An Interactive Territory Defining Evolutionary Algorithm: iTDEA

Murat Köksalan and İbrahim Karahan

Abstract—**We develop a preference-based multiobjective evolutionary algorithm that interacts with the decision maker (DM) during the course of optimization. We create a territory around each solution where no other solutions are allowed. We define smaller territories around the preferred solutions in order to obtain denser coverage of these regions. At each interaction, the algorithm asks the DM to choose his/her best solution among a set of representative solutions to guide the search toward the neighborhood of the selected solution. The algorithm aims to converge to a final preferred region of the DM. We test the algorithm on three problems using three different utility function types to simulate the DM's responses. The results show that the algorithm converges the DM's simulated preferred regions well.**

Index Terms—**Evolutionary algorithms, guidance, interactive, multiobjective optimization, preference incorporation.**

I. Introduction

In REAL-LIFE problems, one usually needs to consider multiple conflicting objectives simultaneously. To handle such problems, many multiobjective evolutionary algorithms (MOEAs) have been proposed in the literature. Since MOEAs work with a population of solutions, they can generate multiple solutions in a single run. In addition, they offer substantial computational savings compared to traditional mathematical modeling-based approaches, especially in combinatorial problems. These characteristics have made MOEAs popular among researchers for over two decades. Many successful MOEAs, such as Pareto archived evolution strategy [1], Pareto Envelope Based Selection II [2], nondominated sorting GA-II (NSGA-II) [3], strength Pareto evolutionary algorithm 2 (SPEA2) [4], indicator based evolutionary algorithm (IBEA) [5], ϵ-MOEA [6], S-metric selection evolutionary multiobjective algorithm [7], and favorable weight-based evolutionary algorithm [8] have been proposed in the literature. MOEA literature has been reviewed by [9]–[13]. In addition, [14] and [15] cover many aspects of evolutionary algorithms in multiobjective optimization. A list of references to publications in the multiobjective evolutionary optimization area is maintained by Coello at http://www.lania.mx/~ccoello/EMOO/EMOObib.html.

Manuscript received August 31, 2009; revised January 19, 2010 and April 27, 2010; accepted July 30, 2010. Date of current version October 1, 2010. This paper was recommended by Associate Editor I. Parmee.

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

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

Digital Object Identifier 10.1109/TEVC.2010.2070070

Most MOEAs try to approximate the entire Pareto-optimal frontier. Although such a task may be manageable for two objectives, it is difficult for more objectives with a limited population. Also as [16] claims, this does not provide any insight into the decision making process. Usually, it is possible for the decision maker (DM) to provide information about his/her preferences. He/she may prefer some regions of the Pareto-optimal frontier over other regions. Solutions in some regions of the frontier may be very poor from the DM's perspective. Utilizing such preference information allows concentrating the search efforts at the desired regions and approximating these regions better. Limiting the search to these regions may also bring substantial computational savings. Recognizing these advantages, some researchers started to consider converging to preferred regions of Pareto-optimal frontier rather than trying to approximate the whole frontier. Review papers by [16]–[18] stress the importance of preference incorporation in MOEAs. There are various studies in the literature to guide the search in MOEAs to the interesting regions.

Several of the approaches obtain some preference information at the start and use this information to guide the algorithm toward the preferred region. Fonseca and Fleming [19] propose the preferability operator. This method allows the DM to set goals and priority levels for each objective. The algorithm uses a modified dominance scheme based on this operator to determine the ranks of the individuals, using the collected preference information. In [20], the DM specifies his/her tradeoffs between each pair of objectives. Then, the *minimal* and *maximal* utility functions are constructed and these functions are used for modifying the dominance scheme based on the DM's preferences. [21] proposes 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. [22] modifies the crowding distance operator of NSGA-II to incorporate preference information. In this method, the DM is asked to specify one or more reference points. Then the algorithm explores the neighborhood of the points to find a set of solutions close to them. Köksalan [23] develops an evolutionary metaheuristic for approximating preference-nondominated solutions (EMAPS) for the multiobjective combinatorial optimization 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. Deb and Kumar [24] utilize the light beam search technique to guide the search toward reference points provided by the DM. incorporate the idea into NSGA-II by modifying the crowding scheme and test it on various problems. The main drawback of these approaches is that they obtain some preference information at the start and rely solely on this information to guide the algorithm. In reality, the DM may not be well aware of the possible solutions, their objective values, or the approximate location of the Pareto-optimal frontier. As the DM gets information about these, he/she may learn more about his/her preferences and make more informed decisions within this context. In the multicriteria decision making (MCDM) literature, many interactive approaches have been developed over the last 40 years in order to help DMs make informed decisions [25]. These approaches obtain preference information from the DM progressively throughout the solution process to guide the search toward the preferred solutions.

More recently, interactive evolutionary computation literature for single objective problems has grown with a somewhat different emphasis [35]. The development of interactive approaches in the MOEA literature is rather recent. Parmee *et al.* [26] discuss the importance of interactive approaches for design processes. They propose to obtain *a priori* pairwise comparisons of objectives from the designer and convert these into weights of objectives. They also gave examples of how insights may be gained by designer/machine interaction. Phelps and Köksalan [27] develop the first interactive MOEA in the spirit of interactive MCDM approaches, to the best of our knowledge. They obtain the responses of the DM during the solution process in order to guide the algorithm toward his/her preferred regions. The algorithm is tested on multiobjective combinatorial optimization problems and good results are obtained. The algorithm requires the DM to compare pairs of alternatives and constructs an estimated utility function based on these preference information. The estimated utility function is one of the main tools to guide the algorithm toward preferred regions and this could be a handicap for some problems. Abbass [28] develops an interactive procedure to be used with evolutionary algorithms for biobjective problems. The approach incorporates visual interaction to help the DM choose desired points. It then solves a series of single objective optimization problems to find solutions in the direction of the points specified by the DM. Branke *et al.* [29] use a linear combination of objectives in an attempt to interactively guide the search toward preferred solutions. They demonstrate their approach on the test problem ZDT1, which has a convex Pareto-optimal frontier of two objectives. Linear combinations of objectives are known to be unsuitable for problems when the Pareto-optimal frontier is not convex. Furthermore, the approach requires solving linear programs after each interaction phase, which may cause a computational inefficiency. Other interactive approaches typically use achievement scalarizing functions and require the DM to provide reference points representing his/her preferences. Deb and Kumar [30] incorporate the ideas from the interactive approach of [31] into NSGA-II, and make use of *achievement scalarizing functions* to guide the search toward preferred regions. Thiele *et al.* [32] develop preference-based evolutionary algorithm, which uses the indicator-based evolutionary algorithm [5] as a basis. The authors incorporate an achievement scalarizing function into the indicator function to utilize the preference information provided by the DM interactively. Providing reference points that represent his/her preferences may not be an easy task for DMs. The DMs typically make decisions evaluating the outcome (criteria) values and comparing different alternatives.

In this paper, we address the problem of choosing a highly preferred solution for the DM by obtaining preference information from the DM progressively. Our approach is in the spirit of interactive approaches from the multiple criteria decision making field. We develop an interactive territory defining evolutionary algorithm (iTDEA) that utilizes the territory definition of the territory defining evolutionary algorithm (TDEA) [33]. The territory idea has been shown to work well in converging the Pareto-optimal frontier as well as focusing to desired parts of the frontier. Our current algorithm aims to converge to the preferred region by interacting with the DM during optimization. We ask the DM to compare several solutions and choose the one he/she prefers. Such an interaction mechanism has been considered to be relevant to DMs and has been used extensively in the multiple criteria decision making literature [25]. We use a weighted Tchebycheff function in order to concentrate the search around the Pareto-optimal region of the solutions preferred by the DM. The weighted Tchebycheff function has an important property that for each Pareto-optimal solution there exists a set of weights for which this solution outperforms all other solutions. This property makes the weighted Tchebycheff function capable of reaching any Pareto-optimal solution regardless of the shape of the Pareto-optimal frontier. We conduct experiments on various problems and with different utility functions and demonstrate that the algorithm works well.

This paper is organized as follows. In Section II, we present a review of TDEA and preference-region territory defining evolutionary algorithm (prTDEA) of [33]. In Section III, we develop iTDEA. We present computational experiments in Section IV and conclude in Section V

II. Review of TDEA and prTDEA

Karahan and Köksalan [33] introduce the TDEA (see Fig. 1). This algorithm maintains two populations. A regular population which may contain both dominated and nondominated individuals and an archive population which contains only nondominated individuals. While the regular population has a fixed size $\bar{N}$, the archive population is flexible. In each generation, a single offspring is created. If the offspring is dominated by any individual in the regular population, it is rejected and a new offspring is created. Otherwise, it is accepted into the regular population by replacing an individual it dominates (a random one if it does not dominate an individual). Then, it is evaluated to enter the archive population. An offspring dominated by any individual in the archive is rejected. Otherwise, individuals dominated by the offspring

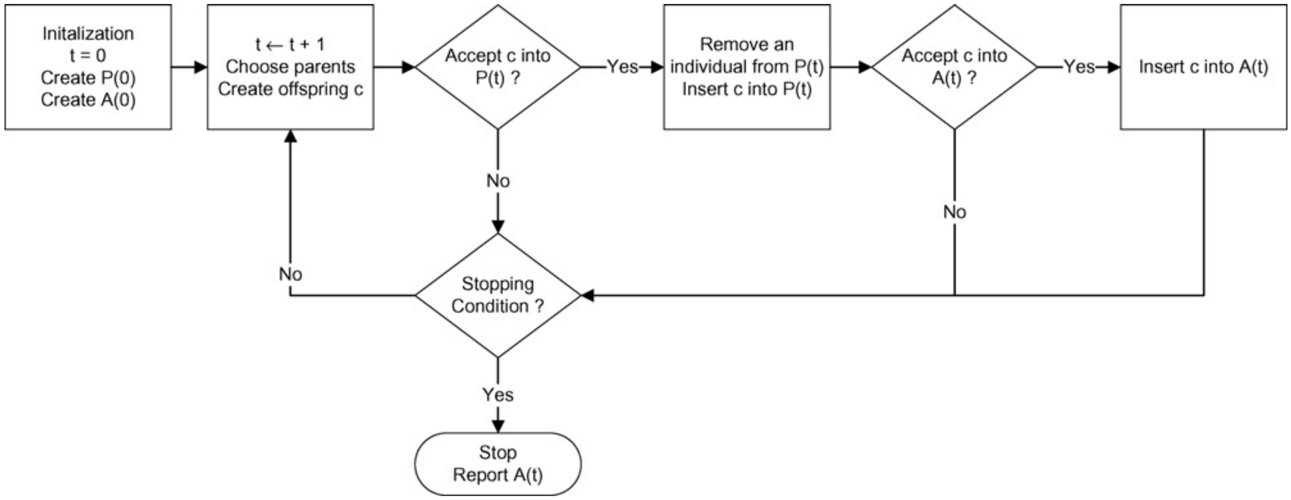

Fig. 1. Flowchart of TDEA.

are removed from the archive. Then, a *territory* of size τ is defined around the individual closest to the offspring. A nondominated offspring is only accepted if it does not violate this territory. If the offspring violates the defined territory, it is rejected to prevent crowding. This *territory defining property* maintains the diversity of the archive population. It works fast and an additional diversity preservation mechanism is not needed.

An illustration of the territory of an individual in 2-D space is given in Fig. 2. Territory can be roughly regarded as the space each individual occupies in the objective space. Each individual in the archive population controls a territory and disallows other individuals in its territory. Since territories can overlap, an individual may occupy smaller space than its territory in the objective space.

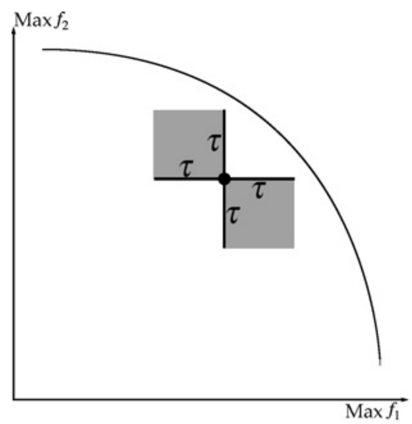

Fig. 2. Territory in two dimensions.

The authors demonstrate the effectiveness of territory preservation in maintaining the diversity on various problems. They show that the territory size parameter τ can be used to control the number of individuals in the final archive population. Based on this idea, they present a version of TDEA, prTDEA, which includes a mechanism to incorporate the DM's preferences into the search. This mechanism modifies the territory size of an individual depending on its location on the Pareto-optimal frontier. Individuals in those regions that are appealing to the DM are assigned smaller territories. On the other hand, individuals located elsewhere have larger territories. In this way, the densities of desired regions are allowed to be higher than other regions, leading to better approximations. An illustration is given in Fig. 3. In addition, individuals in desired regions have more chance to participate in genetic operations, since they outnumber individuals in other regions. Consequently, search in desired regions is encouraged. At the end, the algorithm finds a detailed approximation of the desired regions without increasing the number of individuals in other regions.

TDEA's territory preservation is similar to the hyperbox mechanism of ϵ-MOEA [6]. In ϵ-MOEA, the objective space is divided into hyperboxes of prespecified size. Crowding is prevented by allowing each hyperbox to hold at most one individual. The main difference between hyperboxes and territories is their locations. While the territory of an individual depends on the individual's location, hyperboxes are fixed before optimization. Thus, ϵ-MOEA is more prone to losing individuals toward the extremes of the Pareto-optimal frontier. In ϵ-MOEA, individuals close to the extremes ϵ-dominate the rest of extreme regions and disallow other individuals in those regions. On the other hand, territories in TDEA disallow individuals only in a limited region, circumventing ϵ-MOEA's shortcomings. This is illustrated in Fig. 4 using identical hyperbox and territory sizes.

Karahan and Köksalan [33] compare TDEA with leading MOEAs (ϵ-MOEA, IBEA, NSGA-II, SPEA2) and show on extensive tests that TDEA outperforms them. They also show that the focusing mechanism in prTDEA further improves the results both in terms of the quality of solutions obtained in the focused region and the computational time. That is, prTDEA brings important improvements over TDEA if the goal is to find good solutions in a specific region. Based on these observations, we develop an interactive approach that aims to converge the preferred solutions of the DM by progressively obtaining preference information from the DM.

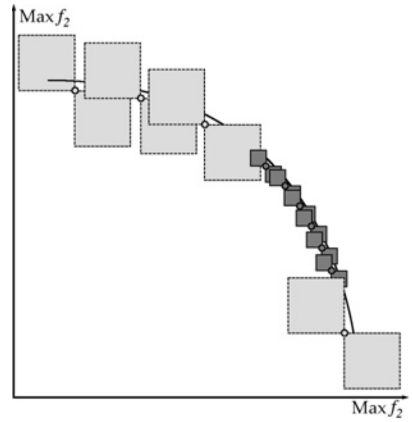

Fig. 3. Different territory sizes.

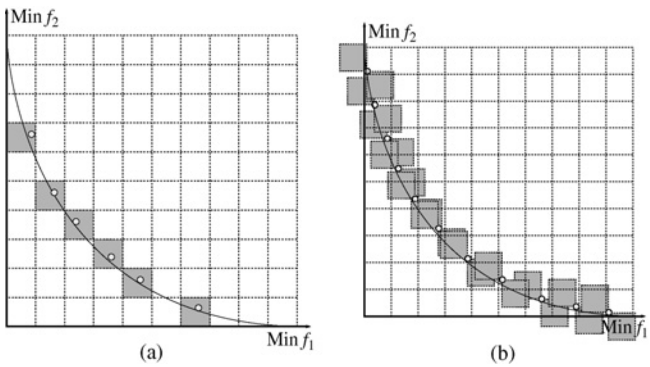

Fig. 4. Hyperbox vs. territory. (a) ϵ-MOEA. (b) TDEA.

III. Development of iTDEA

Our algorithm, iTDEA, identifies the region appealing to the DM during optimization by interacting with the DM. It starts by finding individuals over the entire Pareto-optimal frontier. At a predetermined generation, it pauses and presents a small representative sample of individuals found so far to the DM. Among these, the DM chooses the one that he/she prefers the most. If the DM is indifferent between several solutions, they may all be considered as best solutions. These individuals are then used for estimating the first preferred regions and the algorithm continues to run. Individuals falling in these regions are assigned smaller territories than those located elsewhere, so that the density of the preferred regions is higher. After some generations, the algorithm pauses and presents a sample of individuals to the DM again. The second preferred regions are then estimated in the neighborhood of the individuals chosen by the DM and the algorithm continues. To prioritize each preferred region over the previously created ones, the individuals falling in the most recently created regions are assigned smaller territories. In addition, the size of preferred regions is made smaller than the previous ones so that the algorithm converges to a final preferred region.

At interaction stages, the DM is presented several solutions to choose from. In order to keep the information load reasonable and avoid user fatigue, we suggest to keep the number of solutions presented to the DM small and we suggest to employ few interaction stages. The required information type is also easy since comparing several real solutions among themselves is a task the DMs are familiar with.

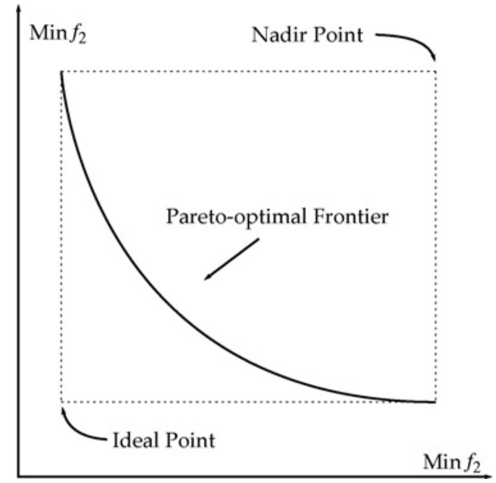

Fig. 5. Ideal and nadir vectors.

A. Definitions

A multiobjective optimization problem (MOP) has multiple objective functions, each is either maximized or minimized. In its general form, a MOP can be formulated as follows:

$$\text{“Minimize”} \quad \mathbf{y} = (f_1(\mathbf{x}), \ldots, f_i(\mathbf{x}), \ldots, f_m(\mathbf{x})) \tag{1}$$

$$\text{subject to} \quad \mathbf{x} \in \mathbf{X} \tag{2}$$

where

$\mathbf{x} = (x_1, \ldots, x_n)$	decision vector
$\mathbf{X} \subseteq \Re^n$	feasible decision space
$\mathbf{Y} \subseteq \Re^m$	feasible objective space (solution space)
$\mathbf{y} \in \mathbf{Y}$	m-vector of objective values.

Unlike single objective optimization problems, most MOPs do not have a single solution that optimizes all m objectives. Instead, the search is for finding those solutions for which improvement in one objective can only occur with the worsening of at least one other objective. These are called *nondominated solutions*.

Consider a problem with m objectives to be minimized. Let $y_i = (f_{i1}, f_{i2}, \ldots, f_{im})$ denote a solution i having an objective function value of f_{ij} in objective j. Solution y_1 is said to *dominate* solution y_2 if $f_{1j} \leq f_{2j}$ for $j = 1, 2, \ldots, m$ and $f_{1j} < f_{2j}$ for at least one j. If there exists no solution dominating y_1, then y_1 is said to be *nondominated* or *Pareto optimal*.

Let $y^{**} = \left[f_1^{**}, f_2^{**}, \ldots, f_m^{**}\right]^T$ be a vector of objective values whose element j is the optimal value of the jth objective function among all nondominated solutions, and $y^w = \left[f_1^w, f_2^w, \ldots, f_m^w\right]^T$ be a vector of objective values whose element j is the worst value of jth objective function among all nondominated solutions. Then, y^{**} is called the *ideal objective vector* and y^w is called the *nadir objective vector*. Ideal and nadir points are illustrated in Fig. 5.

The *territory* of individual y is defined as the region within a distance τ of y in each objective among the regions that neither dominate nor is dominated by y_i. Here, τ determines the territory size. Mathematically, the territory of y contains all points in V, defined by

$$V = \{y' : \ |f_j - f_j'| < \tau \text{ for } j = 1, 2, \ldots, m \ \wedge \ y \text{ and } y' \text{ do not dominate each other}\} \tag{3}$$

where f_j and f_j' are the jth objective values of y and y', respectively.

As in prTDEA, we will employ the idea of favorable weights to identify the location of an individual. The *favorable weights* $\mathbf{w_i} = \{w_{i1}, w_{i2}, \ldots, w_{im}\}$ of individual y_i are a set of weights that minimizes its weighted Tchebycheff distance from the ideal point. An important property of a Tchebycheff function is that the weighted Tchebycheff distance of any Pareto-optimal solution with its own favorable weights is shorter than that of any other Pareto-optimal solution. This is true regardless of the location of the Pareto-optimal solution under consideration. Hence, the favorable weights of a solution favors it against all other solutions in calculating its weighted Tchebycheff distance from the ideal point. The kth element in the favorable weight vector $\mathbf{w_i}$ of y_i is calculated as follows:

$$w_{ik}=\begin{cases} \frac{1}{f_{ik}-f_k^{**}}\left[\sum_{j=1}^{m}\frac{1}{f_{ij}-f_j^{**}}\right]^{-1} & \text{if } f_{ij} \neq f_j^{**} \text{ for all } j = 1, 2, \ldots, m \\ 1 & \text{if } f_{ik} = f_k^{**} \\ 0 & \text{if } f_{ik} \neq f_k^{**} \text{ but } \exists j \text{ such that } f_{ij} = f_j^{**} \end{cases} \tag{4}$$

where f_{ij} is the jth objective value of y_i and f_j^{**} is the jth element of the ideal objective vector [36, p. 425]. These weights are derived by setting the weighted distances of the solution from the ideal point in each objective equal and by normalizing the weights to sum to 1.0. The idea of favorable weights was first used by [8] within the context of MOEAs.

Let h denote the interaction stage counter. The weight space R^h of a *preferred region* R^h is defined by a set of Tchebycheff weight ranges; $R^h = \left[\mathbf{l^h}, \mathbf{u^h}\right] = \left\{\left[l_1^h, u_1^h\right], \left[l_2^h, u_2^h\right], \ldots, \left[l_m^h, u_m^h\right]\right\}$. We define R^0 as the entire weight range, i.e., $\left[l_j^0, u_j^0\right] = [0, 1] \ \forall \ j = 1, 2, \ldots, m$.

B. Overview of the Algorithm

Our approach can be flexible in the amount of involvement of the DM. At the minimum, the DM may only choose the solution he/she prefers among the several solutions provided at each interaction stage. In this case, the remaining information can be set by the algorithm. Alternatively, the DM may set the values of some parameters such as the number of interaction stages, H, and the number of solutions, P, he/she will evaluate at each interaction stage.

Interaction stages $h = 1, 2, \ldots, H$ are scheduled at the generations $G_1, G_2, \ldots, G_H$, respectively. The lengths of these generations are set such that the algorithm sufficiently converges to the Pareto frontier around the preferred solution of the most recent interaction stage. The algorithm starts by finding individuals in the entire Pareto-optimal frontier, R^0. The starting territory size is τ_0 and the final territory size is τ_H. The algorithm pauses at generations $t = G_1, G_2, \ldots, G_H$ for interaction and presents a small representative sample from the individuals found so far to the DM. At interaction stage h, a new preferred weight region R^h is defined and a new territory size parameter τ_h is assigned to it. A new preferred weight region is always smaller than the previously defined preferred weight regions. Also $\tau_k > \tau_l \ \forall k < l$. This helps the algorithm to converge to the preferred region of the DM and obtain denser solutions around the preferred solution of the most recent interaction stage. When t is equal to the maximum number of iterations T, the algorithm stops and reports the final population. We give an outline of the algorithm below.

1) Specify the parameters $\bar{N}$, T, H, τ_0, τ_H.
2) Set the iteration count $t = 0$ and and the interaction count $h = 0$. Schedule interaction stages at generations $G_1, G_2, \ldots, G_H$ (see Section III-F).
3) 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.
4) Set $t \leftarrow t+1$ and $h \leftarrow h+1$. Choose a parent from each of the populations $P(t)$ and $A(t)$. Recombine parents to create a new offspring and apply mutation.
5) Check whether the offspring satisfies the acceptance condition into $P(t)$ (see Section II). If it does, insert it into $P(t)$ and go to the next step. Otherwise, go to Step 4.
6) Check whether the offspring satisfies the acceptance condition into $A(t)$ (see Section III-C). If it does, insert it into $A(t)$.
7) If $t < G_h$, go to Step 8. Otherwise, pause for an interaction with the DM.
   a) Determine the set of individuals F_h^2 to present the DM by filtering $A(t)$.
   b) Ask the DM to choose his/her best individual y^* among F_h^2.
   c) Estimate a new preferred weight region R^h using y^*.
   d) Compute and assign τ_h to R^h.
   e) Set $h \leftarrow h + 1$.
8) If $t = T$, stop and report the archive population. Otherwise go to Step 4.

An offspring is accepted into the regular population if no individual in the regular population dominates it. The offspring replaces one of the individuals that it dominates (or a random one if it does not dominate an individual). Archive acceptance is more involved and its details are given in Section III-C.

In our experiments, we use the parent selection scheme used in TDEA. Other selection schemes may be employed if needed, as [33] indicate. If tournament selection is used, tournament size may be increased to increase the selection pressure. In addition, various strategies can be used to choose parents that compete in the tournament.

Using two populations is only a design choice. It is possible to implement the algorithm combining the two populations into a single population.

C. Archive Updating

iTDEA modifies the archive acceptance procedure of TDEA, since it involves the determination of the preferred weight region of an offspring c. As in TDEA, the process starts with a dominance check. If any individual in the archive dominates offspring c, then c is rejected and the process terminates. Otherwise, we start the second stage of the archive acceptance by removing all individuals dominated by c. Then, we determine the weight region c belongs to. For this purpose, we compute the favorable weights, $\mathbf{w}_c$, of c and determine the regions whose weight ranges contain these weights. Among these regions, we select the last created region k, as it has the the smallest τ_k, and set $\tau = \tau_k$ (see item 5 below). The rest of the process is the same as in TDEA. We first determine the closest individual, y_{i^*}, to c in terms of the scaled rectilinear distance. Then, we check whether c is in the territory of y_{i^*}. If it is, we reject c. Otherwise, we accept c.

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

1) Test c against each individual $y_i \in A(t)$ for dominance. Mark the individuals dominated by c. If c is dominated by at least one y_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 the next step.
4) Calculate the favorable weights $\mathbf{w}_c$ of c with (4) using scaled objective values.
5) Let $\mathbf{R} = \left\{R^k : k \leq h,\ w_{cj} \in \left[l_j^k, u_j^k\right]\ \forall j = 1, 2, \dots, m\right\}$. Then, $k^* = \operatorname{argmax}_k \left(R^k \in \mathbf{R}\right)$ and $\tau = \tau_{k^*}$.
6) Calculate the rectilinear distance $d_{ci} = \sum_{j=1}^{m} \left|\hat{f}_{cj} - \hat{f}_{ij}\right|$ of c to each individual $y_i \in A(t)$.
7) Find $i^* = \operatorname{argmin}_i(d_{ci})$, i.e., the individual y_{i^*} closest to c.
8) Find the maximum scaled absolute objective difference between c and y_{i^*}. That is, find
$$\delta = \max_{j=1,2,\dots,m} \left|\hat{f}_{cj} - \hat{f}_{i^*j}\right|. \tag{5}$$
9) If $\delta \geq \tau$, accept and insert c into $A(t)$. Otherwise, reject c.

The schedule of interaction stages is important for the correct functioning of the algorithm. The weight mechanism works well only if the population has somewhat converged to the Pareto-optimal frontier. Otherwise, the selected individual's weights may mislead the algorithm. Hence, the first interaction should be scheduled at a generation where the population is not too far from the Pareto-optimal frontier. Schedule of the last interaction stage is also important. Since the final preferred weight region is determined, the algorithm should be allowed to spend enough time to concentrate and converge well to that region.

D. Scaling

Each objective in an MOP may have a different range and scale. The cohesion of objectives without a proper scaling mechanism may lead to biases. To address this issue, scaling techniques have been proposed in the literature [36, see for details]. In iTDEA, we employ the scaling scheme in [8]. Their method treats the objective values between the ideal $y^{**} = \left[f_1^{**}, f_2^{**}, \dots, f_m^{**}\right]^T$ and nadir point $y^w = \left[f_1^w, f_2^w, \dots, f_m^w\right]^T$ differently from those beyond the nadir point. While they use linear scaling for the former, the latter are scaled using the following sigmoid function:

$$\Psi(\gamma) = \left(\frac{1}{1 + e^{-\frac{t}{\lambda}}} - C\right) G \tag{6}$$

where C and G are parameters to control the shape, λ is a parameter for controlling the slope and γ is the value to be scaled. We set $C = 0.5$ and $G = 2$ so that the sigmoid function takes values between 0 and 1. As done by the authors, we scale the values in the efficient range into a large portion of [0, 1]. The rest is scaled into the remaining narrow interval, since they are not as important as the nondominated range. For this purpose, we set λ in such a way that the sigmoid function has a very small slope m at the nadir point (y^w). This is done by solving the following equation using a very small slope of the sigmoid function at the nadir point:

$$\frac{d}{d\gamma}\Psi(\gamma)\bigg|_{y=f_j^w} = \frac{e^{-\frac{f_j^w}{\lambda}}}{\lambda\left(1 + e^{-\frac{f_j^w}{\lambda}}\right)^2} G = m = \text{slope at nadir point.} \tag{7}$$

We present the steps of the scaling procedure below for a minimization type objective function. Note that maximization type objective functions have to be converted into minimization before scaling.

1) Shift the objective value and nadir point until the ideal point becomes 0, i.e., set $\bar{f}_j = f_j - f_j^{**}$ and $\bar{f}_j^w = f_j^w - f_i^{**}$.
2) Using $\bar{f}_j^w$ and a very small m, solve (7) to determine λ.
3) The scaled objective $\hat{f}_j$ is then found as follows:

$$\hat{f}_j = \begin{cases} \dfrac{\Psi(\bar{f}_j^w)}{\bar{f}_j^w}\bar{f}_j & \text{if } f_j \leq f_j^w \\ \left(\dfrac{1}{1 + e^{-\frac{\bar{f}_j}{\lambda}}} - C\right) G & \text{otherwise} \end{cases}. \tag{8}$$

An illustration of scaling is given in Fig. 6. In this paper, we estimate the nadir points using payoff tables and determine λ values using a trial-and-error method. Once the objectives are scaled between 0 and 1, the territory sizes can be meaningfully selected in proportion to the scaled ranges of the objectives.

E. Preferred Weight Region Estimation

As mentioned earlier, iTDEA starts by finding individuals over the entire Pareto-optimal frontier. At interaction stages, it presents a set of individuals for DM to select his/her best among them. This individual is then used to build a *preferred region* by restricting the weight space. A preferred region contains individuals which DM prefers to other individuals outside. In the case of multiple preferred regions, DM prefers

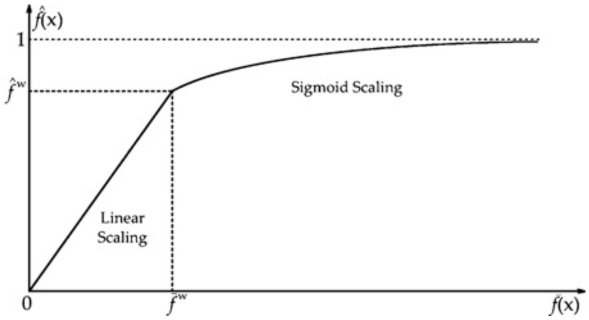

Fig. 6. Scaling.

an individual located in a preferred region to other individuals in preferred regions created earlier.

iTDEA concentrates on a preferred region by shrinking the territories of the individuals falling in them. This is done by simply using a smaller τ for such offspring in the archive evaluation stage. We keep decreasing τ value for each newly created preferred region. In this way, we increase resolution and have a better approximation of preferred regions. On the other hand, individuals located elsewhere are evaluated using a larger τ, which keeps less dense populations in the regions that are less desirable to the DM.

We use the method developed by [34] at each interaction stage h to find the preferred weight region R^h (see also [36, pp. 446–450], [37, pp. 154–161]). This method was tested with human users in an experiment against several other methods and was found to be the most favored by the users [38].

Let l and u denote the lower and upper bounds, respectively, and r be the reduction factor. The algorithm starts with $h = 0$ and R^0 having $[l_j^0, u_j^0] = [0, 1]$ for $j = 1, 2, \ldots, m$. Suppose that the DM chooses y^* as his/her best individual in interaction stage h. We compute the favorable weights $\mathbf{w}^*$ of the chosen solution y^* using (4). Then, the next preferred region $R^h = [\mathbf{l^{(h)}}, \mathbf{u^{(h)}}]$ is determined using the following formula:

$$\left[l_j^{(h)}, u_j^{(h)}\right] = \begin{cases} \left[0, r^h\right] & \text{if } w_j^* - \frac{r^h}{2} \leq 0 \\ \left[1 - r^h, 1\right] & \text{if } w_j^* + \frac{r^h}{2} \geq 1 \\ \left[w_j^* - \frac{r^h}{2}, w_j^* + \frac{r^h}{2}\right] & \text{otherwise} \end{cases} \quad (9)$$

for all $j = 1, 2, \ldots, m$, where r is a reduction factor raised to the power h to determine the amount to shrink the preferred weight region around the favorable weights of the preferred solution at interaction stage h.

F. Parameter Values

Steuer [36, pp. 446–450] lists a set of intervals as a guide to set the parameters of the algorithm. Let λ be the desired width for the weights of each objective in the final preferred weight region, P be the number of solutions to be presented to the DM in each interaction stage, and H be the total number of interaction stages. Naturally, as the amount of preference information obtained from the DM increases, we expect the algorithm to converge to the preferred region of the DM in a better way. However, requesting too much preference information is not desirable for the sake of keeping the cognitive load on the DM reasonable. The information requirement is closely dependent on the number of objectives, m, as the solution space grows with m. Therefore, Steuer suggests setting the number of interaction stages to approximately m and the number of solutions presented to the DM at each interaction stage to more than m. He suggests the width of the preferred weight region in each objective to be a proportion of m and the reduction factor for the weights in each interaction stage to be approximately determined by λ

$$P \gtrapprox m \quad (10)$$

$$H \approx m \quad (11)$$

$$\frac{1}{2m} \lessapprox \lambda \lessapprox \frac{3}{2m} \quad (12)$$

$$\sqrt[m]{\frac{1}{P}} \lessapprox r \lessapprox \sqrt[H-1]{\lambda}. \quad (13)$$

The DM may also be consulted in choosing the values of some of these parameters. See [34] for a detailed discussion on setting the parameters. Since DM can evaluate a limited number of individuals effectively, it is important to choose a reasonable value for P. We use $P = 2m$ for all interaction stages except the first one. In the first interaction stage and upon termination we present $4m$ individuals to the DM. As the number of objectives increases, the algorithm needs more interactions to converge, since Pareto-optimal frontier's size increases. We try four and six interactions in all our tests. We set the final width as $\frac{1}{m}$ in all problems.

As seen in (9), regions shrink faster in earlier interaction stages. Toward the end, the amount of shrinkage gets smaller. We employ a similar method for setting τ_h to control the number of individuals in the population after defining R^h . We choose an exponential decrease for τ. If we set the initial territory size as τ_0, and the final territory size as τ_H, then the rate of change for each interaction stage, ρ, and the territory size in each interaction stage, τ_h, can be calculated. In the earlier interaction stages, τ decreases faster and the decrease gets slower with each interaction stage

$$\tau_h = \tau_H e^{(H-h)\rho} \quad (14)$$

where

$$\rho = \frac{\ln \frac{\tau_0}{\tau_H}}{H}. \quad (15)$$

We schedule the first interaction stage with the DM at one-third of the maximum number of iterations T, i.e., $G_1 = \frac{T}{3}$. The algorithm is allowed to run $\frac{T}{6}$ generations after the final interaction stage. The number of iterations between each subsequent interaction stage is $\frac{T}{2(H-1)}$.

G. Filtering

After each interaction stage, the algorithm runs sufficiently long to get the population denser around the most-recently preferred solutions utilizing the new territory sizes. Once the new population with the desired properties is obtained, it is time to get additional preference information from the DM to further guide the search.

In order to keep the information load of the DM reasonable, we choose a small subset of individuals from the population to present to the DM. Here, the chosen individuals should be a good representation of the last preferred region. Otherwise, the individual selected by the DM may mislead the algorithm and cause an inaccurate estimate for the next preferred region. For choosing individuals to present the DM, we use a filtering procedure that utilizes modified dominance scheme similar to ϵ-dominance. In this scheme, an individual is said to be nondominated if there is no individual that is at least an ϵ amount better than that individual in every objective and more than an ϵ amount better at least in one objective. The steps of the filtering procedure at interaction stage $h = 2, 3, \ldots, H-1$ are as follows.

1) Form the first filtered list F_h^1 by taking all individuals corresponding to weight region R^{h-1} only. That is, $F_h^1 = \left\{y_i \in A(G_h) :\ \mathbf{w}_i \in \left[\mathbf{l}^{(h-1)}, \mathbf{u}^{(h-1)}\right]\right\}$.
2) Test each pair of individuals $y_k, y_l \in F_h^1$ for modified dominance using $\epsilon = \tau_{h-1}$. If y_k dominates y_l and is not dominated by y_l, remove y_l from F_h^1, and vice versa.
3) Calculate rectilinear distances d_{kl} between each pair of individuals $k, l \in F_h^1$.
4) Initialize the second filtered list F_h^2 by moving a pair of individuals $(y_k, y_l) = \operatorname{argmax}_{(y_u, y_v) \in F_h^1}(d_{uv})$ from F_h^1 to F_h^2. That is, choose the pair of individuals that are farthest to each other in rectilinear distance and move them from F_h^1 to F_h^2.
5) Fill F_h^2 until its size is equal to P, each time moving solution $y_k = \operatorname{argmax}_{y_u \in F_h^1}\left(\min_{y_v \in F_h^2}(d_{uv})\right)$ to F_h^2. That is, move the individual y_k in F_h^1 which is at maximum distance to its closest individual in F_h^2.
6) Present solutions in F_h^2 to the DM.

IV. Computational Experiments

We conduct our simulation runs on test problems ZDT4 [14], DTLZ1, and DTLZ2 [39]. We use the simulated binary crossover operator [40] with probability of crossover, $p_c = 1.0$ and crossover parameter $\eta_c = 20$. The mutation operator is the polynomial mutation operator [41] with probability of mutation, $p_{\text{mut}} = \frac{1}{\#\text{ of variables}}$ and mutation parameter $\eta_{\text{mut}} = 20$. The implementation is made in C++ and the program is built with GNU C/C++ Compiler 4.2.3. All computational tests are made on a Pentium IV 2.8 GHz, 1 GB RAM computer running Kubuntu Linux 8.04.

We first simulate the DM's preferences using a Tchebycheff utility function of the following form:

$$U = \operatorname{Min}_{y_i} \operatorname{Max}_{j=1,2,\ldots,m}\left[w_j \left| f_{ij} - f_j^{**}\right|\right] \tag{16}$$

where f_{ij} is the jth objective value of y_i, f_j^{**} is the jth element of the ideal vector, and w_j is the jth element of the weight vector $\mathbf{w}$ of the utility function. We also simulate the DM's preferences using linear and quadratic underlying utility functions later.

TABLE I

Test Parameters

	ZDT4	DTLZ1	DTLZ2
Test 1 weights, $\mathbf{w}$	(0.5, 0.5)	(0.33, 0.33, 0.33)	(0.33, 0.33, 0.33)
Test 2 weights, $\mathbf{w}$	(0.2, 0.8)	(0.2, 0.3, 0.5)	(0.2, 0.3, 0.5)
Test 3 weights, $\mathbf{w}$	(0.65, 0.35)	(0.7, 0.2, 0.1)	(0.7, 0.2, 0.1)
Ideal vector, $\mathbf{f}^*$	(0, 0)	(0, 0, 0)	(0, 0, 0)
No. of interactions H	4, 6	4, 6	4, 6
Population size $\bar{N}$	200	400	400
τ_0	0.1	0.1	0.1
τ_H	0.00001	0.005	0.005
No. of iterations T	80 000	320 000	320 000
No. of replications	50	50	50

For each problem, we choose three different weight sets for the utility function. For each setting, we interact 4 or 6 times with the DM to observe how the number of interactions affects the final population. Each run is replicated 50 times. Our aim is to see whether the algorithm is robust for different types of preference structures and problem types. The parameter compositions of the simulation runs are given in Table I.

To observe the effect of filtering on the results, we run each instance with and without filtering. In the unfiltered mode, we assume that the DM chooses his/her best solution among the entire archive population. In the end, we report the individual (denoted as "No Filter") with the best utility. In the filtering case, we filter P individuals ($2P$ in the first interaction stage) to present to the DM at each interaction stage. At the end of the run, we report two individuals. The first one (denoted as "Filter 1") is the one with the best utility in the entire archive population. The second individual (denoted as "Filter 2") is the one with the best utility among $2P$ filtered individuals from the final preferred region.

We calculate the absolute deviation (Δ) and percentage deviation ($\bar{\Delta}$) as follows:

$$\Delta = U^* - U \tag{17}$$

$$\bar{\Delta} = \frac{U^* - U}{U^* - U^w} \tag{18}$$

where U is the utility value of the individual reported by the algorithm, U^* is the true optimal solution's utility value, and U^w is the worst utility value among the true entire Pareto-optimal frontier.

We conduct statistical tests of significance between the No Filter and Filter 1 cases as well as between Filter 1 and Filter 2 cases. In reporting the results, we mark those cases where the differences are not significant. All the unmarked cases are statistically significant. We also present the corresponding confidence intervals later in order to demonstrate the magnitudes of the differences.

A. 2-Objective Tests

1) *Test 1:* In the first test, the best solution is at the center of the Pareto-optimal frontier. Table II shows the results found for all cases. We observe that more interactions with the DM

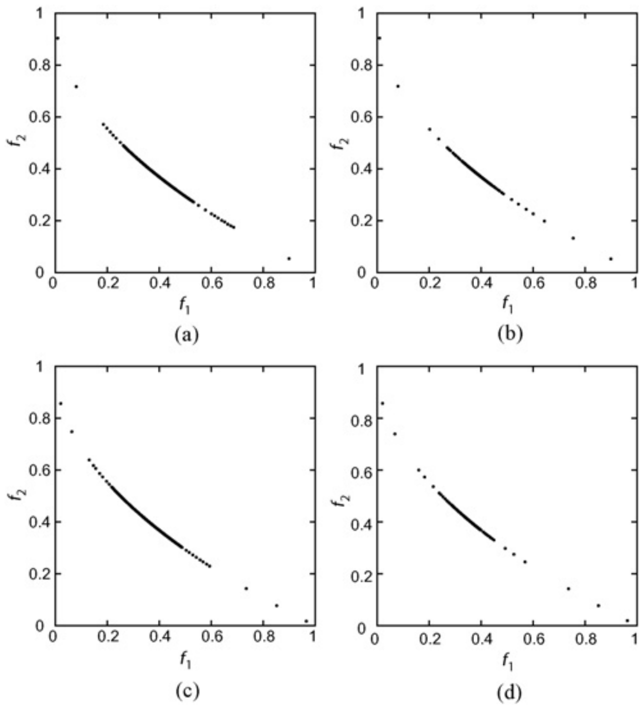

Fig. 7. ZDT4 Test 1 plots. (a) Unfiltered with four interactions. (b) Unfiltered with six interactions. (c) Filtered with four interactions. (d) Filtered with six interactions.

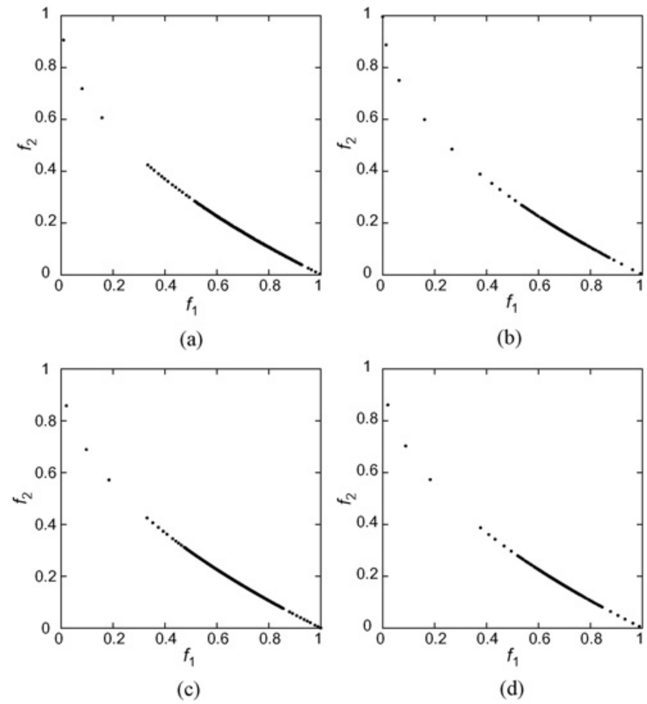

Fig. 8. ZDT4 Test 2 plots. (a) Unfiltered with four interactions. (b) Unfiltered with six interactions. (c) Filtered with four interactions. (d) Filtered with six interactions.

allows better convergence to the desired location. In addition, it can be seen that iTDEA successfully converges in both filtered and unfiltered cases. The deviation from the optimal solution is very small in all cases. It slightly increases when filtering is applied to the final population. Fig. 7 displays the plots of sample runs. The final regions and the deviations from the best solution are smaller for the 6-interaction cases.

2) *Test 2:* In the second test, the utility function favors the second objective more than the first objective. We test whether such a bias affects the convergence. As seen in Table III, the results are very similar to those of Test 1. We obtain better results with six interactions than with four interactions. No Filter and Filter 1 cases perform very close to each other. We present the plots of sample runs in Fig. 8.

3) *Test 3:* In Test 3, the first objective is favored with a large relative weight. The results show some slight variations but again the deviations from the best solution are very small in all cases (see Table IV). Fig. 9 displays the sample plots of the final archive population of well-guided runs.

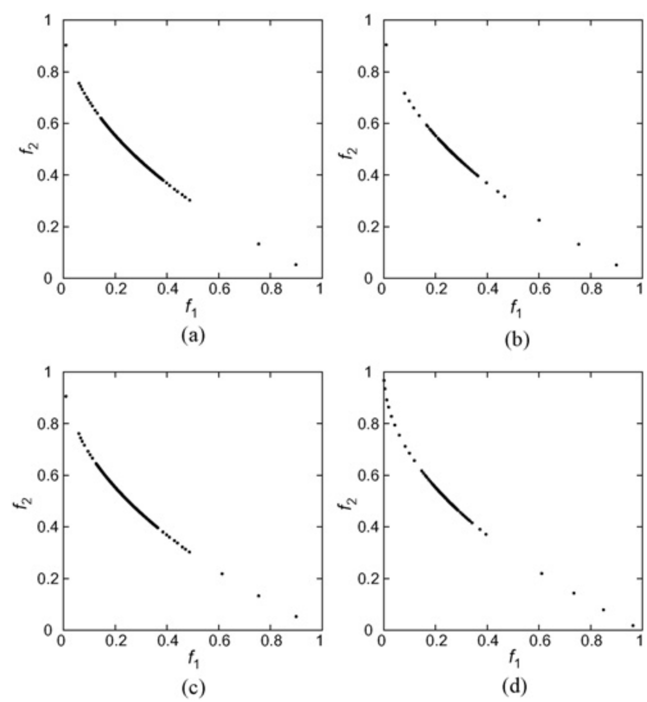

Fig. 9. ZDT4 test three plots. (a) Unfiltered with four interactions. (b) Unfiltered with six interactions. (c) Filtered with four interactions. (d) Filtered with six interactions.

B. 3-Objective Tests

We use problems DTLZ1 and DTLZ2 for the 3-objective tests. The Pareto-optimal frontiers of these two problems have different shapes and we test whether the algorithm handles both cases well.

1) *Test 1:* In both DTLZ1 and DTLZ2 tests (Tables V and VI), we observe that the relative differences between filtered and unfiltered cases are a little larger than those of ZDT4 tests. The standard deviations of filtered cases are also higher. The algorithm finds good individuals in the filtered case; however, the final filtering may cause slightly worse individuals to be reported. The algorithm still obtains good solutions that are very close to the best solutions in the absolute sense. In the plots (Figs. 10 and 11) we observe the progress of the preferred regions. In the unfiltered cases, the algorithm finely advances to the desired final preferred region. In the filtered cases, the individual chosen by the DM slightly

TABLE II

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 1 OF ZDT4

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.19115[1]	0.000132	0.00017	0.0540%			
Filter 1	0.19114	0.000100	0.00016	0.0508%	4	0.19098	0.500
Filter 2	0.19281	0.001290	0.00183	0.5912%			
No Filter	0.19111[1]	0.000099	0.00013	0.0411%			
Filter 1	0.19110	0.000080	0.00012	0.0379%	6	0.19098	0.500
Filter 2	0.19143	0.000239	0.00045	0.1446%			

[1]No statistical difference at 95% level compared to Filter 1 case.

TABLE III

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 2 OF ZDT4

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.13739[1]	0.000092	0.00013	0.0199%			
Filter 1	0.13740	0.000100	0.00014	0.0214%	4	0.13726	0.800
Filter 2	0.13894	0.001100	0.00168	0.2537%			
No Filter	0.13734	0.000057	0.00008	0.0123%	6	0.13726	0.800
Filter 1	0.13736	0.000070	0.00010	0.0153%			
Filter 2	0.13756	0.000197	0.00030	0.0455%			

[1]No statistical difference at 95% level compared to Filter 1 case.

TABLE IV

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 3 OF ZDT4

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.17081	0.000095	0.00015	0.0313%			
Filter 1	0.17083	0.000121	0.00017	0.0355%	4	0.17066	0.650
Filter 2	0.17215	0.001050	0.00149	0.3108%			
No Filter	0.17076[1]	0.000063	0.00010	0.0209%			
Filter 1	0.17075	0.000050	0.00009	0.0188%	6	0.17066	0.650
Filter 2	0.17105	0.000199	0.00039	0.0814%			

[1]No statistical difference at 95% level compared to Filter 1 case.

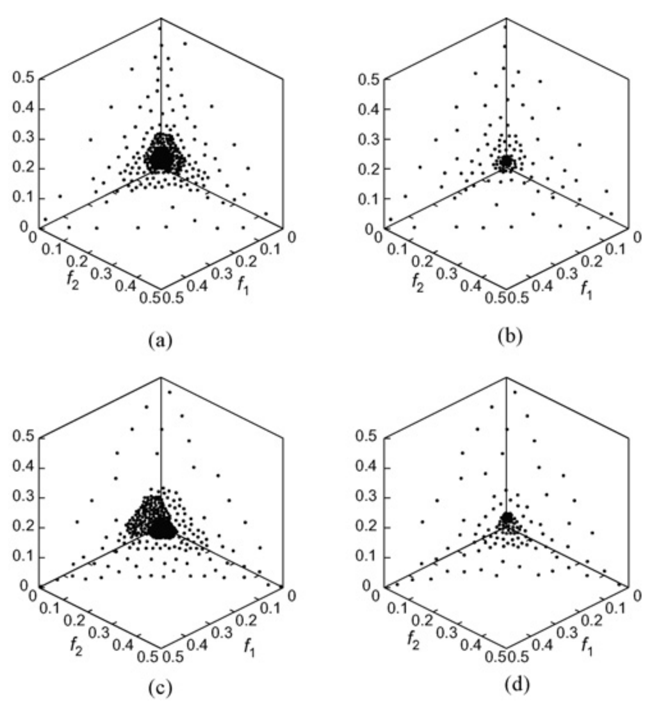

Fig. 10. DTLZ1 Test 1 plots. (a) Unfiltered with four interactions. (b) Unfiltered with six interactions. (c) Filtered with four interactions. (d) Filtered with six interactions.

misleads the algorithm in the first interaction stages. When the number of interaction stages is increased from four to six, the algorithm performs better for the Filter 2 case as well.

2) *Test 2:* In this test, the second objective is favored more than the other two objectives. Incorporating such a bias to the utility function does not affect the results and they are very similar to those found in Test 1 (Tables VII and VIII). One unusual observation is that Filter 2 results with four interactions are better than those with six interactions in problem DTLZ2 (Table VIII). This indicates that the algorithm converges to a slightly different region after the last interaction in some runs. Since the final region is larger with four interactions, the effect is smaller in the 6-interaction case. The patterns we observe in the plots given in Figs. 12 and 13 are also similar to those of Test 1.

3) *Test 3:* In Test 3, the third objective is favored more than the others. As can be seen from Tables IX and X all versions of iTDEA perform well. We observe slightly better performance of the unfiltered case, followed by the "Filter 1" case, as before. The patterns in the plots (Figs. 14 and 15) are also similar to those of the previous 3-objective tests.

We constructed 95% confidence intervals for the equality of means between No Filter and Filter 1 cases as well as Filter 1 and Filter 2 cases. We take the paired differences and employ a t-distribution for the average of 50 differences. The test is justified due to the large enough sample size. The resulting

TABLE V

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 1 OF DTLZ1

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.05539	0.000161	0.00039	0.3573%			
Filter 1	0.05591	0.001221	0.00091	0.8291%	4	0.05500	0.165
Filter 2	0.05793	0.002646	0.00293	2.6618%			
No Filter	0.05536	0.000149	0.00036	0.3273%			
Filter 1	0.05593	0.001380	0.00093	0.8455%	6	0.05500	0.165
Filter 2	0.05619	0.001669	0.00119	1.0791%			

TABLE VI

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 1 OF DTLZ2

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.04870	0.000154	0.00031	0.1537%			
Filter 1	0.04960	0.001203	0.00122	0.6031%	4	0.04839	0.250
Filter 2	0.05257	0.003496	0.00418	2.0737%			
No Filter	0.04872	0.000125	0.00033	0.1641%			
Filter 1	0.04961	0.001220	0.00122	0.6046%	6	0.04839	0.250
Filter 2	0.05091	0.003194	0.00252	1.2499%			

TABLE VII

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 2 OF DTLZ1

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.03062	0.000080	0.00019	0.0592%			
Filter 1	0.03112	0.000780	0.00068	0.2135%	4	0.03043	0.350
Filter 2	0.03337	0.002389	0.00293	0.9169%			
No Filter	0.03080	0.000324	0.00037	0.1155%			
Filter 1	0.03146	0.000953	0.00102	0.3199%	6	0.03043	0.350
Filter 2	0.03307	0.002540	0.00264	0.8256%			

TABLE VIII

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 2 OF DTLZ2

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.19136[1]	0.000376	0.00083	0.5983%			
Filter 1	0.19154	0.000961	0.00101	0.7273%	4	0.19053	0.330
Filter 2	0.19757	0.006420	0.00704	5.0507%			
No Filter	0.19137	0.000346	0.00084	0.6054%			
Filter 1	0.19226	0.001880	0.00173	1.2435%	6	0.19053	0.330
Filter 2	0.19391	0.003770	0.00338	2.4265%			

[1]No statistical difference at 95% level compared to Filter 1 case.

TABLE IX

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 3 OF DTLZ1

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.15839	0.000190	0.00050	0.1448%			
Filter 1	0.15998	0.001790	0.00209	0.6095%	4	0.15789	0.500
Filter 2	0.16568	0.004380	0.00779	2.2757%			
No Filter	0.15835	0.000200	0.00046	0.1331%			
Filter 1	0.16078	0.002400	0.00289	0.8434%	6	0.15789	0.500
Filter 2	0.16456	0.004830	0.00667	1.9483%			

TABLE X

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 3 OF DTLZ2

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.08895	0.000148	0.00022	0.0366%			
Filter 1	0.09005	0.001271	0.00132	0.2165%	4	0.08872	0.700
Filter 2	0.09504	0.003928	0.00632	1.0335%			
No Filter	0.08901	0.000197	0.00029	0.0479%			
Filter 1	0.08969	0.000911	0.00097	0.1588%	6	0.08872	0.700
Filter 2	0.09267	0.004392	0.00395	0.6461%			

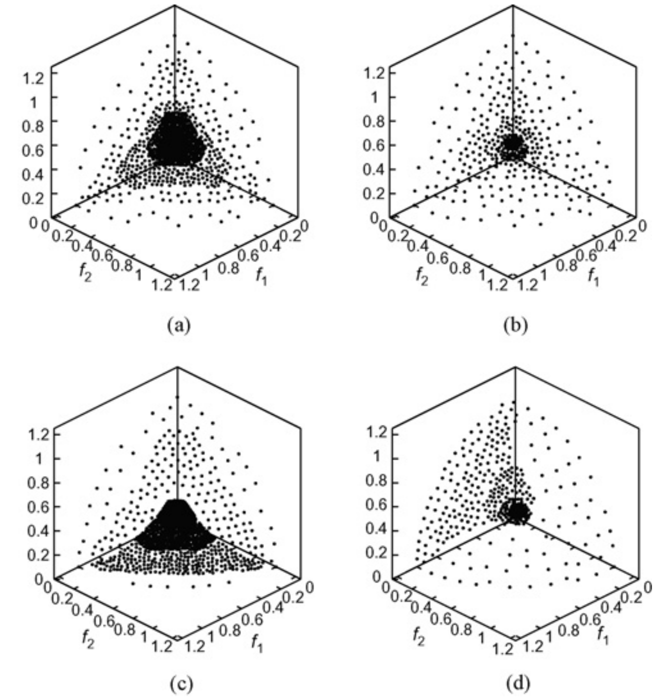

Fig. 11. DTLZ2 Test 1 plots. (a) Unfiltered with four interactions. (b) Unfiltered with six interactions. (c) Filtered with four interactions. (d) Filtered with six interactions.

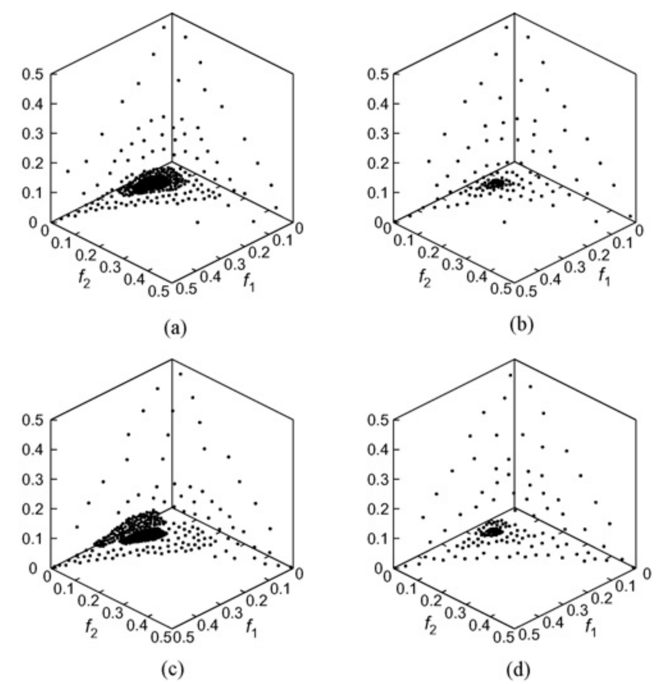

Fig. 12. DTLZ1 Test 2 plots. (a) Unfiltered with four interactions. (b) Unfiltered with six interactions. (c) Filtered with four interactions. (d) Filtered with six interactions.

confidence intervals are presented in Table XI. For almost all cases, there is statistically significant difference between means. This indicates that, as we get more information from the DM, the quality of the resulting solution improves, as expected. However, the magnitudes of the differences are rather small implying that the results obtained with less information are close to the best solutions as well.

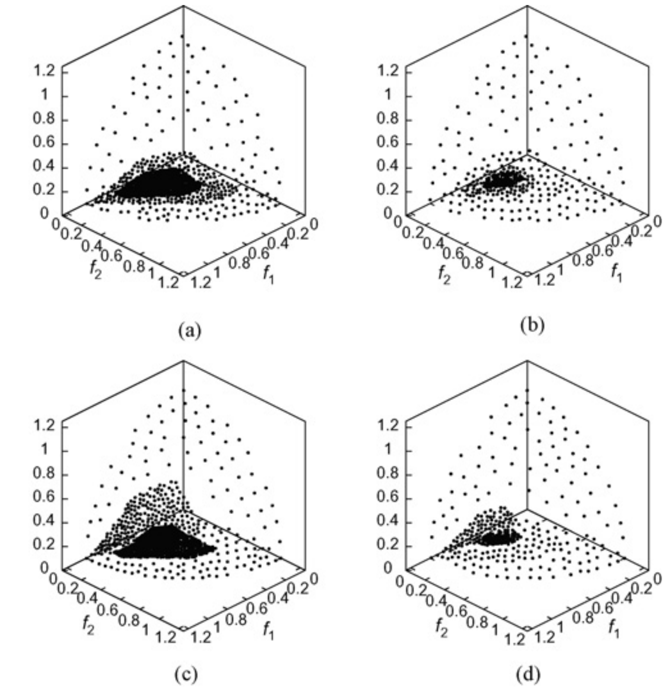

Fig. 13. DTLZ2 Test 2 plots. (a) Unfiltered with four interactions. (b) Unfiltered with six interactions. (c) Filtered with four interactions. (d) Filtered with six interactions.

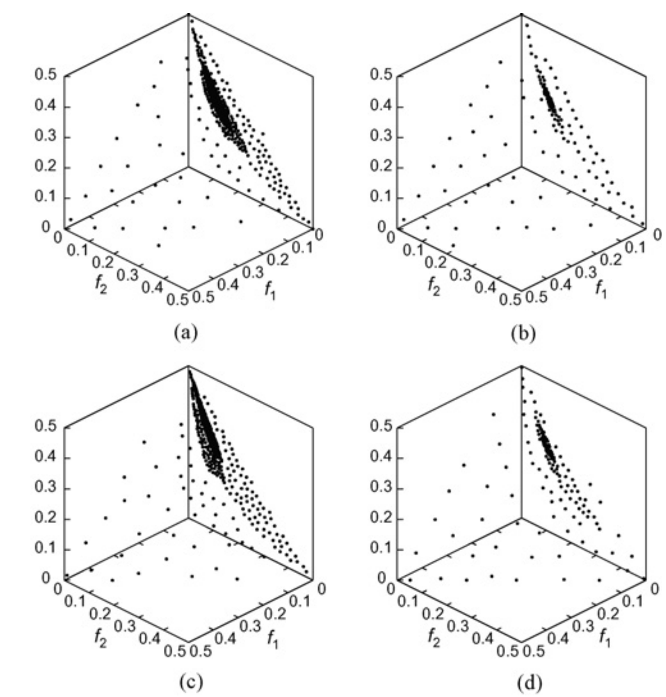

Fig. 14. DTLZ1 Test 3 plots. (a) Unfiltered with four interactions. (b) Unfiltered with six interactions. (c) Filtered with four interactions. (d) Filtered with six interactions.

C. Response Errors

In previous tests, we assumed that the DM's responses are always accurate. In order to see the effects of response errors, we incorporate a Gaussian noise in the utility calculations. Let U be the DM's true utility for a solution. Assume that the DM perceives a distorted utility U' that is calculated by

TABLE XI
95% CONFIDENCE INTERVALS FOR TCHEBYCHEFF UTILITY FUNCTION (IN 10^{-5})

		Four Interactions		Six Interactions	
Problem	Test	No Filter vs. Filter 1	Filter 1 vs. Filter 2	No Filter vs. Filter 1	Filter 1 vs. Filter 2
ZDT1	1	(−2.3, 4.3)	(−203.5, −130.8)	(−1.5, 4.6)	(−39.8, −27.0)
	2	(−3.5, 0.9)	(−185.8, −123.3)	(−3.7, −0.1)	(−25.3, −14.2)
	3	(−4.4, −0.2)	(−162.4, −101.9)	(−1.2, 2.6)	(−35.9, −24.1)
DTLZ1	1	(−86.7, −17.2)	(−254.1, −148.9)	(−95.9, −18.1)	(−39.7, −11.7)
	2	(−126.0, −55.2)	(−374.6, −218.5)	(−124.8, −52.8)	(−194.4, −65.8)
	3	(−71.3, −27.2)	(−283.8, −165.8)	(−93.8, −36.8)	(−217.1, −106.1)
DTLZ2	1	(−45.7, 10.4)	(−779.6, −425.6)	(−144.3, −32.5)	(−241.6, −89.2)
	2	(−210.2, −108.4)	(−662.5, −476.2)	(−310.9, −175.7)	(−475.3, −279.5)
	3	(−147.6, −72.4)	(−591.4, −407.3)	(−93.9, −41.7)	(−403.7, −192.2)

TABLE XII
DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 1 OF ZDT4, WITH NOISE

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No noise	0.19281	0.001290	0.00183	0.5912%			
5% noise	0.19887	0.005060	0.00789	2.5523%	4	0.19098	0.500
10% noise	0.19895	0.004910	0.00797	2.5782%			
20% noise	0.19913	0.005240	0.00815	2.6364%			
No noise	0.19143	0.000239	0.00045	0.1446%			
5% noise	0.19220	0.000942	0.00122	0.3938%	6	0.19098	0.500
10% noise	0.19237	0.001670	0.00139	0.4488%			
20% noise	0.19382	0.003550	0.00284	0.9181%			

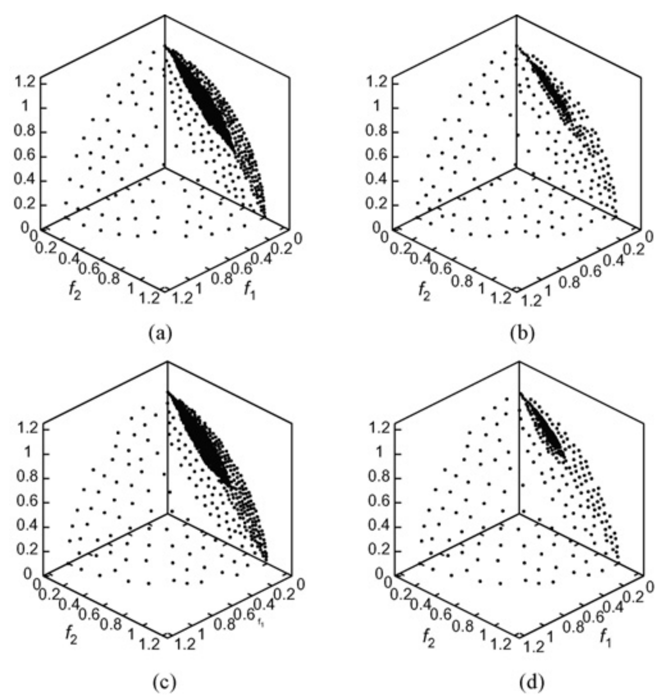

Fig. 15. DTLZ2 Test 3 plots. (a) Filtered with four interactions. (b) Unfiltered with six interactions. (c) Filtered with four interactions. (d) Filtered with six interactions.

$$U' = U \times [1 + \epsilon] \tag{19}$$

where $\epsilon \sim N\left(0, \sigma^2\right)$ and σ is a proportion of U; $\sigma = \alpha U$.

Employing the Filter 2 case, we run each test with α = 5%, 10%, and 20% and compare results with the corresponding results without any response errors. Tables XII–XIV show the results of ZDT4 tests. Deviation from the optimal values increase slightly when noise is introduced. The results seem to be robust to the amount of noise within the considered ranges. That is, the deviations seem to be slight for all noise levels. Furthermore, the algorithm can offset the effects of noise when there are more interactions with DM. Tables XV–XVII show similar results for DTLZ1 tests. Interestingly, deviation with noise is slightly less than that without noise in Test 1 with six interactions. The effects of noise for the DTLZ2 problem are shown in Tables XVIII–XX. These results are also in agreement with the previous results that additional interactions somewhat compensate for the inaccuracies caused by noise.

D. Other Utility Functions

In order to inspect how the algorithm performs with different utility functions, we repeat the same tests using linear and quadratic functions as the underlying utility function of the DM. Their formal definitions are given in (20) and (21), respectively

$$U = \mathrm{Min}_{y_i} \sum_{j=1,2,\ldots,m} w_j \left| f_{ij} - f_j^{**} \right| \tag{20}$$

$$U = \mathrm{Min}_{y_i} \sqrt{\sum_{j=1,2,\ldots,m} \left[w_j \left(f_{ij} - f_j^{**} \right) \right]^2} \tag{21}$$

where f_{ij}, y_i, f_j^{**}, and w_j are as defined before.

Since DTLZ1 has a linear Pareto-optimal frontier, all nondominated solutions are optimal for linear utility functions having equal weights (Test 1) in all objectives. Similarly, due to DTLZ2's spherical Pareto-optimal frontier, all nondominated solutions in DTLZ2 are optimal for quadratic utility functions that have equal weights (Test 1) in all objectives.

TABLE XIII

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 2 OF ZDT4, WITH NOISE

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No noise	0.13894	0.001100	0.00168	0.2537%			
5% noise	0.14723	0.005110	0.00997	1.5046%	4	0.13726	0.800
10% noise	0.14668	0.005250	0.00942	1.4216%			
20% noise	0.14692	0.005250	0.00966	1.4578%			
No noise	0.13756	0.000197	0.00030	0.0455%			
5% noise	0.13880	0.001070	0.00154	0.2326%	6	0.13726	0.800
10% noise	0.13911	0.001400	0.00185	0.2794%			
20% noise	0.13969	0.001790	0.00243	0.3669%			

TABLE XIV

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 3 OF ZDT4, WITH NOISE

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No noise	0.17215	0.001050	0.00149	0.3108%			
5% noise	0.17628	0.003850	0.00562	1.1724%	4	0.17066	0.650
10% noise	0.17633	0.004340	0.00567	1.1829%			
20% noise	0.17735	0.005310	0.00669	1.3957%			
No noise	0.17105	0.000199	0.00039	0.0814%			
5% noise	0.17206	0.000907	0.00140	0.2921%	6	0.17066	0.650
10% noise	0.17240	0.001460	0.00174	0.3630%			
20% noise	0.17340	0.002690	0.00274	0.5716%			

TABLE XV

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 1 OF DTLZ1, WITH NOISE

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No noise	0.05793	0.002646	0.00293	2.6618%			
5% noise	0.05800	0.001527	0.00300	2.7264%	4	0.05500	0.165
10% noise	0.05795	0.001540	0.00295	2.6836%			
20% noise	0.05831	0.002573	0.00331	3.0100%			
No noise	0.05619	0.001669	0.00119	1.0791%			
5% noise	0.05593	0.000520	0.00093	0.8473%	6	0.05500	0.165
10% noise	0.05593	0.000528	0.00093	0.8491%			
20% noise	0.05596	0.000533	0.00096	0.8745%			

TABLE XVI

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 2 OF DTLZ1, WITH NOISE

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No noise	0.05257	0.003496	0.00418	2.0737%			
5% noise	0.05767	0.011350	0.00928	4.6043%	4	0.04839	0.250
10% noise	0.05836	0.012420	0.00997	4.9466%			
20% noise	0.05772	0.011030	0.00933	4.6291%			
No noise	0.05091	0.003194	0.00252	1.2499%			
5% noise	0.05515	0.010430	0.00676	3.3544%	6	0.04839	0.250
10% noise	0.05508	0.010400	0.00669	3.3197%			
20% noise	0.05524	0.010470	0.00685	3.3990%			

TABLE XVII

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 3 OF DTLZ1, WITH NOISE

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No noise	0.03337	0.002389	0.00293	0.9169%			
5% noise	0.03812	0.005910	0.00769	2.4052%	4	0.03043	0.350
10% noise	0.03802	0.005819	0.00759	2.3736%			
20% noise	0.03802	0.005487	0.00758	2.3720%			
No noise	0.03307	0.002540	0.00264	0.8256%			
5% noise	0.03517	0.004178	0.00474	1.4830%	6	0.03043	0.350
10% noise	0.03517	0.004176	0.00473	1.4808%			
20% noise	0.03494	0.003716	0.00450	1.4085%			

TABLE XVIII

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 1 OF DTLZ2, WITH NOISE

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No noise	0.19757	0.006420	0.00704	5.0507%			
5% noise	0.20218	0.005140	0.01165	8.3559%	4	0.19053	0.330
10% noise	0.20188	0.005180	0.01135	8.1409%			
20% noise	0.20636	0.011880	0.01583	11.3529%			
No noise	0.19391	0.003770	0.00338	2.4265%			
5% noise	0.19446	0.002500	0.00393	2.8209%	6	0.19053	0.330
10% noise	0.19476	0.003000	0.00423	3.0360%			
20% noise	0.20283	0.012760	0.01230	8.8220%			

TABLE XIX

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 2 OF DTLZ2, WITH NOISE

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No noise	0.16568	0.004380	0.00779	2.2757%			
5% noise	0.18517	0.011970	0.02728	7.9728%	4	0.15789	0.500
10% noise	0.18336	0.012720	0.02547	7.4437%			
20% noise	0.18445	0.012810	0.02656	7.7623%			
No noise	0.16456	0.004830	0.00667	1.9483%			
5% noise	0.17498	0.012910	0.01709	4.9942%	6	0.15789	0.500
10% noise	0.17671	0.013180	0.01882	5.4998%			
20% noise	0.17656	0.014340	0.01867	5.4560%			

TABLE XX

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 3 OF DTLZ2, WITH NOISE

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No noise	0.09504	0.003928	0.00632	1.0335%			
5% noise	0.09840	0.003717	0.00967	1.5827%	4	0.08872	0.700
10% noise	0.09874	0.003159	0.01002	1.6385%			
20% noise	0.09871	0.003039	0.00999	1.6345%			
No noise	0.09267	0.004392	0.00395	0.6461%			
5% noise	0.09628	0.003974	0.00756	1.2365%	6	0.08872	0.700
10% noise	0.09678	0.003690	0.00806	1.3190%			
20% noise	0.09698	0.003494	0.00826	1.3515%			

TABLE XXI

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 1 OF ZDT4 WITH A LINEAR UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.37520[1]	0.000126	0.00020	0.1600%			
Filter 1	0.37523	0.000189	0.00023	0.1840%	4	0.37500	0.500
Filter 2	0.37533	0.000227	0.00033	0.2640%			
No Filter	0.37512[1]	0.000096	0.00012	0.0960%			
Filter 1	0.37514	0.000075	0.00014	0.1120%	6	0.37500	0.500
Filter 2	0.37517	0.000089	0.00017	0.1360%			

[1]No statistical difference at 95% level compared to Filter 1 case.

We omit those tests since they would not yield any useful information.

In Tables XXI–XXIII, we present the results of 2-objective tests with linear utility function. We observe that iTDEA successfully converges to the final preferred region of the DM. The results show that relative deviations from optimal solutions are less than 0.26% for all versions of the algorithm, and much smaller in most cases.

In Tables XXIV–XXVIII, the results of 3-objective tests with linear utility functions are shown. Despite the additional difficulty brought by the third objective, the algorithm finds solutions very close to the corresponding best solutions. The overall performance of iTDEA is slightly better when a linear utility function is used instead of a Tchebycheff utility function.

Tables XXIX–XXXI show the performance of iTDEA on 2-objective tests with quadratic utility functions. They are followed by the results of 3-objective tests in Tables XXXII–XXXVI. We obtain results similar to those of the linear utility function case in these tests. The algorithm performs very well for the quadratic utility function case too.

We construct 95% confidence intervals for the equality of means of No Filter vs. Filter 1, and Filter 1 vs. Filter 2 as before, for linear and quadratic utility functions. The results are presented in Tables XXXVII and XXXVIII. Compared to Tchebycheff utility function, there are less

TABLE XXII

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 2 OF ZDT4 WITH A LINEAR UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.20042[1]	0.000364	0.00042	0.0700%	4	0.20000	0.800
Filter 1	0.20042[2]	0.000365	0.00042	0.0700%			
Filter 2	0.20043	0.000379	0.00043	0.0717%			
No Filter	0.20025[1]	0.000196	0.00025	0.0417%	6	0.20000	0.800
Filter 1	0.20026[2]	0.000202	0.00026	0.0433%			
Filter 2	0.20026	0.000203	0.00026	0.0433%			

[1]No statistical difference at 95% level compared to Filter 1 case.
[2]No statistical difference at 95% level compared to Filter 2 case.

TABLE XXIII

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 3 OF ZDT4 WITH A LINEAR UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.30309[1]	0.000152	0.00009	0.0259%			
Filter 1	0.30306	0.000111	0.00006	0.0173%	4	0.30300	0.650
Filter 2	0.30320	0.000162	0.00020	0.0576%			
No Filter	0.30303[1]	0.000111	0.00003	0.0087%			
Filter 1	0.30302	0.000103	0.00002	0.0058%	6	0.30300	0.650
Filter 2	0.30307	0.000147	0.00007	0.0202%			

[1]No statistical difference at 95% level compared to Filter 1 case.

TABLE XXIV

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 2 OF DTLZ1 WITH A LINEAR UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.10031[1]	0.000350	0.00031	0.2067%			
Filter 1	0.10043	0.000397	0.00043	0.2867%	4	0.10000	0.250
Filter 2	0.10154	0.003110	0.00154	1.0267%			
No Filter	0.10037[1]	0.000334	0.00037	0.2467%			
Filter 1	0.10042	0.000355	0.00042	0.2800%	6	0.10000	0.250
Filter 2	0.10113	0.001520	0.00113	0.7533%			

[1]No statistical difference at 95% level compared to Filter 1 case.

TABLE XXV

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 3 OF DTLZ1 WITH A LINEAR UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.05016[1]	0.000231	0.00016	0.0537%			
Filter 1	0.05014[2]	0.000209	0.00014	0.0460%	4	0.05000	0.350
Filter 2	0.05052	0.002293	0.00052	0.1740%			
No Filter	0.05017[1]	0.000233	0.00017	0.0557%			
Filter 1	0.05014[2]	0.000191	0.00014	0.0457%	6	0.05000	0.350
Filter 2	0.05111	0.003978	0.00111	0.3693%			

[1] No statistical difference at 95% level compared to Filter 1 case.
[2] No statistical difference at 95% level compared to Filter 2 case.

TABLE XXVI

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 1 OF DTLZ2 WITH A LINEAR UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.33002[1]	0.000027	0.00002	0.0083%			
Filter 1	0.33003	0.000109	0.00003	0.0124%	4	0.33000	0.572
Filter 2	0.33112	0.003100	0.00112	0.4628%			
No Filter	0.33005[1]	0.000133	0.00005	0.0207%			
Filter 1	0.33004	0.000127	0.00004	0.0165%	6	0.33000	0.572
Filter 2	0.33255	0.005770	0.00255	1.0537%			

[1]No statistical difference at 95% level compared to Filter 1 case.

TABLE XXVII

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 2 OF DTLZ2 WITH A LINEAR UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.20100	0.001540	0.00100	0.2398%			
Filter 1	0.20283	0.004480	0.00283	0.6787%	4	0.20000	0.617
Filter 2	0.21504	0.030630	0.01504	3.6067%			
No Filter	0.20175	0.002950	0.00175	0.4197%			
Filter 1	0.20327	0.005300	0.00327	0.7842%	6	0.20000	0.617
Filter 2	0.21611	0.030270	0.01611	3.8633%			

TABLE XXVIII

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 3 OF DTLZ2 WITH A LINEAR UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.10038[1]	0.000968	0.00038	0.0598%			
Filter 1	0.10070[2]	0.002060	0.00070	0.1102%	4	0.10000	0.735
Filter 2	0.10269	0.014120	0.00269	0.4236%			
No Filter	0.10046[1]	0.000975	0.00046	0.0724%			
Filter 1	0.10040[2]	0.000837	0.00040	0.0630%	6	0.10000	0.735
Filter 2	0.10293	0.015870	0.00293	0.4614%			

[1]No statistical difference at 95% level compared to Filter 1 case.
[2]No statistical difference at 95% level compared to Filter 2 case.

TABLE XXIX

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 1 OF ZDT4 WITH A QUADRATIC UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.26907[1]	0.000101	0.00015	0.0649%	4	0.26892	0.500
Filter 1	0.26908	0.000126	0.00016	0.0692%			
Filter 2	0.26914	0.000152	0.00022	0.0952%			
No Filter	0.26901[1]	0.000061	0.00009	0.0390%	6	0.26892	0.500
Filter 1	0.26903	0.000078	0.00011	0.0476%			
Filter 2	0.26906	0.000095	0.00014	0.0606%			

[1]No statistical difference at 95% level compared to Filter 1 case.

TABLE XXX

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 2 OF ZDT4 WITH A QUADRATIC UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.18071[1]	0.000151	0.00014	0.0226%			
Filter 1	0.18070	0.000126	0.00013	0.0210%	4	0.18057	0.800
Filter 2	0.18078	0.000173	0.00021	0.0339%			
No Filter	0.18066[1]	0.000081	0.00009	0.0145%			
Filter 1	0.18065	0.000062	0.00008	0.0129%	6	0.18057	0.800
Filter 2	0.18067	0.000068	0.00010	0.0161%			

[1]No statistical difference at 95% level compared to Filter 1 case.

TABLE XXXI

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 3 OF ZDT4 WITH A QUADRATIC UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.23301[1]	0.000098	0.00013	0.0312%			
Filter 1	0.23300	0.000100	0.00012	0.0288%	4	0.23288	0.650
Filter 2	0.23306	0.000147	0.00018	0.0432%			
No Filter	0.23297[1]	0.000055	0.00009	0.0216%			
Filter 1	0.23298	0.000071	0.00010	0.0240%	6	0.23288	0.650
Filter 2	0.23300	0.000087	0.00012	0.0288%			

[1]No statistical difference at 95% level compared to Filter 1 case.

TABLE XXXII

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 1 OF DTLZ1 WITH A QUADRATIC UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.09527[1]	0.000011	0.00001	0.0095%			
Filter 1	0.09528	0.000068	0.00002	0.0267%	4	0.09526	0.165
Filter 2	0.09532	0.000074	0.00006	0.0812%			
No Filter	0.09527[1]	0.000004	0.00000	0.0037%			
Filter 1	0.09527	0.000019	0.00001	0.0080%	6	0.09526	0.165
Filter 2	0.09527	0.000019	0.00001	0.0123%			

[1]No statistical difference at 95% level compared to Filter 1 case.

TABLE XXXIII

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 2 OF DTLZ1 WITH A QUADRATIC UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.07895	0.000006	0.00001	0.0033%			
Filter 1	0.07899	0.000070	0.00004	0.0237%	4	0.07895	0.250
Filter 2	0.07945	0.000906	0.00051	0.2956%			
No Filter	0.07895	0.000005	0.00000	0.0027%			
Filter 1	0.07902	0.000107	0.00008	0.0448%	6	0.07895	0.250
Filter 2	0.07926	0.000589	0.00031	0.1816%			

TABLE XXXIV

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 3 OF DTLZ1 WITH A QUADRATIC UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.04437	0.000008	0.00001	0.0024%			
Filter 1	0.04460	0.000538	0.00024	0.0786%	4	0.04436	0.350
Filter 2	0.04570	0.001675	0.00134	0.4392%			
No Filter	0.04437	0.000017	0.00001	0.0033%			
Filter 1	0.04462	0.000468	0.00026	0.0855%	6	0.04436	0.350
Filter 2	0.04560	0.001549	0.00124	0.4048%			

TABLE XXXV

DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 2 OF DTLZ2 WITH A QUADRATIC UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.20001[1]	0.000046	0.00001	0.0033%			
Filter 1	0.20004[2]	0.000083	0.00004	0.0133%	4	0.20000	0.500
Filter 2	0.20022	0.000913	0.00022	0.0733%			
No Filter	0.20001	0.000027	0.00001	0.0033%			
Filter 1	0.20006[2]	0.000153	0.00006	0.0200%	6	0.20000	0.500
Filter 2	0.20030	0.000913	0.00030	0.1000%			

[1]No statistical difference at 95% level compared to Filter 1 case.
[2]No statistical difference at 95% level compared to Filter 2 case.

statistically significant differences between No Filter and Filter 1 cases as well as between Filter 1 and Filter 2 cases. These results are not surprising since we may expect to converge the preferred regions with less preference information for less complicated underlying preference structures such as those represented by linear and quadratic utility functions.

E. Discussions

We demonstrate that iTDEA converges to the final preferred region of the DM interactively on the selected test problems with different types of utility functions. The algorithm finely converges to the correct region when the chosen best individual accurately indicates the true preferences of the DM. In addition, as the number of interactions with the DM increases, the final focus region becomes more accurate in general. However, when individuals are filtered before they are presented to the DM, the algorithm's performance may slightly deteriorate in the Tchebycheff utility function case. However, all versions of the algorithm perform satisfactorily with only several interactions.

There may be variations of our approach. We already mentioned that the level of automation can be reduced, allowing the DM to set the values of some of the parameters if he/she chooses to do so. It is also possible to allow the DM choose to continue the algorithm until he/she finds a satisfactory solution, rather than deciding on the number of interaction stages up front. This can be implemented with minor changes in the mechanisms of reducing territory sizes in consecutive iterations. The algorithm runs

TABLE XXXVI
DEVIATIONS FROM THE BEST SOLUTIONS FOR TEST 3 OF DTLZ2 WITH A QUADRATIC UTILITY FUNCTION

Solution	Mean	Standard Deviation	Absolute Deviation	Relative Deviation	Interactions	U^*	U^w
No Filter	0.10001[1]	0.000019	0.00001	0.0017%			
Filter 1	0.10001[2]	0.000015	0.00001	0.0017%	4	0.10000	0.700
Filter 2	0.10012	0.000450	0.00012	0.0200%			
No Filter	0.10000[1]	0.000013	0.00000	0.0000%			
Filter 1	0.10000[2]	0.000009	0.00000	0.0000%	6	0.10000	0.700
Filter 2	0.10009	0.000540	0.00009	0.0150%			

[1]No statistical difference at 95% level compared to Filter 1 case.
[2]No statistical difference at 95% level compared to Filter 2 case.

TABLE XXXVII
95% CONFIDENCE INTERVALS FOR LINEAR UTILITY FUNCTION (IN 10^{-5})

		Four Interactions		Six Interactions	
Problem	Test	No Filter vs. Filter 1	Filter 1 vs. Filter 2	No Filter vs. Filter 1	Filter 1 vs. Filter 2
ZDT1	1	(−7.3, 2.1)	(−13.2, −6.6)	(−4.5, 1.1)	(−4.4, −2.0)
	2	(−1.8, 0.5)	(−2.4, 0.2)	(−1.9, 0.4)	(−0.4, 0.0)
	3	(−0.8, 5.6)	(−16.9, −11.4)	(−2.4, 3.4)	(−7.4, −3.5)
DTLZ1	2	(−24.5, 0.3)	(−200.5, −21.5)	(−12.8, 3.1)	(−111.7, −31.4)
	3	(−3.3, 8.0)	(−103.5, 26.5)	(−3.6, 9.6)	(−210.5, 16.3)
DTLZ2	1	(−4.6, 1.6)	(−196.5, −21.0)	(−0.4, 2.7)	(−414.8, −86.9)
	2	(−318.6, −48.4)	(−2,110.0, −331.0)	(−272.1, −32.8)	(−2,160.0, −407.0)
	3	(−93.9, 29.6)	(−549.0, 151.0)	(−9.4, 21.6)	(−705.0, 198.0)

TABLE XXXVIII
95% CONFIDENCE INTERVALS FOR QUADRATIC UTILITY FUNCTION (IN 10^{-5})

		Four Interactions		Six Interactions	
Problem	Test	No Filter vs. filter 1	Filter 1 vs. Filter 2	No Filter vs. Filter 1	Filter 1 vs. Filter 2
ZDT1	1	(−5.3, 1.7)	(−7.7, −4.1)	(−4.0, 0.9)	(−4.4, −1.8)
	2	(−2.9, 3.4)	(−9.9, −5.6)	(−1.7, 2.9)	(−2.6, −1.5)
	3	(−1.6, 3.9)	(−8.3, −4.2)	(−3.4, 1.2)	(−3.1, −1.6)
DTLZ1	1	(−3.1, 0.8)	(−4.5, −3.1)	(−0.8, 0.3)	(−0.4, −0.2)
	2	(−5.5, −1.6)	(−71.7, −21.3)	(−10.2, −4.2)	(−38.7, −8.2)
	3	(−38.6, −8.0)	(−152.9, −67.4)	(−38.4, −11.9)	(−136.4, −58.7)
DTLZ2	2	(−5.4, 0.0)	(−43.5, 8.5)	(−8.8, −0.3)	(−49.8, 1.0)
	3	(−0.5, 0.6)	(−24.0, 1.6)	(−0.1, 0.4)	(−23.7, 7.0)

fast and the cognitive burden on the DM has been kept low deliberately. If the DM wishes to experiment and see several parts of the Pareto frontier, he/she can easily run the algorithm several times. Alternatively, it is possible for the DM to choose several solutions as best in any iteration and guide the algorithm to focus into several regions simultaneously. It is also possible to modify the algorithm so that it will allow the DM to change preferences and explore regions he/she did not like in previous iterations. As the algorithm focuses at preferred regions by obtaining denser solutions in those regions, it keeps generating a sparse set of solutions from other regions as well. At an interaction stage, we may present the DM two sets of solutions in this case. The first set would represent solutions from the preferred regions as before and the second set would represent solutions from regions the DM did not prefer in earlier interaction stages. If the DM decides to explore regions not preferred earlier, he/she would simply choose solutions from the second set. Another variation could be to allow the DM to set aside some solutions he/she found interesting at any stage. The algorithm could allow the DM to retrieve any such solutions in succeeding iterations, insert them in the current population and focus the search around some of those solutions. Implementing any of these variations are rather straight forward, thanks to the convenient contracting mechanism of the territories around the chosen solutions.

V. CONCLUSION

In this paper, we proposed an iTDEA. The algorithm interacts with the DM during optimization to identify his/her preferences. Using this information, it converges to the final preferred region of the DM.

The algorithm is tested on three problems with two and three objectives. We simulated the preferences of the DM with Tchebycheff, linear and quadratic utility functions. We observed that the algorithm accurately converges to the final preferred region of the DM in all settings of the utility function used, the number of objectives, and the shape of the Pareto-optimal frontier. All experiments are replicated with four and six interactions with the DM to investigate the number of interactions on the quality of the solutions. It was observed that more interactions with the DM yield better convergence. In addition, it helps the algorithm recover easier if it is misled in the early interactions.

During the interactions, it may not be practical for the DM to choose the best individual from the entire population. To address this issue, we filtered out a representative set to present the DM in each iteration stage. We observed that filtering slightly degrades the performance of the algorithm as expected. However, the results are still good in the absolute sense. It is also observed that filtering may mislead the algorithm in the early interactions. One method to avoid that is to increase the number of individuals presented to the DM so that the accuracy of the selected individual increases. However, this has a practical limit. Another way is to implement a better filtering mechanism to ensure that the selected individuals are a better representation of the population. An iterative selection method can also be employed to explore the population at hand.

It should be noted that the details of the algorithm can be implemented in various different ways. One such variation we intend to implement is related to the archive updating procedure. As in TDEA, we eliminated the individuals dominated by a candidate solution before territory violation check. Alternatively, those dominated solutions can be maintained in the archive if the candidate solution is not admitted to the archive. The performance of such a variation needs to be tested.

References

[1] 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.

[2] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proc. GECCO*, vols. 7–11. 2001, pp. 283–290.

[3] 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.

[4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Proc. Evol. Methods Design, Optimization Control Appl. Ind. Probl. (EUROGEN 2001)*, 2002, pp. 95–100.

[5] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. PPSN VIII*, LNCS 3242. Sep. 2004, pp. 832–842.

[6] 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.

[7] 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, Sep. 2007.

[8] B. Soylu and M. Köksalan, "A favorable weight based evolutionary algorithm for multiple criteria problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 2, pp. 191–205, Apr. 2010.

[9] 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.

[10] H. Tamaki, H. Kita, and S. Kobayashi, "Multiobjective optimization by genetic algorithms: A review," in *Proc. ICEC*, 1996, pp. 517–522.

[11] 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. 1999, pp. 3–13.

[12] K. Deb, "Evolutionary algorithms for multi-criterion optimization in engineering design," in *Evolutionary Algorithms in Engineering Computer Scienec*, K. Miettinen, M. M. Mäkelä, P. Neittaanmäki, and J. Periaux, Eds. Chichester, U.K.: Wiley, 1999, pp. 135–161.

[13] 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.

[14] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.

[15] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Berlin, Germany: Springer, 2006.

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

[17] 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.

[18] L. Rachmawati and D. Srinivasan, "Preference incorporation in multi-objective evolutionary algorithms: A survey," in *Proc. IEEE CEC*, 2006, pp. 3385–3391.

[19] 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., Part A: Syst. Hum.*, vol. 28, no. 1, pp. 26–37, Jan. 1998.

[20] J. Branke, T. Kaußler, and H. Schmeck, "Guidance in evolutionary multi-objective optimization," *Adv. Eng. Softw.*, vol. 32, no. 6, pp. 499–507, Jun. 2001.

[21] 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.

[22] K. Deb and J. Sundar, "Reference point based multiobjective optimization using evolutionary algorithms," in *Proc. GECCO*, vol. 1. Jul. 2006, pp. 635–642.

[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] K. Deb and A. Kumar, "Light beam search based multiobjective optimization using evolutionary algorithms," in *Proc. IEEE CEC*, 2007, pp. 2125–2132.

[25] M. Köksalan, "Multiobjective combinatorial optimization: Some approaches," *J. Multi-Criteria Decision Anal.*, vol. 15, nos. 3–4, pp. 69–78, 2009.

[26] I. C. Parmee, D. C. Cvetković, A. H. Watson, and C. R. Bonham, "Multiobjective satisfaction within an interactive evolutionary design environment," *Evol. Comput.*, vol. 8, no. 2, pp. 197–222, 2000.

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

[28] H. A. Abbass, "An economical cognitive approach for bi-objective optimization using bliss points, visualization, and interaction," *Soft Comput.—A Fusion Found., Methodol. Appl.*, vol. 10, no. 8, pp. 687–698, 2006.

[29] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz, "Interactive evolutionary multiobjective optimization using robust ordinal regression," in *Proc. 5th Int. Conf. EMO*, 2009, pp. 554–568.

[30] K. Deb and A. Kumar, "Interactive evolutionary multiobjective optimization and decision-making using reference direction method," in *Proc. GECCO*, vol. 1. Jul. 2007, pp. 781–788.

[31] 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.

[32] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based interactive evolutionary algorithm for multiobjective optimization," *Evol. Comput.*, vol. 17, no. 3, pp. 411–436, 2009.

[33] İ. Karahan and M. Köksalan, "A territory defining multiobjective evolutionary algorithm and preference incorporation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 4, pp. 636–664, Aug. 2010.

[34] R. E. Steuer and E.-U. Choo, "An interactive weighted Tchebycheff procedure for multiple objective programming," *Math. Progr.*, vol. 26, no. 3, pp. 326–344, Oct. 1983.

[35] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," *Proc. IEEE*, vol. 89, no. 9, pp. 1275–1296, Sep. 2001.

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

[37] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA: Kluwer, 1999.

[38] J. T. Buchanan and H. G. Daellenbach, "A comparative evaluation of interactive solution methods for multiple objective decision models," *Eur. J. Oper. Res.*, vol. 29, no. 3, pp. 353–359, Jun. 1987.

[39] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," Comput. Eng. Netw. Lab.

(TIK), Swiss Federal Instit. Technol. (ETH), Zurich, Switzerland, Tech. Rep. 112, 2001.

[40] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.

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

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 from the University of Illinois at Urbana-Champaign, Urbana.

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

Murat Köksalan is currently a Professor with the Department of Industrial Engineering, Middle East Technical University, Ankara, Turkey. He has been a Visiting Professor with the State University of New York at Buffalo, Buffalo, Purdue University, West Lafayette, IN, Helsinki University of Technology, Helsinki, Finland, and the Helsinki School of Economics, Helsinki, on various occasions. His current academic interests include multiple criteria decision making, combinatorial optimization, heuristic search, combinatorial auctions, and preparing teaching cases.

He is the recipient of the Young Researcher Award of the Scientific and Technological Research Council of Turkey and the MCDM Gold Medal of the International Society on Multiple Criteria Decision Making. He is the Founding President of the INFORMS Section on MCDM. He received the First Prize at the INFORMS Case Competition with various co-authors three times.