDEA performs well on the hypervolume, the inverted generational distance and the additive ϵ-indicator performance metrics in all problems. In some problems, it outperforms other algorithms in all metrics. In addition, it is the second fastest algorithm among all contenders.

TDEA is simple to implement and obtains well-spread final populations without the need for a computationally demanding diversity operator. Instead, the algorithm makes use of its territory defining property and reduces computation time requirements substantially.

As in any evolutionary algorithm, there may be many different ways in implementing the details of our algorithm. One such variation we intend to implement is related to the archive updating procedure. Currently, the members dominated by a candidate solution are eliminated. Alternatively, those dominated solutions can be maintained in the archive if the candidate solution is not admitted to the archive. This may help keep good solutions. How well this variation may work awaits further testing.

While TDEA is very successful in approximating the entire Pareto-optimal frontier, we have an additional goal of providing a mechanism that can exploit the DM's preferences when they are available. For this purpose, we introduce the preference-based TDEA that incorporates a-priori preference information of the DM to guide the search toward more desirable regions. In prTDEA, we utilize the territory defining property to set different territories and favorable weights. We tested the algorithm on various test problems with 2, 3, and 5-objective problems using different preference regions. The experiments showed that preference incorporation leads to better approximation of the desired regions, while maintaining a good sample from the rest of the Pareto-optimal frontier. Also, we demonstrate that focusing reduces the computational effort substantially. We observe that the effects of preference incorporation increase as the number of objectives increases. This is an expected result since the Pareto-frontier enlarges with increasing number of objectives and it becomes harder to represent the whole frontier with a limited population.

REFERENCES

[1] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.

[2] P. A. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.

[3] J. Branke, T. Kaußler, and H. Schmeck, "Guidance in evolutionary multiobjective optimization," *Advances Eng. Softw.*, vol. 32, no. 6, pp. 499–507, Jun. 2001.

[4] C. A. C. Coello, "An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends," in *Proc. Congr. Evol. Comput.*, vol. 1. Washington D.C., 1999, pp. 3–13.

[5] C. A. C. Coello, "Handling preferences in evolutionary multiobjective optimization: A survey," in *Proc. Congr. Evol. Comput.*, Piscataway, NJ, 2000, pp. 30–37.

[6] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Springer-Verlag, 2006.

[7] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proc. Genetic Evol. Comput. Conf. (GECCO)*, San Francisco, CA: Morgan Kaufmann, 2001, pp. 283–290.

[8] D. Cvetković and I. C. Parmee, "Preferences and their application in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 42–57, Feb. 2002.

[9] K. Deb, "Evolutionary algorithms for multicriterion optimization in engineering design," in *Evolutionary Algorithms in Engineering and Computer Science*. Chichester, U.K.: Wiley, 1999, pp. 135–161.

[10] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[11] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1985.

[12] K. Deb and M. Goyal, "A combined genetic adaptive search (geneas) for engineering design," *Comput. Sci. Informat.*, vol. 26, no. 4, pp. 30–45, 1996.

[13] K. Deb and A. Kumar, "Interactive evolutionary multiobjective optimization and decision-making using reference direction method," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, vol. 1. London, U.K.: ACM, 2007, pp. 781–788.

[14] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ϵ-domination based multiobjective evolutionary algorithm for a quick computation of pareto-optimal solutions," *Evol. Comput.*, vol. 13, no. 4, pp. 501–525, 2005.

[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[16] K. Deb and J. Sundar, "Reference point based multiobjective optimization using evolutionary algorithms," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, vol. 1. Seattle, WA: ACM, 2006, pp. 635–642.

[17] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," Comput. Eng. Netw. Lab., Swiss Federal Instit. Technol., Zurich, Switzerland, Tech. Rep. 112, 2001.

[18] J. J. Durillo, A. J. Nebro, F. Luna, B. Dorronsoro, and E. Alba, "jMetal: A Java framework for developing multiobjective optimization metaheuristics," Dept. Lenguajes Ciencias Computación, Univ. Málaga ETSI Informática, Málaga, Spain, Tech. Rep. ITI-2006-10, 2006.

[19] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evol. Comput.*, vol. 3, no. 1, pp. 1–16, 1995.

[20] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms-part i: A unified formulation," *IEEE Trans. Syst. Man Cybern. A, Syst. Humans*, vol. 28, no. 1, pp. 26–37, Jan. 1998.

[21] C. K. Goh and K. C. Tan, "An investigation on noisy environments in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 3, pp. 354–381, Jun. 2007.

[22] İ. Karahan, "Preference-based flexible multiobjective evolutionary algorithms," M.S. thesis, Dept. Ind. Eng., Middle East Tech. Univ., Ankara, Turkey, 2008.

[23] M. Köksalan and S. P. Phelps, "An evolutionary metaheuristic for approximating preference-nondominated solutions," *Informs J. Comput.*, vol. 19, no. 2, pp. 291–301, 2007.

[24] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, 2000.

[25] P. J. Korhonen and J. Laakso, "A visual interactive method for solving the multiple criteria problem," *Eur. J. Oper. Res.*, vol. 24, no. 2, pp. 277–287, 1986.

[26] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 263–282, 2002.

[27] S. P. Phelps and M. Köksalan, "An interactive evolutionary metaheuristic for multiobjective combinatorial optimization," *Manage. Sci.*, vol. 49, no. 12, pp. 1726–1738, 2003.

[28] L. Rachmawati and D. Srinivasan, "Preference incorporation in multiobjective evolutionary algorithms: A survey," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, 2006, pp. 3385–3391.

[29] B. Soylu and M. Köksalan, "A favorable weight based evolutionary algorithm for multiple criteria problems," *IEEE Trans. Evol. Comput.*, 2009, to be published.

[30] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation and Application*. New York: Wiley, 1986.

[31] H. Tamaki, H. Kita, and S. Kobayashi, "Multiobjective optimization by genetic algorithms: A review," in *Proc. Int. Conf. Evol. Comput. (ICEC)*, Nagoya, Japan, 1996, pp. 517–522.

[32] K. C. Tan, E. F. Khor, T. H. Lee, and R. Sathikannan, "An evolutionary algorithm with advanced goal and priority specification for multiobjective optimization," *J. Artif. Intell. Res.*, vol. 18, no. 1, pp. 183–215, Jan. 2003.

[33] D. A. V. Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithms: Analyzing the state-of-the-art," *Evol. Comput.*, vol. 8, no. 2, pp. 125–147, 2000.

[34] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. Parallel Problem Solving Nature (PPSN)*, Lecture Notes in Computer Science 3242. Birmingham, U.K.: Springer-Verlag, 2004, pp. 832–842.

[35] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," in *Proc. EUROGEN 2001: Evolutionary Methods Design Optimization Control Appl. Ind. Problems*, Athens, Greece, 2002, pp. 95–100.

[36] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms: A comparative case study," in *Proc. Parallel Problem Solving Nature (PPSN V)*, Lecture Notes in Computer Science 1498. London, U.K.: Springer-Verlag, 1998, pp. 292–301.

İbrahim Karahan received the B.S. and M.S. degrees from the Department of Industrial Engineering, Middle East Technical University, Ankara, Turkey, in 2006 and 2008, respectively. He is currently pursuing the M.S. degree in systems and entrepreneurial engineering at the University of Illinois, Urbana-Champaign.

He was a Research Assistant at Middle East Technical University from 2006 to 2008. He has been working as a Teaching Assistant with the University of Illinois since 2008. His research interests include multiobjective optimization, evolutionary algorithms, combinatorial optimization, and decision analysis.

Murat Köksalan received the B.S. and M.S. degrees in industrial engineering from the Department of Industrial Engineering, Middle East Technical University, Ankara, Turkey, in 1979 and 1980, respectively, and the Ph.D. degree in industrial engineering from the Department of Industrial Engineering, State University of New York (SUNY), Buffalo, in 1984.

He is a Professor at the Department of Industrial Engineering, Middle East Technical University. He was a Visiting Professor at SUNY, Purdue University, West Lafayette, IN, Helsinki University of Technology, Helsinki, Finland, and the Helsinki School of Economics, Helsinki, Finland, on various occasions. His research interests include multiple criteria decision making, combinatorial optimization, heuristic search, combinatorial auctions, and preparing teaching cases.

Dr. Köksalan received the Young Researcher Award from the Scientific and Technological Research Council of Turkey, and the Multiple Criteria Decision Making Gold Medal from the International Society on Multiple Criteria Decision Making. He won the First Prize from the Institute for the Operations Research and the Management Sciences Case Competition with various co-authors, three times.