An Improved Progressively Interactive Evolutionary Multi-objective Optimization Algorithm with a Fixed Budget of Decision Maker Calls

Ankur Sinha, Pekka Korhonen, Jyrki Wallenius

Firstname.Secondname@aalto.fi,
Department of Information and Service Economy, Aalto University School of Economics
PO Box 21220, 00076 Aalto, Helsinki, Finland

Kalyanmoy Deb[1]

Deb@iitk.ac.in
Department of Mechanical Engineering, Indian Institute of Technology Kanpur
PIN 208016, Kanpur, India

KanGAL Report Number: 20110014

Abstract

This paper presents a preference-based method to handle problems with a large number of objectives. With an increase in number of objectives the complexity of the problem rises exponentially and it becomes difficult for evolutionary multi-objective techniques to produce the entire front. In this paper an evolutionary multi-objective procedure is hybridized with preference information from the decision maker during the intermediate runs of the algorithm. The preference information from the decision maker helps in guiding the algorithm towards the most preferred point on the front. The methodology is an improvement of an earlier progressively interactive approach which uses implicitly defined value functions. The proposed approach offers multiple advantages over the previous technique with the most important advantage being optimizing the problem in a fixed budget of decision maker calls. In the previous method, there was no control over the number of decision maker calls required to optimize the problem. The suggested approach works by constructing polyhedral cones which are used to modify the

[1]Also, Department of Information and Service Economy, Aalto University School of Economics, PO Box 21220, 00076 Aalto, Helsinki, Finland

domination principle and conduct a focussed search. The methodology has been tested, and comparison has been performed with the earlier approach, on two to five objectives unconstrained as well as constrained test problems.

Keywords:
Evolutionary multi-objective optimization, multiple criteria decision-making, interactive multi-objective optimization, sequential quadratic programming, preference based multi-objective optimization.

1. Introduction

The evolutionary multi-objective optimization (EMO) algorithms have demonstrated their ability in solving difficult and complicated multiple objective problems [1, 2]. They have been shown to handle problems with two to three objectives effectively, but thereafter, the deterioration in performance becomes noticeable both in terms of convergence and diversity. In this paper, we introduce a methodology which can be integrated with any evolutionary multi-objective optimization algorithm allowing it to effectively handle problems with large number of objectives.

Commonly used EMO algorithms find well spread solutions close to the Pareto-optimal front, the decision maker (DM) then chooses the most suitable point from the objective space and implements the design based on the decision variables corresponding to that point. However, in this paper we propose to integrate the decision maker with the optimization run of an EMO algorithm in a way such that the preferences of the decision maker can be incorporated into the intermediate generations of the algorithm, and progress towards the most preferred point is made. The most preferred point is the point on the Pareto-optimal front which gives maximum utility/satisfaction to the decision maker when compared with other points on the front. It is the point which the decision maker will choose if he/she is provided with an entire set of perfect Pareto-solutions. This point is of course unknown at the start of the optimization run and the proposed algorithm tries to get as close to this point as possible based on the preference information provided by the decision maker. Such a procedure where a decision maker is involved in the intermediate generations of an EMO algorithm is called a *progressively interactive EMO approach* (PI-EMO). Some of the recent studies using PI-EMO approach are [4, 5, 7, 6]. A progressively interactive approach is a decision maker-oriented approach which allows the decision maker to guide the algorithm towards a point which he/she likes the best. Seeking the most preferred point, instead of the entire Pareto-optimal front, saves us from the intricacies involved in generating the multi-dimensional front.

The paper proposes a simple PI-EMO approach which seeks preference information periodically from the decision maker in the form of the best point from a provided set. The information obtained from the decision maker is used to construct a polyhedral cone, and modify the domination principle which helps in getting close to the most preferred point. Before the start of the optimization process the algorithm takes as input, the number of times the decision maker will be made available to provide preference information. Thereafter, the algorithm tries to optimize the problem in the provided budget of DM calls [2]. A fine-tuning strategy is also incorporated in the algorithm which allows a better utilization of the budget of decision maker calls. The concept has been integrated with the NSGA-II algorithm [9] and the working of the algorithm has been demonstrated on three unconstrained and two constrained test problems having two, three and five objectives. A comparison of the algorithm has been done with the progressively interactive EMO approach using value functions, and the efficacy of the proposed procedure has been confirmed.

2. Multi-Objective Optimization Problems

In a multi-objective optimization problem there are two or more objectives which are conflicting, and are supposed to be simultaneously optimized subject to a given set of constraints. These problems can be commonly found in the fields of science, engineering, economics or any other field where optimal decisions are to be taken in the presence of trade-offs between two or more conflicting objectives. Usually such problems do not have a single solution which simultaneously maximizes/minimizes each of the objectives; instead there is a set of solutions which are optimal. These optimal solutions are called Pareto-optimal solutions. There does not exist any other solution in the feasible set which is better than a Pareto solution in terms of all the objectives. A general multi-objective problem can be described as follows:

[2] A decision maker call or a DM call is an event where the algorithm seeks preference information from the decision maker (DM)

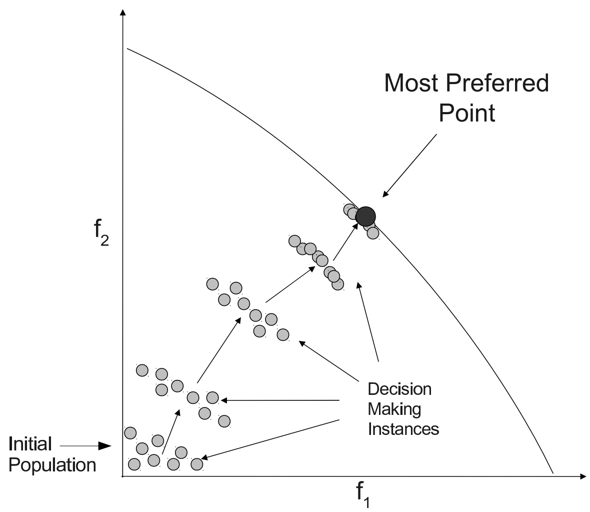

Figure 1: Progressively interactive approach to handle a multi-objective optimization problem.

$$\begin{array}{ll} \text{Maximize} & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_M(\mathbf{x}))\,, \\ \text{subject to} & \\ & \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0}, \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, \ldots, n. \end{array} \tag{1}$$

Usually EMO algorithms have been oriented towards exploring the entire set of solutions, however, there is just a single solution which is the most preferred for a decision maker and he/she selects a solution from the presented set manually or using a decision making aid. This can be a preferred approach as long as the EMO algorithms are capable of finding the entire set of solutions for the multi-objective problem, but the EMO algorithms suffer when the number of objectives is high. We draw our motivation from the fact that the decision maker is finally going to choose just one point from the Pareto-optimal set, therefore, it would be a wise idea to explore the most preferred point instead of the entire Pareto-optimal set. Aided by preference information from a decision maker at different stages of the algorithm, an EMO can be made to explore the most preferred solution, which using such a strategy can be found with less computational expense and a high accuracy for many objective problems. It is noteworthy that not many studies exist which integrate an EMO algorithm with a decision maker. Using a progressively interactive approach, the decision maker can be easily integrated with an existing evolutionary multi-objective optimization algorithm. The working of such an approach can be observed from Figure 1. It can be observed from the figure that preference information is accepted from a decision maker at various stages of the algorithm and the population converges towards the most preferred point. The existing EMO algorithms would have produced a diverse set of solutions over the entire Pareto-optimal front, however, in a progressively interactive approach the algorithm performs a focussed search only in the region of interest to the decision maker and converges towards a single solution. In the next section, we provide a survey of some of the recently proposed progressively interactive methods, and thereafter we discuss the algorithm proposed in this paper.

3. A Survey of Progressively Interactive Methods

Not many studies are yet available in the direction of progressive use of preference information provided by the decision maker during the intermediate generations of an algorithm. Some of the studies from the recent past provide the decision maker a number of points in objective space and accept preference information from him/her. A few attempts made in the direction of progressively interactive methods are by Phelps and Köksalan [10], Fowler et al. [17], Jaszkiewicz [11], Branke et al. [7] and Korhonen, Moskowitz and Wallenius [8].

In a recent study [4] the authors have proposed a progressively interactive EMO using value functions (PI-EMO-VF) where the information from the decision maker is elicited and used to construct a value function. In this approach, at different stages of the algorithm, an implicit value function is defined, which maps the preference information provided by the decision maker and is subsequently used for making decisions. Including PI-EMO-VF, most of the other previous studies employ a value function for convergence towards the most preferred point. Another recently suggested methodology [6] eliminates the process of constructing a value function by using polyhedral cones, and we extend the work in this paper. The method proposed in this paper uses preference information from the decision maker in the form of the best point from a given set. This information is used to formulate a polyhedral cone which is used to drive the EMO procedure towards the region of interest.

The PI-EMO-VF approach suffers from a few drawbacks which have been improved in this paper. Firstly, in the earlier approach, the number of decision maker calls required is unknown at the start of the algorithm, as there is no means to limit the number of calls . Secondly, the method requires a partial or complete ordering of a given set of points, with the set of points to be ordered being a parameter in the algorithm. Thirdly, another optimization problem is required to be solved to construct the decision maker's value function. Fourthly, the number of generations after which the decision maker call is made is another parameter in the algorithm. The proposed method is able to successfully handle these drawbacks by eliminating a number of parameters and offering an improved performance at the same time. The algorithm takes into consideration the time of the decision maker which is usually valuable, and optimizes the problem within a specified budget of decision maker calls.

4. Incorporating Decision Maker's Preference Information

There are various ways to elicit information from a decision maker. One approach, suggested in [4], is to provide a small set of points to a decision maker and ask for a complete or a partial ordering of the points. Based on this information a value function is fitted, which requires solving an optimization problem. The optimization problem, though simple to solve, calls for additional computational resources. Moreover, there are a few parameters, as to how many points to give to a decision maker to generate a value function. Any error made by the decision maker during the ranking of the points may distort the shape of the value function or may not allow a value function to be fitted. The approach, though elegant, suffers from these drawbacks. This motivates us to think of a different approach to elicit information from a decision maker and incorporate it into an EMO algorithm. Another strategy which one can readily think of is asking a decision maker to choose the best point from a given set of points. Various decision making tools can be utilized here to help the decision maker choose the best point. Based on the choice of the best point by the decision maker, the idea is to restrict the search of the EMO algorithm in the region of interest. However, with a limited information available, it is not possible to exactly determine the region of interest. In this section, we aim to approximate the region with the help of polyhedral cones formed with hyperplanes in higher dimensions. The polyhedral cone provides only an approximation of the preferred region. However, the approximation is sufficient to guide an EMO algorithm towards the most preferred point by eliciting preference information from the decision maker at various stages of the algorithm.

Before discussing the procedure for incorporating preference information, we shall introduce some concepts used in the paper.

4.1. Cones

A cone is a nonempty set $C \subset \mathbb{R}^M$ for which $c \in C \Rightarrow \lambda c \in C$ where $\lambda > 0$. A cone is convex if $c^{(1)}, c^{(2)} \in C \Rightarrow c^{(1)} + c^{(2)} \in C$. In the following definitions we define a polyhedral cone and a polyhedral set.

Definition 1. *Polyhedral Cone: If $A \subset \mathbb{R}^{L \times M}$ is a matrix, then a polyhedral cone $C(A) \subset \mathbb{R}^M$ determined by A is defined as:*

$$C(A) := \{c \in \mathbb{R}^M : Ac \geq 0\} \tag{2}$$

Definition 2. *Polyhedral Set: If $A \subset \mathbb{R}^{L \times M}$ is a matrix, and $b \subset \mathbb{R}^L$ is a vector, then a polyhedral set $C(A, b) \subset \mathbb{R}^M$ determined by A and b is defined as:*

$$C(A, b) := \{c \in \mathbb{R}^M : Ac \geq b\} \tag{3}$$

A polyhedral cone is a special kind of cone which is defined by a solution set of a homogeneous system of linear inequalities. If $rank(A) = M$, then the polyhedral cone is referred as a pointed polyhedral cone. In M dimensions polyhedral cones may be constructed with any finite number of bounding hyperplanes. However, in the context of this paper we shall consider only pointed polyhedral cones constructed with M different hyperplanes in M dimensions. A polyhedral set is an intersection of a finite number of halfspaces and is always convex. A polyhedral cone represents a set of directions in the hyperspace, while a polyhedral set represents a set of points in the hyperspace.

In multi-objective optimization problems we wish to optimize all the objectives where the optimality in the M dimensional objective space is determined by a cone. The simplest polyhedral cone can be defined by considering $A = I^{M \times M}$ which yields an orthant in $\mathbb{R}^M$. This cone is known as Pareto-cone and is commonly used in multi-objective evolutionary algorithms to generate the entire Pareto-frontier. For problems where all objectives are to be maximized, the Pareto domination cone D_{par} is defined as:

$$D_{par} := \{c \in \mathbb{R}^M : Ic \leq 0\} \tag{4}$$

and the Pareto preference cone P_{par} is defined as:

$$P_{par} := \{c \in \mathbb{R}^M : Ic \geq 0\} \tag{5}$$

It is clear from the above definitions that $P_{par} = -D_{par}$. The directions suggested by domination cone leads to dominated solutions and the directions suggested by preference cone leads to preferred solutions. It is straightforward to generalize this to any polyhedral cone and define domination cone and preference cone. The idea of domination cone was introduced by [20]. A domination cone $D_c \subset \mathbb{R}^M$ for maximization problems can be defined as follows:

$$D_c(A) := \{c \in \mathbb{R}^M : Ac \leq 0\} \tag{6}$$

A domination cone represents all the directions which are poor. Similarly, a preference cone $P_c \subset \mathbb{R}^M$ can be defined by simply considering $P_c = -D_c$. A preference cone represents all the directions which are good. This result is well known in the multi-criteria decision making literature [19, 21, 22, 23]

In evolutionary algorithms, at any generation we have a population of non-dominated solutions. If one of the solutions is known to be the best in the set, then we wish to create a polyhedral cone to find the preferred region. There can be various strategies to create a polyhedral cone, and in this section we state two such strategies. Figures 2 and 3 show the non-dominated population members at a particular generation of the algorithm. From among the members searched so far, the decision maker chooses the best point. Using the best point and other population members, a polyhedral cone can be constructed. Figure 2 shows a strategy where polyhedral cone is constructed using the best point and the end points of the non-dominated front. The second strategy in Figure 2 shows a polyhedral cone constructed from the best point and the means of other members in the population. Similarly, some other strategy could be thought of, which provides a preferred region based on the best point. It is difficult to say which strategy better approximates the preferred region, as in absence of complete information this answer is not known. However, these approximations are sufficient to guide an EMO algorithm towards the most preferred point, if elicitations are made progressively from the decision maker during the course of an optimization run. In this paper, we choose to construct the polyhedral cones based on first strategy, and shall discuss the construction procedure in higer dimensions in the later part of the paper.

5. Progressively Interactive EMO Based on Polyhedral Cones (PI-EMO-PC)

In this section, we propose a progressively interactive EMO algorithm (PI-EMO-PC) which uses the polyhedral cone to modify the domination criteria of an EMO and drives it towards a single most preferred point on an high dimensional Pareto-optimal frontier of an M objective problem. The algorithm takes as input

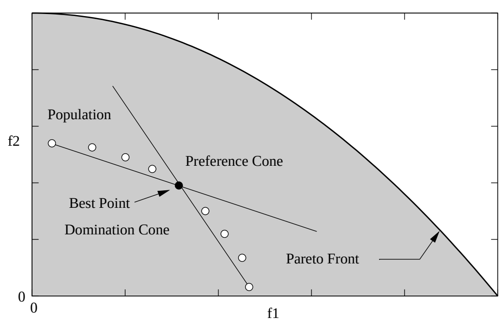

Figure 2: Polyhedral cone construction with best point and end points.

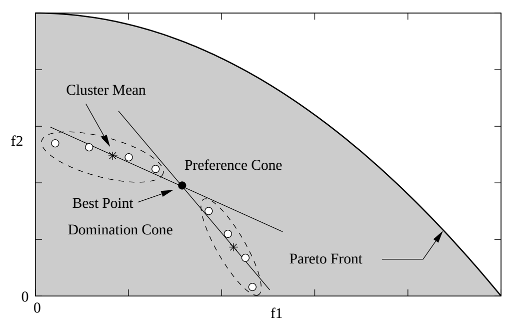

Figure 3: Polyhedral cone construction with best point and cluster means.

a parameter T_{DM}, which is the maximum number of times preference information can be elicited from the decision maker.

The algorithm requires the ideal point $\vec{I}$ to start. The ideal point is determined by maximizing each of the objectives individually. An individual maximization of the objectives gives M points $\vec{P_1}, \vec{P_2}, \ldots, \vec{P_M}$. A two-objective case has been shown in Figure 4 where maximization of the objectives gives 2 points $\vec{P_1}$ and $\vec{P_2}$. It should be noted that a single objective optimization will give any of the weak Pareto points for either of the objectives. [3]

Once the ideal point is known, the initial random population is created. The point ($\vec{P^{ib}}$) closest to the ideal point ($\vec{I}$) is chosen and its distance from the ideal point is denoted as $D_I = |\vec{I} - \vec{P^{ib}}|$. $\vec{D_I} = \vec{I} - \vec{P^{ib}}$ denotes the vector from $\vec{P^{ib}}$ to $\vec{I}$. This distance D_I is divided into $d_I = \frac{D_I}{T_{DM}+1}$ equal parts. This is done to make sure that the preference information from the decision maker is elicited only after a progress of d_I has been made. After the creation of a random population the EMO algorithm is run, until a point is found which dominates $\vec{P^{ib}}$, is closest to $\vec{I}$, and whose projected distance from the point $\vec{P^{ib}}$ on the vector $\vec{D_I}$ is more than d_I. The number of function evaluations required to make such progress is stored as f_I. During the progress the algorithm stores all non-dominated solutions being produced in an archive set. For a two objective case this improvement has been represented from point $\vec{A}$ to $\vec{B}$ in Figure 4. This initial process gives an idea of

[3]If the single objective optimization of individual objectives produces points for each of the objectives that are not weakly dominated, then we also get the Nadir Point in the objective space along with the ideal point.

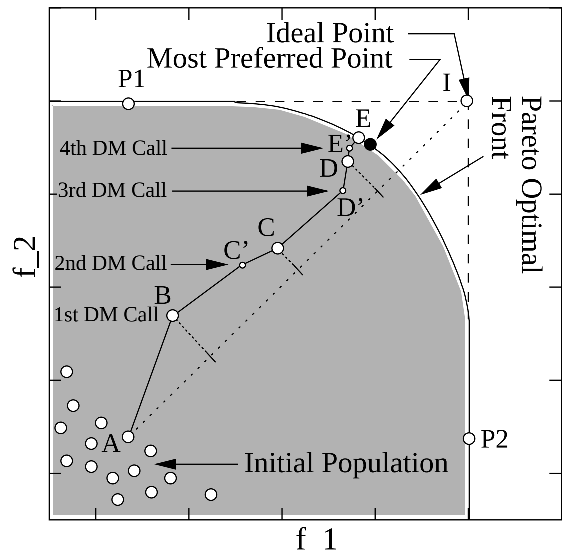

Figure 4: Progress of the algorithm on a two objective problem with preference information elicitation at B, C and D.

the approximate number of function evaluations needed to make a progress of d_I.

After this the decision maker is called and preference information is elicited. A polyhedral cone, which is used to modify the domination principle (discussed in Section 5.4), is formed based on the information. The preference information is elicited from the decision-maker by providing an archive set containing non-dominated solutions and the decision maker is expected to choose the best solution from this set. The best solution $A_1^{\vec{best}}$ is picked using an advanced selection technique known as VIMDA [18] [4].

The modified domination criterion is used for the next few generations of the EMO run until f_I number of function evaluations are done. After this the decision maker is again called for the preference information and the best solution $A_c^{\vec{best}}$ is elicited. Distance between $A_c^{\vec{best}}$ and $A_{c-1}^{\vec{best}}$ is computed and its projection is taken on the vector $\vec{D_I}$. Here subscript c denotes the decision maker call. If the projected

[4]It is a visual interactive method which uses the reference point technique to allow the decision maker to select the best point from a set of non-dominated points. Its efficacy could be well demonstrated for high dimensional objective vectors when a geometrical representation is not possible.

distance is found to be less than d_I then a local search (discussed in Section 5.2) is done along the search direction determined from the polyhedral cone using the current best solution $\vec{A_c^{best}}$ as the reference point. In every iteration of the local search $\vec{A_c^{best}}$ is updated with the better solution found. The local search is stopped if the distance between current best and $\vec{A_{c-1}^{best}}$ exceeds d_I. For a two objective case, the improvements have been shown in Figure 4 as progress from point $\vec{B}$ to $\vec{C'}$ for EMO search and $\vec{C'}$ to $\vec{C}$ for local search. After this the EMO generations are started and the process discussed in this paragraph is iterated. If the local search terminates because of no improvement in solution then the algorithm is terminated which has been represented in Figure 4 as point $\vec{E}$. Point $\vec{E}$ is the final outcome of the algorithm which is quite close to the most preferred point shown in the figure. More decision maker calls ensure that the algorithm moves towards the Pareto-front with smaller steps and a higher accuracy is achieved.

The principle may be integrated with any standard EMO algorithm (such as NSGA-II [9], SPEA-II [12] and others) which works with a population of points in each iteration and prefers a sparse set of non-dominated points in a population. The integration of the above principle with an EMO algorithm modifies the working of the algorithm and helps in finding the most preferred solution instead of the entire Pareto-optimal set.

A step-by-step generic procedure of the proposed PI-EMO-PC methodology is as follows:

Step 1: Initialize a population Par_0 and set iteration counter $t = 0$. Initialize archive set A [5]. Determine the point closest ($\vec{P^{ib}}$) to the ideal point $\vec{I}$ and find the vector $\vec{D_I} = \vec{I} - \vec{P^{ib}}$. Store the distance $d_I = \frac{D_I}{T_{DM}}$.

Step 2: Increment the counter as $t \leftarrow t + 1$ and execute the EMO algorithm generation with the usual definition of dominance [13]. The best member $\vec{P^{cb}}$ in the current population is the one closest to the ideal point which dominates $\vec{P^{ib}}$. All the feasible non-dominated solutions found at the end of the generation are added to the archive A. The archive, at each iteration, is updated by removing the non-dominated solutions. If the archive size exceeds $|A|^{max}$, k-mean clustering is used to keep the diverse set of $|A|^{max}$ clusters and rest of the solutions are deleted.

[5]The initial size of the archive is $|A| = 0$. The maximum size the archive can have is $|A|^{max}$.

Step 3: If projected distance of $\vec{P^{cb}} - \vec{P^{ib}}$ on $\vec{D_I}$ is less than d_I then go to Step 2 otherwise store the function evaluations required during the improvement as f_I. Initialize c as 1 and call the decision maker to choose the best solution $A_c^{\vec{best}}$, from the archive A_t using VIMDA. Here c is the decision maker call. Choose the end points from the non-dominated front of the current parent population Par_t as rest of the solutions. This makes the chosen solution count as $\eta = M + 1$.

Step 4: Construct the sides of the polyhedral cone from the chosen set of $\eta = M + 1$ points, described in Section 5.2. Set the function evaluation counter $f = 0$.

Step 5: An offspring population, Off_t is created from the parent population, Par_t by using a modified domination principle (discussed in Section 5.4) based on the current polyhedral cone and crossover-mutation operators. Increment the function evaluation counter f by number of function evaluations done in this step.

Step 6: From parent population, Par_t and offspring population, Off_t a new population Par_{t+1} is determined, using modified domination and EMO algorithm's diversity preserving operator. The iteration counter is incremented as $t \leftarrow t+1$ and the algorithm moves to Step 5 if f is less than f_I; otherwise it proceeds to Step 7.

Step 7: c is incremented by 1 and the decision maker is called to choose the best solution $A_c^{\vec{best}}$, from the archive A_t. The previous best solution chosen by the decision maker is stored as $A_{c-1}^{\vec{best}}$. The polyhedral cone is constructed and the search direction is determined. Projected distance of vector $A_c^{\vec{best}} - A_{c-1}^{\vec{best}}$ on $\vec{D_I}$ is computed and stored as d_A.

Step 8: If the projected distance d_A is less than d_I then a local search is performed with $A_c^{\vec{best}}$ as the reference point along the search direction. The best solution, $A_c^{\vec{best}}$, chosen by the decision maker is updated at each iteration of local search. The local search is stopped if $|A_c^{\vec{best}} - A_{c-1}^{\vec{best}}| > d_I$, the function evaluation counter is reset as $f = 0$, and the algorithm proceeds to Step 5. If local search is unable to produce better solutions then the algorithm is terminated and the current best solution is chosen as the final outcome.

The PI-EMO-PC algorithm requires the parameters $|A|^{max}$ and T_{DM} in addition to the EMO algorithm's parameters.

5.1. Fine-tuning

Often it will be found while solving a problem that not the entire budget of function evaluations are utilized i.e. $c < T_{DM}$ at the end of the algorithm. In this case we recommend to move back at A^{best}_{α} where $\alpha = c - integer(\frac{T_{DM}-c}{2})$, update $d_I = \frac{d_I}{2}$ and start the algorithm from that point again. To implement fine-tuning, it is necessary to store the population members and the archive set at the end of each local search i.e Step 8 of the algorithm above. The stored parent population and the archive set is used to start the fine-tuning algorithm with a smaller d_I value.

The above fine-tuning procedure can also be utilized if the decision maker is unhappy with the final solution. In that case the decision maker specifies T'_{DM} and the algorithm moves back to A^{best}_{α} where $\alpha = c - integer(\frac{T'_{DM}}{2})$ and d_I is updated as $d_I = \frac{d_I}{2}$.

5.2. Polyhedral Cone

At an instance of a DM call, $M + 1$ members need to be selected to create M different hyperplanes which form the sides of the polyhedral cone. The decision maker is asked to choose the best solution from the archive set and the end points (M in number) of the non-dominated front are selected as other members of the cone. The end points of the front are those members in the archive which have maximum value for one of the objectives. Therefore, an M objective front will have M number of end points and including the best point chosen by the decision maker we have $M + 1$ points. Now, $M + 1$ number of different hyperplanes can be constructed using these $M + 1$ points in an M-dimensional hyperspace. From the set of $M + 1$ hyperplanes the hyperplane not containing the best point from the archive is removed which leaves us with remaining M hyperplanes. Since all the points are non-dominated with respect to each other, the normals to all the M hyperplanes will have positive direction cosines. The M planes together form a polyhedral cone with the best member from the archive as the vertex. Each hyperplane represents one of the sides of the polyhedral cone.

For instance Figure 5 and Figure 6 show the polyhedral cones in two and three dimensions. The equation of each hyperplane can be written as $P_i(f_1, \ldots, f_M) = 0, i \in \{1, \ldots, M\}$. If a given point $(f_1^{(1)}, \ldots, f_M^{(1)})$, in the objective space has $P_i > 0 \; \forall \; i \in \{1, \ldots, M\}$, then the point lies inside the polyhedral cone; otherwise it lies outside. In Figure 5, the shaded region has $P_i < 0$ for at least one $i \in \{1, \ldots, M\}$

and the unshaded region has $P_i > 0\,\forall\, i \in \{1,\ldots,M\}$. In Figure 6, the shaded polyhedral cone represents $P_i < 0\,\forall\, i \in \{1,\ldots,M\}$ and the unshaded polyhedral cone represents $P_i > 0\,\forall\, i \in \{1,\ldots,M\}$.

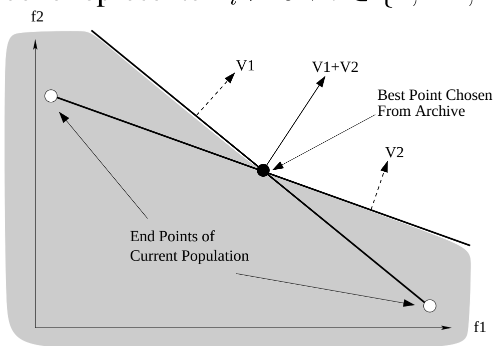

Figure 5: Cone in two dimensions.

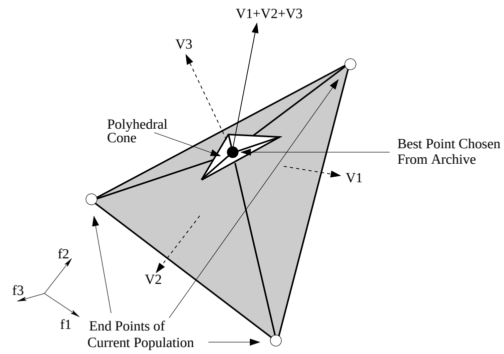

Figure 6: Polyhedral cone in three dimensions.

5.3. Local Search

Once the polyhedral cone is determined, it provides an idea for a search direction. The normal unit vectors ($\hat{V}_i$) of all the M hyperplanes can be summed up to get a search direction ($\vec{W} = \sum_{i=1}^{n} \hat{V}_i$). $\hat{W}$ is the unit vector along $\vec{W}$ and $W_i\ \forall\ i \in \{1,\ldots,M\}$ represents the direction cosines of the vector $\vec{W}$. This direction has been used to determine if the optimization process should be terminated or not. To implement this idea we perform a single-objective search along the identified direction.

We solve the following achievement scalarizing function (ASF) problem [14] for the best point from the archive $A_c^{\vec{best}} = \mathbf{z}^b$:

$$\begin{array}{ll} \text{Maximize} & \left(\min\limits_{i=1}^{M} \frac{f_i(\mathbf{X})-z_i^b}{W_i}\right) + \rho \sum_{j=1}^{M} \frac{f_j(\mathbf{X})-z_j^b}{W_j}. \\ \text{subject to} & \mathbf{x} \in \mathcal{S}. \end{array} \tag{7}$$

In the above formulation $\mathcal{S}$ denotes the feasible decision variable space of the original problem. The second term has a small constant ρ ($= 10^{-10}$ is suggested) which prevents the method from converging to a weak Pareto-optimal point. The sequential quadratic programming (SQP) optimization method is used to solve the above problem and the intermediate solutions ($\mathbf{z}^{(i)},\ i = 1, 2, \ldots$) are recorded. If at any intermediate point, $\frac{(\mathbf{Z}^{(i)} - A_{c-1}^{\vec{best}}).\vec{D_I}}{D_I}$ is larger than d_I, the achievement scalarizing function optimization is stopped and we continue with the EMO algorithm.

In this case, we replace $A_c^{\vec{best}}$ with $\mathbf{z}^{(i)}$ in the archive set, and update the archive set A_t, by deleting the dominated members. $A_c^{\vec{best}}$ replaces the member closest to it in the parent population Par_t. Figure 7 depicts this scenario. On the other hand, if at the end of the SQP run, the final SQP solution (say, $\mathbf{z}^T$) does not meet the criteria $\frac{(\mathbf{Z}^{(i)} - A_{c-1}^{\vec{best}}).\vec{D_I}}{D_I} > d_I$, we terminate the EMO algorithm and declare $\mathbf{z}^T$ as the final preferred solution. This situation indicates there does not exist any solution in the search space along the search direction which can further improve the best solution obtained so far. Hence, we can terminate the optimization run. Figure 8 shows such a situation, warranting a termination of the PI-EMO procedure.

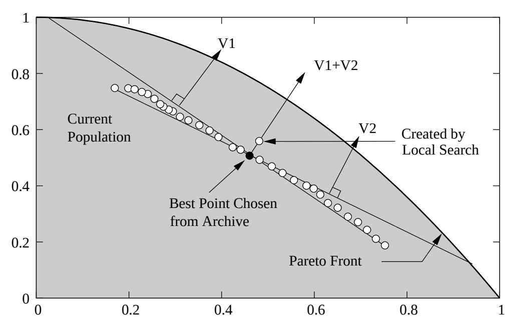

Figure 7: Local search, when far away from the front, finds a point which meets the distance criteria. Hence, no termination of the PI-EMO-PC.

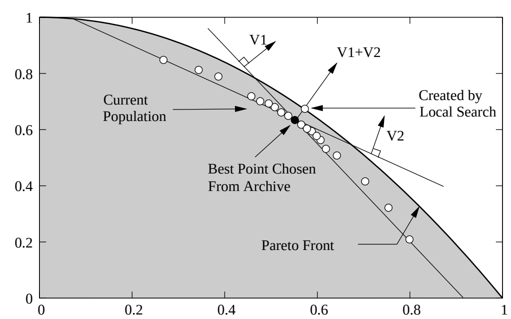

Figure 8: Local search does not find a point which meets the distance criteria. Hence, the PI-EMO-PC is terminated.

5.4. Modified Domination Principle

The polyhedral set has been used to modify the domination principle in order to emphasize and create preferred solutions.

Let us assume that the polyhedral set from the most recent decision-making interaction is represented by a set of hyperplanes $P_i(f_1, \ldots, f_M) = 0$ for $i \in \{1, \ldots, M\}$. Then, any two feasible solutions $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ can be compared with their objective function values by using the following modified domination criteria:

1. If for both the solutions $P_i(f_1, \ldots, f_M) > 0 \, \forall \, i \in \{1, \ldots, M\}$, then the two points are compared based on the usual dominance principle.
2. If for both the solutions $P_i(f_1, \ldots, f_M) < 0$ for at least one $i \in \{1, \ldots, M\}$, then the two points are compared based on the usual dominance principle.

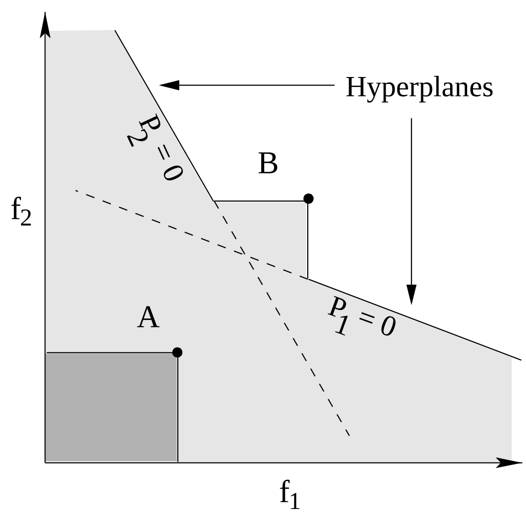

Figure 9: Dominated regions for two points A and B using the modified definition principle.

3. If one solution has $P_i(f_1, \ldots, f_M) > 0 \, \forall \, i \in \{1, \ldots, M\}$, and the other solution has $P_i(f_1, \ldots, f_M) < 0$ for at least one $i \in \{1, \ldots, M\}$, then the former dominates the latter.

Figure 9 illustrates the region dominated by two points A and B. The set formed by the linear equations has been shown. The point A lies in the region in which $P_i(f_1, \ldots, f_M) < 0$ for at least one $i \in \{1, \ldots, M\}$. The region dominated by point A is shaded. This dominated area is identical to that which can be obtained using the usual domination principle. However, point B lies in the region $P_i(f_1, \ldots, f_M) > 0 \quad \text{for} \quad i \in \{1, \ldots, M\}$. For this point, the dominated region is different from that which would be obtained using the usual domination principle. In addition to the usual region of dominance, the dominated region includes all points which have $P_i(f_1, \ldots, f_M) < 0$ for at least one $i \in \{1, \ldots, M\}$.

The constrained handling mechanism is used as defined in [9]. When two solutions under consideration for a domination check are *feasible*, then the above domination principle is used. If one point is feasible and the other is infeasible then the feasible solution is considered to be dominating the infeasible solution. If both points are infeasible then the one having smaller overall constraint violation is declared to be dominating the solution with higher overall constraint violation.

6. Implementation of the Algorithm

The PI-EMO-PC procedure has been implemented on the NSGA-II algorithm. However, it is possible to integrate the procedure with any other multi-objective EMO algorithm. Firstly a few generations of the algorithm, as discussed in the steps above, are performed according to the usual NSGA-II algorithm [9] and the archive set is maintained. Thereafter, the NSGA-II algorithm is modified by using the modified domination principle (discussed in Section 5.4) in the elite-preserving operator and tournament selection for creating the offspring population. The NSGA-II recombination operator (SBX) has been used in this study without any modification. The crowding distance operator of NSGA-II is not used in this implementation and has been replaced with k-mean clustering for maintaining diversity among solutions of the same non-dominated front.

An archive A is maintained which contains all the non-dominated members found in the current as well as the previous iterations of the optimization run. The maximum size an archive can have is $|A|^{max}$. Archiving makes sure that none of the non-dominated solutions generated is lost even if the decision maker makes an error while providing preference information.

For local search (discussed in Section 5.3), the SQP code of KNITRO [15] software has been used to solve the single objective optimization problem. The SQP algorithm is terminated either when $\frac{(\mathbf{Z}^{(i)} - \vec{A_{c-1}^{best}}).\vec{D_I}}{D_I} > d_I$ or when the KKT error measure is less than or equal to 10^{-6}. If the local search terminates due to KKT measure, then the overall PI-EMO-PC algorithm gets terminated. Therefore, the procedure guarantees that the final point achieved is one of the Pareto-optimal points.

As already discussed, we need to find out the ideal point solution before the start of the algorithm. We use the parent centric crossover (PCX) based single objective algorithm [3] for the single objective optimization. The algorithm is run for each of the objectives ignoring all the other objectives but keeping the constraints. Once each objective is optimized separately, putting all the optimum objective values together provides us the ideal point in the objective space.

7. Decision Making Using the Visual Interactive Method

The Visual Interactive Method for Decision Analysis (VIMDA) [18] has been used in the PI-EMO-PC algorithm for choosing the best point from the archive set. VIMDA provides a simple method for decision makers to examine efficient alternatives with no restrictive assumptions about the underlying value function of the decision maker. The method utilizes the reference direction approach [14] to search for the best solution from a given set.

The procedure is implemented by accepting information about aspiration levels of the decision maker for each of the criteria. A reference direction is generated from the preferred point in the current iteration of VIMDA to the aspiration level set by the decision maker. The reference direction is projected on the set of solutions in the archive by using Wierzbicki's achievement function. A few solutions are chosen based on the attainment values and the subset is visually presented to the decision maker for consideration. The solutions are ranked by their attainment values obtained using the achievement function. The visual tool allows the decision maker to perform a comparison of the presented alternatives. After weighing the tradeoffs, the decision maker chooses the preferred alternative and is then provided with an opportunity to modify the aspiration levels. If the decision maker modifies the aspiration levels, the procedure proceeds to the next iteration and a new set of solutions is presented; otherwise the procedure is terminated and the last chosen alternative is taken as the best solution from the archive.

8. Parameter Setting

For all the simulation runs done in this study we have used the following parameter values. The results obtained with this setting have been presented in the next section.

1. Number of decision maker calls: $T_{DM} = 10$ and 20 for each of the test problems.
2. Crossover probability and the distribution index for the SBX operator: $p_c = 0.9$ and $\eta_c = 15$.
3. Mutation probability and the distribution index for polynomial mutation: $p_m = 0.1$ and $\eta_m = 20$.
4. Population size: $N = 10M$, where M is the number of objectives.
5. Maximum Archive Size: $A^{max} = 10N$, where N is the population size.

9. Results

In this section we present the results of the PI-EMO-PC procedure on two, three, and five objective unconstrained as well as constrained test problems. ZDT1, DTLZ2, DTLZ8 and DTLZ9 test problems are adapted to create maximization problems.

After presenting the results in this section, we perform a parametric study with T_{DM} and $|A|^{max}$ in the next section.

9.1. Two-Objective Unconstrained Test Problem

Problem 1 is a modified formulation of ZDT1 test problem and has 30 variables.

$$\begin{aligned} \text{Maximize} \quad & \mathbf{f}(\mathbf{x}) = \left\{ \begin{array}{c} x_1 \\ \frac{10-\sqrt{x_1 g(\mathbf{X})}}{g(\mathbf{X})} \end{array} \right\}, \\ \text{where} \quad & g(\mathbf{x}) = 1 + \tfrac{9}{29} \textstyle\sum_{i=2}^{30} x_i, \\ & 0 \le x_i \le 1, \quad \text{for } i = 1, 2, \ldots, 30, \end{aligned} \tag{8}$$

The Pareto-optimal front of this problem is given by the curve $f_2 = 10 - \sqrt{f_1}$ and is shown in Figure 10. The values of the decision variables for solutions corresponding to the Pareto-optimal front are, $\mathbf{x}_i = 0$ for $i = 2, 3, \ldots, 30$ and $x_1 \in [0, 1]$.

This problem is modified to have a non-convex Pareto front. This test problem assesses the algorithm's ability in handling problems with non-convex frontier. Algorithms relying only on linear value functions will fail to solve such a problem. In our simulations, we assume a particular value function which acts as a representative of the DM, but any information from the value function is not used in the algorithm other than while deciding the best solution from the decision maker. The most preferred point $\mathbf{z}^*$ can be pre-determined from the chosen value function, which allows us to compare the solution obtained from PI-EMO-PC with $\mathbf{z}^*$.

We use the following value function which emulates the DM. The value function is non linear and is only used in finding the best point from the archive:

$$V(f_1, f_2) = \frac{1}{(f_1 - 0.35)^2 + (f_2 - 9.6)^2}. \tag{9}$$

The contours of this value function are shown in Figure 10 along with the most preferred point ($\mathbf{z}^* = (0.25, 9.50)$) corresponding to the value function.

Table 1 contains the solutions obtained from multiple runs of PI-EMO-PC procedure on the modified ZDT1 test problem. MPP denotes the exact most preferred point corresponding to the value function emulating the DM. The best, median and worst optimized points obtained from 21 different runs of the algorithm are presented in the table. The optimized point obtained from a single run of the PI-EMO-PC procedure is evaluated based on its distance from the exact most preferred point. The closer the point to the most preferred point, the better it is. Results are presented for two different budget of DM calls, i.e. $T_{DM} = 10$ and 20.

Table 2 shows best, median and worst accuracy and the number of overall function evaluations recorded from 21 runs. The accuracy measure shown in the

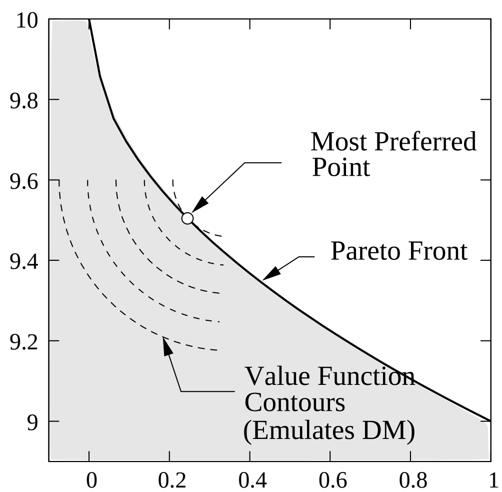

Figure 10: Contours of the chosen value function (acts as a DM) and the most preferred point corresponding to the value function. The figure is taken from [4].

Table 1: Final solutions obtained by PI-EMO-PC for the modified ZDT1 problem.

		$T_{DM} = 10$			$T_{DM} = 20$		
	MPP	Best	Median	Worst	Best	Median	Worst
f_1	0.2500	0.2505	0.2646	0.1164	0.2495	0.2412	0.2742
f_2	9.5000	9.4995	9.4856	9.6589	9.5005	9.5088	9.4764

table is computed based on the distance of the optimized point from the exact most preferred point ($\mathbf{z}^*$). Results have been presented for two different budget of DM calls, i.e. $T_{DM} = 10$ and 20. As expected, when the budget of calls is increased from 10 to 20, the accuracy increases. It should be noted that the accuracy and the number of function evaluations given in the table may not correspond to the same run. The accuracy can be best for one of the 21 runs and the number of function evaluations may be best for some other run. The table indicates that the proposed PI-EMO-PC procedure is able to find a solution close to the exact most preferred solution.

9.2. Three-Objective Unconstrained Test Problem

The DTLZ2 test problem [16] is a scalable test problem where all points in the objective space are bounded by two spherical surfaces in the first octant. In this

Table 2: Distance of obtained solution from the most preferred solution, function evaluations, and the number of DM calls required by the PI-EMO-PC for the modified ZDT1 problem.

	$T_{DM} = 10$			$T_{DM} = 20$		
	Best	Median	Worst	Best	Median	Worst
Accuracy	0.0008	0.0205	0.2076	0.0007	0.0125	0.0338
# of Function Evals.	5681	6472	7619	6055	7152	8302

paper, we maximize each of the objectives of the DTLZ2 test problem, therefore the outer spherical surface becomes the Pareto-optimal front. A modified M-objective DTLZ2 problem for maximization is defined as follows:

$$\begin{array}{rl} \text{Maximize} & \mathbf{f}(\mathbf{x}) = \\ & \left\{ \begin{array}{l} (1.0+g(\mathbf{x}))\cos(\frac{\pi}{2}x_1)\cos(\frac{\pi}{2}x_2)\cdots\cos(\frac{\pi}{2}x_{M-1}) \\ (1.0+g(\mathbf{x}))\cos(\frac{\pi}{2}x_1)\cos(\frac{\pi}{2}x_2)\cdots\sin(\frac{\pi}{2}x_{M-1}) \\ \vdots \\ (1.0+g(\mathbf{x}))\cos(\frac{\pi}{2}x_1)\sin(\frac{\pi}{2}x_2) \\ (1.0+g(\mathbf{x}))\sin(\frac{\pi}{2}x_1) \end{array} \right\}, \\ \text{subject to} & 0 \le x_i \le 1, \quad \text{for } i = 1, \ldots, 12, \\ \text{where} & g(\mathbf{x}) = \sum_{i=3}^{12}(x_i - 0.5)^2. \end{array} \tag{10}$$

The Pareto-optimal front for the maximization problem defined above is shown in Figure 12. The points in the objective space corresponding to the Pareto-optimal front follow the relation: $f_1^2 + f_2^2 + f_3^2 = 3.5^2$. The decision variables corresponding to the Pareto-optimal front are $x_1 \in [0, 1]$, $x_2 \in [0, 1]$ and $x_i = 0$ or 1 for $i = 3, 4, \ldots, 12$.

As in the previous example, we choose a value function emulating the decision maker. For this test problem we assume that the decision maker follows a linear value function which is defined as:

$$V(f_1, f_2, f_3) = 1.25f_1 + 1.50f_2 + 2.9047f_3. \tag{11}$$

The most preferred point for this value function is: $\mathbf{z}^* = (1.25, 1.50, 2.9047)$.

The PI-EMO-PC is executed 21 times with different initial population each time. A population size of $N = 30$ is used in the simulations. As in the previous test problem, we present the optimized points obtained from best, median and worst performing runs in Table 3. In Table 4, accuracy and number of overall

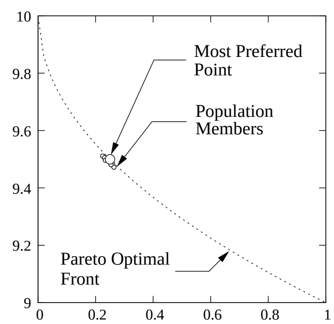

Figure 11: Final population members after termination of the algorithm for two-objective modified ZDT1 problem.

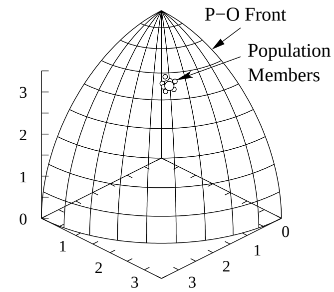

Figure 12: Final population members after termination of the algorithm for three-objective modified DTLZ2 problem.

function evaluations needed by the algorithm are given. Once again the results have been presented for two different values of the budget (T_{DM}) of DM calls. The accuracy achieved suggests that the obtained points are in proximity to the most preferred point $\mathbf{z}^*$. Figure 12 shows the population at the final generation of a typical PI-EMO-PC run.

Table 3: Final solutions obtained by PI-EMO-PC for the three-objective modified DTLZ2 problem.

		$T_{DM} = 10$			$T_{DM} = 20$		
	MPP	Best	Median	Worst	Best	Median	Worst
f_1	1.2500	1.2499	1.3076	0.8277	1.2512	1.2545	1.2108
f_2	1.5000	1.5037	1.5569	0.9600	1.5018	1.5682	1.3443
f_3	2.9047	2.9029	2.8489	3.2624	2.9033	2.8666	2.9961

9.3. Three-Objective Constrained Test Problem

Now we consider a three objective constrained test problem. The chosen test problem is DTLZ8 [16] which is scalable to any number of objectives. As suggested in [16] we choose the number of variables as 30 ($n = 10M$) for three objectives. The original problem is a minimization problem, but we wish convert it into a maximization problem. Therefore, a negative sign has been used before both objectives. The test problem for M number of objectives and n number of

Table 4: Distance of obtained solution from the most preferred solution, number of function evaluations, and number of DM calls required by PI-EMO-PC on the three-objective modified DTLZ2 problem.

	$T_{DM} = 10$			$T_{DM} = 20$		
	Best	Median	Worst	Best	Median	Worst
Accuracy	0.0041	0.0984	0.7732	0.0026	0.0782	0.1847
# of Function Evals.	3752	4253	4891	5451	6252	7444

variables is described below:

$$\begin{aligned}
\text{Maximize} \quad & f_j(\mathbf{x}) = -\frac{1}{\lfloor \frac{n}{M} \rfloor} \sum_{i=\lfloor (j-1)\frac{n}{M} \rfloor}^{\lfloor j\frac{n}{M} \rfloor} x_i, \\
& \quad j = 1, 2, \ldots, M, \\
\text{subject to} \quad & g_j(\mathbf{x}) = f_M(\mathbf{x}) + 4f_j(\mathbf{x}) - 1 \geq 0, \\
& \quad j = 1, 2, \ldots, M-1, \\
& g_M(\mathbf{x}) = 2f_M(\mathbf{x}) + \\
& \quad \min_{\substack{i, j = 1 \\ i \neq j}}^{M-1} [f_i(\mathbf{x}) + f_j(\mathbf{x})] - 1 \geq 0, \\
& 0 \leq x_i \leq 1, \quad \text{for } i = 1, \ldots, n,
\end{aligned} \tag{12}$$

The test problem defined above for M objectives has $M-1$ number of constraints. The Pareto-optimal front is formed by a straight line and a hyper plane. Intersection of the first $M-1$ constraints with $f_1 = f_2 = \ldots = f_{M-1}$ gives the straight line and the hyper-plane is given by g_M. Figure 13 shows the Pareto-optimal front for a three-objective DTLZ8 problem.

For this test problem we choose the following non-linear value function emulating the decision maker:

$$V(\mathbf{f}) = 1 / \sum_{i=1}^{3} (f_i - a_i)^2, \tag{13}$$

where $\mathbf{a} = (0.0, 0.0, -0.775)^T$. This most preferred point corresponding to this value function is: $\mathbf{z}^* = (-0.05, -0.05, -0.80)$.

The PI-EMO-PC is executed 21 number of times with random initial population and a population size of $N = 30$ is used for each run. Table 5 contains the solutions from the best, median and worst performing runs. Table 6 contains

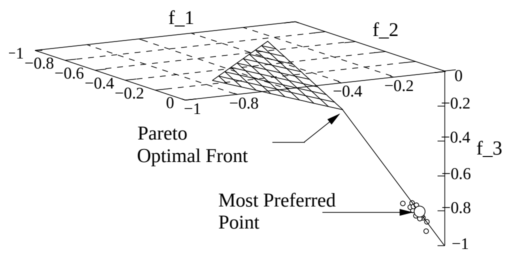

Figure 13: Final population members after termination of the algorithm for three-objective modified DTLZ8 problem.

the accuracy and number of overall function evaluations needed by the procedure. Results are presented for two different values of the budget (T_{DM}) of DM calls.

Table 5: Final solutions obtained by PI-EMO-PC for the three-objective DTLZ8 problem.

		$T_{DM} = 10$			$T_{DM} = 20$		
	MPP	Best	Median	Worst	Best	Median	Worst
f_1	0.0500	0.0498	0.0451	0.0648	0.0500	0.0500	0.0501
f_2	0.0500	0.0499	0.0527	0.0284	0.0500	0.0505	0.0463
f_3	0.8000	0.8008	0.8195	0.7408	0.8000	0.8002	0.7996

9.4. Five-Objective Unconstrained Test Problem

Now we evaluate the performance of the algorithm on a five objective test problem. The chosen test problem is a five-objective version of the DTLZ2 problem described before. Once again the search space is defined by two spherical surfaces in the first octant, with the outer surface being the Pareto-optimal front. The relationship among the objective values on the Pareto-optimal front is $f_1^2 + f_2^2 + f_3^2 + f_4^2 + f_5^2 = 3.5^2$. For this test problem, we choose the following non-linear DM-emulated value function.

$$V(\mathbf{f}) = \frac{1}{\sum_{i=1}^{5}(f_i - a_i)^2} \tag{14}$$

$$\tag{15}$$

where $\mathbf{a} = (1.1, 1.21, 1.43, 1.76, 2.6468)$. The most preferred point corresponding to this value function is: $\mathbf{z}^* = (1.0, 1.1, 1.3, 1.6, 2.4062)$.

Table 6: Distance of obtained solution from the most preferred solution, number of function evaluations, and number of DM calls required by PI-EMO-PC on the three-objective DTLZ8 problem. $d_s = 0.01$

	$T_{DM} = 10$			$T_{DM} = 20$		
	Best	Median	Worst	Best	Median	Worst
Accuracy	0.0008	0.0203	0.0647	0.0000	0.0006	0.0037
# of Function Evals.	5258	6013	7336	8648	10543	12404

Table 7 contains the best, median and worst solutions obtained by PI-EMO-PC algorithm when executed 21 number of times with a population size of 50. Once again, we present the results for two different budgets of DM calls. In Table 8 best, median and worst results obtained for accuracy and number of overall function evaluations are presented for two different budgets of DM calls.

Table 7: Final objective values obtained from PI-EMO-PC for the five-objective modified DTLZ2 problem.

		$T_{DM} = 10$			$T_{DM} = 20$		
	MPP	Best	Median	Worst	Best	Median	Worst
f_1	1.0000	1.0010	1.0240	0.8193	0.9970	0.9628	1.3637
f_2	1.1000	1.1044	1.1580	0.4961	1.1014	1.1231	0.9592
f_3	1.3000	1.2983	1.2528	1.6622	1.3019	1.3282	1.1469
f_4	1.6000	1.6006	1.7173	0.8140	1.6007	1.6090	1.5373
f_5	2.4062	2.4044	2.3112	2.8120	2.4054	2.3892	2.4066

For this test problem we have shown a convergence plot in Figure 14 for one of the runs. The maximum budget of DM calls, T_{DM}, was fixed as 10 for the run. Since the most preferred point is already known for the test problem, we evaluate the convergence properties of the algorithm by calculating the Euclidean distance between the best point chosen by the decision maker (at a particular call) and the most preferred point. It can be seen from the plot that the algorithm improves uniformly from one DM call to another and converges towards the most preferred point. The uniform improvement from one DM call to another can be easily understood by referring to Figure 4. The initial algorithm run gains an

Table 8: Distance of obtained solution from the most preferred solution, function evaluations, and the number of DM calls required by PI-EMO-PC for the five-objective modified DTLZ2 problem.

	$T_{DM} = 10$			$T_{DM} = 20$		
	Best	Median	Worst	Best	Median	Worst
Accuracy	0.0052	0.1701	1.1449	0.0040	0.0555	0.4237
# of Function Evals.	6653	8504	9819	9255	11324	13212

improvement of d_I from one DM call to another and the fine-tuning run gains an improvement of $d_I/2$ from one DM call to another.

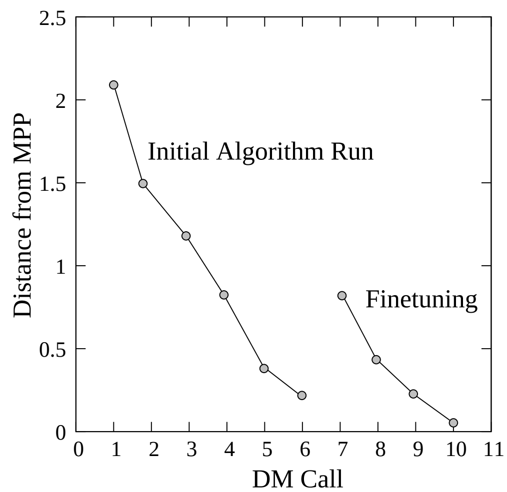

Figure 14: Convergence plot for the 5 objective DTLZ2 test problem. X-axis represents the DM call and the Y-axis represents the distance from the most preferred point.

9.5. Five-Objective Constrained Test Problem

We now consider the DTLZ9 [16] test problem which is scalable to any number of objectives. We consider a five objective ($M = 5$) version of the test problem with the number of variables equal to 50. Once again the original test problem is a minimization problem, but we wish to convert it into a maximization problem. Therefore, a negative sign has been used before each of the objectives to turn the problem into a maximization problem. The problem is defined as follows:

$$
\begin{aligned}
\text{Maximize} \quad & f_j(\mathbf{x}) = -\sum_{i=\lfloor (j-1)\frac{n}{M} \rfloor}^{\lfloor j\frac{n}{M} \rfloor} x_i^{0.1}, \quad j = 1, 2, \ldots, M, \\
\text{subject to} \quad & g_j(\mathbf{x}) = f_M^2(\mathbf{x}) + f_j^2(\mathbf{x}) - 1 \geq 0, \\
& \qquad j = 1, 2, \ldots, M-1, \\
& 0 \leq x_i \leq 1, \quad \text{for } i = 1, \ldots, n,
\end{aligned} \tag{16}
$$

The Pareto-optimal front, for this test problem, is given by $f_1 = f_2 = \ldots = f_{M-1}$. The Pareto-optimal curve lies on the intersection of all $M-1$ constraints. A two dimensional plot of the Pareto-optimal front with f_M and any other objective produces a circular arc of radius 1. The Pareto-optimal front for a five-objective DTLZ9 problem is shown in Figure 15 with objectives f_1 and f_5. The other objective values (f_2, f_3, f_4) are equal to f_1.

For this problem, we choose a DM emulating value function which is non-linear in nature, and defined as follows:

$$
V(\mathbf{f}) = 1/\sum_{i=1}^{5}(f_i - a_i)^2, \tag{17}
$$

where $\mathbf{a} = (-0.175, -0.175, -0.175, -0.175, -0.4899)^T$. The most preferred point for this value function is: $\mathbf{z}^* = (-0.2, -0.2, -0.2, -0.2, -0.9798)$.

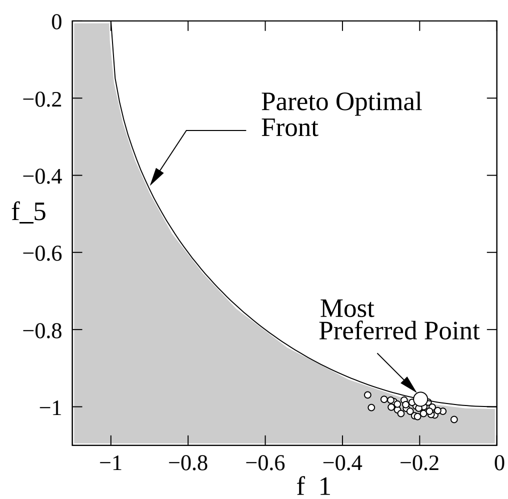

Figure 15: Final population members after termination of the algorithm for five-objective DTLZ9 problem.

Table 9 contains the best, median and worst obtained solutions from 21 different runs of the PI-EMO-PC procedure. Each run had a different initial population with a size of 50. Table 10 shows the accuracy and the number of overall function

Table 9: Final objective values obtained from PI-EMO-PC for the five-objective DTLZ9 problem.

		$T_{DM} = 10$			$T_{DM} = 20$		
	MPP	Best	Median	Worst	Best	Median	Worst
f_1	0.2000	0.2016	0.2186	0.2519	0.2001	0.2014	0.2110
f_2	0.2000	0.1988	0.2389	0.4054	0.1992	0.2161	0.3433
f_3	0.2000	0.1999	0.2011	0.2082	0.2002	0.2043	0.2068
f_4	0.2000	0.1998	0.1966	0.1677	0.2001	0.1951	0.1721
f_5	0.9798	0.9800	0.9805	0.9858	0.9800	0.9808	0.9851

evaluations required from different runs.

Table 10: Distance of obtained solution from the most preferred solution, function evaluations, and the number of DM calls required by PI-EMO-PC for the five-objective DTLZ9 problem. $d_s = 0.01$.

	$T_{DM} = 10$			$T_{DM} = 20$		
	Best	Median	Worst	Best	Median	Worst
Accuracy	0.0020	0.0433	0.2094	0.0008	0.0174	0.2024
# of Function Evals.	6296	7836	9290	9462	11468	13029

For all the test problems, the algorithm is able to produce the solution with a high accuracy. The number of function evaluations required for two and three objective are much lesser than what would be required if an EMO algorithm is used to generate the frontier with a similar accuracy. In five dimensions, an EMO algorithm (including NSGA-II) will fail [16] to produce a solution with a similar accuracy. Therefore, integration of preference information with the EMO algorithm is a viable way to handle high objective optimization problems. The preference information helps in a number of ways, like reducing the computational expense, leading to an enhanced accuracy and rendering high objective problems solvable.

10. Parametric Study

The algorithm uses two important parameters, T_{DM}, which is the maximum number of times a decision maker will be available to provide preference information, and $|A|^{max}$, which is the maximum archive size. The other parameters used in the algorithm are the usual parameters associated with an evolutionary algorithm, namely, population size, crossover probability, mutation probability, crossover index and mutation index. A parametric study has been done using T_{DM} and $|A|^{max}$ while the other usual EMO parameters have been kept fixed. Results for $T_{DM} = 10$ and $T_{DM} = 20$ have already been presented, but in this section the algorithm is evaluated for higher values of T_{DM}. Effect of maintaining a small or a large archive has also been studied and the results are reported.

10.1. Effect of DM Calls (T_{DM})

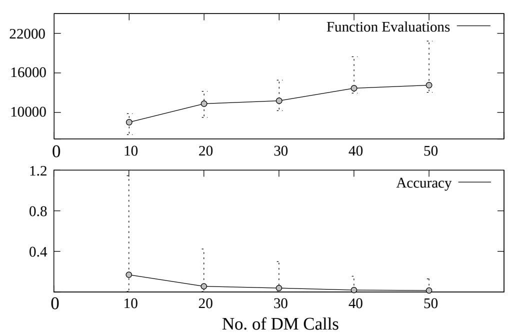

Figure 16: Performance measures on five-objective modified DTLZ2 problem for different T_{DM} values.

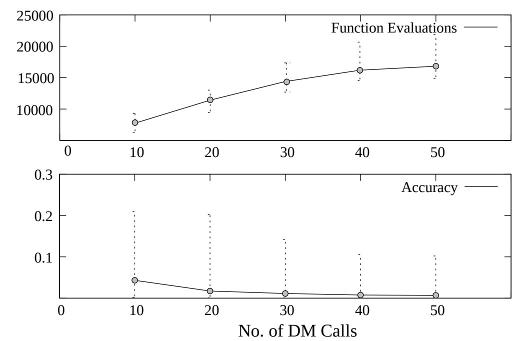

Figure 17: Performance measures on five-objective modified DTLZ9 problem for different T_{DM} values.

The effect of T_{DM} has been studied by considering five different values: 10, 20, 30, 40 and 50. The parameter $|A|^{max}$ has been kept fixed at $10N$. The PI-EMO-PC procedure has been run 21 times with different initial random populations and the best, median and worst performance measures have been presented.

Figures 16 and 17 show the accuracy and the number of function evaluations when T_{DM} is increased from 10 to 50 for five objective modified DTLZ2 and DTLZ9 test problems; respectively.

The figures represent the Euclidean distance of the obtained point through PI-EMO-PC algorithm from the most preferred point. A decrease in Euclidean distance means a higher accuracy which has been plotted. It can be observed from the two figures that with increase in preference information in terms of number of DM calls leads to improvement in accuracy. The function evaluations are found to increase with increasing DM calls as the algorithm moves in a zig-zag path with

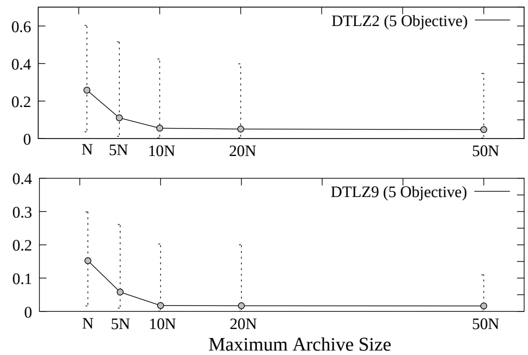

Figure 18: Accuracy (distance from MPP) against maximum archive size, $|A|^{max}$, for five objective modified DTLZ2 and DTLZ9 test problems.

small steps towards the most preferred point, whereas in case of low DM calls the algorithm still moves in a zig-zag path but in larger steps.

Figure 18 shows the change in accuracy with increase in the maximum archive. As the archive size increases there is a sharp increase in accuracy as well. Once a good accuracy has been achieved there is not much improvement with increase in archive size, though the precision of the algorithm improves as the maximum archive size increases.

11. Comparison with Another PI-EMO Technique

In this section we compare the algorithm suggested in this paper with the Progressively Interactive EMO which uses value functions (PI-EMO-VF) to converge towards the most preferred point. The PI-EMO-VF algorithm as suggested in [4, 5] provides the decision maker with $\eta = 5$ solutions in the objective space and expects a complete or partial ordering of the points. Elicitation of the order of the points is termed as a single DM call. In case of the PI-EMO-PC algorithm a DM call refers to the selection of the best solution in the objective space from a provided set of solutions in the archive set. The decision maker usually uses a decision tool to figure out the best solution from the archive. Elicitation of preference information is, therefore, entirely different for the two algorithms, and it is not a wise idea to directly compare the two ways in which preference information is elicited from the decision maker. Depending on the problem as well as the decision maker, ordering a small set of points might be easier when compared to finding the best from a much bigger set or vice versa. In this section, the aim is to compare the accuracy of the final solution obtained using the same number of DM calls for the two algorithms.

In the PI-EMO-VF algorithm it is not possible to fix the number of DM calls. Therefore, in order to make a comparison, the PI-EMO-VF algorithm is executed for a particular test problem 21 number of times. The median of the required number of DM calls is noted along with the final obtained solution. The median of the number of DM calls is fed as the maximum budget to the PI-EMO-PC algorithm for the same test problem. The algorithm is executed and the final solution obtained by the the PI-EMO-PC algorithm is compared against the final solution produced by PI-EMO-VF. The comparison is made in terms of accuracy which is the Euclidean distance from the most preferred point and the number of function evaluations. Same DM emulated value function is used with both algorithms to provide preference information during the intermediate generations. A comparison has been done on two objective ZDT1, three objective DTLZ2 and five objective DTLZ2 test problems.

The comparison results have been reported in Figures 19, 20 and 21. From the results it is found that the PI-EMO-PC algorithm performs slightly better than the PI-EMO-VF algorithm. There are more number of instances in the figure where the accuracy is better and the function evaluations is less for the PI-EMO-PC algorithm. Though, there are also cases where PI-EMO-VF performs better in terms of accuracy or function evaluations or both. For example, in case of the ZDT1 test problem PI-EMO-VF is found to provide a better accuracy with lesser number of function evaluations for $d_s = 0.01$ [6]. The reason for PI-EMO-PC performing slightly better than PI-EMO-VF can be attributed to judiciously chosen step size between two decision maker calls and the fine-tuning incorporated in the algorithm. Better performance of PI-EMO-PC over PI-EMO-VF is clear in the three objective and five objective test problem.

Few of the advantages of the PI-EMO-PC procedure compared with the PI-EMO-VF procedure are as follows:

1: The PI-EMO-PC procedure does not require the construction of a value function. While using the PI-EMO-VF procedure a value function has to be constructed after every few generations, and for this, a value function form (Polynomial, Cobb-Douglas, CES etc) has to be assumed. Construction of a polyhedral cone does not require us to assume any such form. Moreover,

[6] It is a stopping parameter in the PI-EMO-VF algorithm. A low value ensures a better accuracy and requires a higher number of decision maker calls

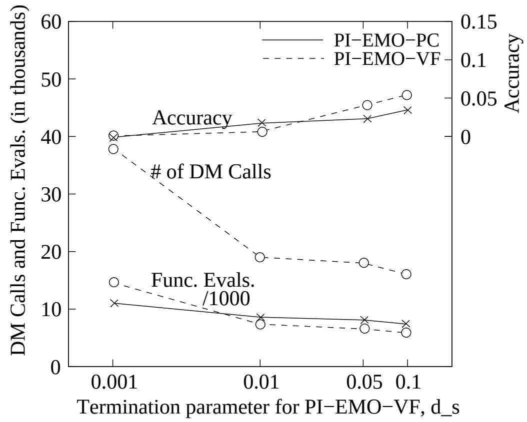

Figure 19: Comparison of PI-EMO-VF and PI-EMO-PC on ZDT1 test problem, $|A|^{max} = 10N$

during the construction of a value function, error while providing the rank-order information may lead to a significantly different value function. The PI-EMO-PC method requires to pick up the best solution from the archive set; if a DM makes an error while picking up the best solution and ends up picking up a solution close to the best, the constructed polyhedral cone will not change significantly. Therefore the PI-EMO-PC method is more robust towards errors.

2: For higher objectives, rank-ordering of solutions might be difficult. The PI-EMO-PC method uses a visually interactive decision tool which makes it easier for a decision maker to pick up the best point from a given set.

3: For the PI-EMO-VF method, a fixed number of points are given to the decision maker for rank-ordering. However, with increasing number of objectives, the number of points given to the decision maker for ordering should increase. There is no prescribed rule as to how many points be given to the decision maker for rank-ordering.

4: From the results obtained, it can be seen that with increasing number of objectives, the performance of PI-EMO-PC gets better than PI-EMO-VF. The reason for this is: the number of points given to a decision maker for rank-ordering has been kept fixed in the PI-EMO-VF code which leads to poorly constructed value functions for higher objectives.

5: The PI-EMO-PC approach has a fine-tuning procedure inbuilt which is not present in the PI-EMO-VF approach.

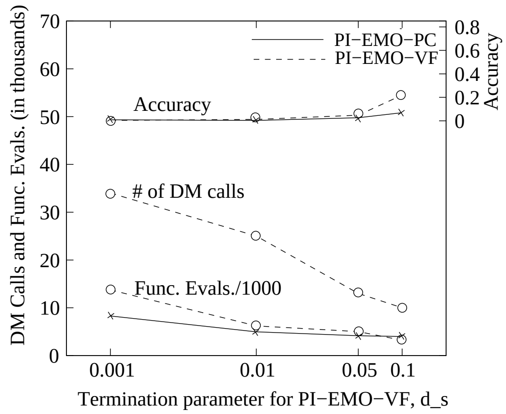

Figure 20: Comparison of PI-EMO-VF and PI-EMO-PC on 3-objective DTLZ2 test problem, $|A|^{max} = 10N$

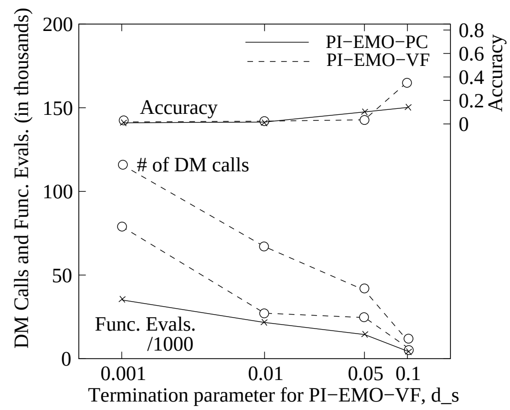

Figure 21: Comparison of PI-EMO-VF and PI-EMO-PC on 5-objective DTLZ2 test problem, $|A|^{max} = 10N$

12. Future Studies

1. Once the final point has been produced on the front, a diverse set of solutions close to the most preferred point found by the algorithm can be produced using a mutation based search. This will provide the decision maker with a wide variety of solutions in the region of interest. Each point produced by mutation can be ensured to be a Pareto point solution by performing a single objective optimization of the achievement scalarizing function (ASF) [14] using the point created by mutation as the reference point and any direction with positive cosines as a reference direction.
2. D_I and d_I in the algorithm can be changed adaptively during the progress of the algorithm instead of keeping it fixed. This might help to integrate the fine-tuning step of the algorithm with the main algorithm and lead to an even better performance.
3. A real world problem with a real decision maker needs to be analyzed to evaluate the performance of the algorithm, and investigate the behavioral aspects and challenges involved while taking such decisions.

13. Conclusions

In this paper, a progressively interactive EMO algorithm has been proposed which uses a polyhedral cone and local search to move towards the most preferred point. Before the start of the optimization run the decision maker informs about the maximum number of times he/she will be available to provide preference information. The preference information is accepted in terms of choosing the best member from the archive set. Results presented on two to five objective constrained and unconstrained test problems suggest this approach to be a viable technique to handle high objective problems where the EMO algorithms tend to fail. The approach brings the decision maker and the algorithm together and provides more control to the decision maker over the optimization process. Progressive information elicitation provides the algorithm information to focus the search on a specific area of the search space.

The algorithm has been improved from an early version [6] in terms of reduction in parameters; the algorithm has been rendered self adaptive such that it makes a decision when to switch to a local search and when to elicit preference information from the decision maker. A parametric study has been done for the parameters, maximum number of DM calls and maximum archive size. It has been shown that these parameters are critical in defining the accuracy of the EMO run, with higher values leading to better accuracy. The performance and the efficacy of the algorithm show that such progressively interactive techniques are capable to handle high objective problems efficiently.

14. Acknowledgements

The authors wish to thank the Academy of Finland for their support of this study.

References

[1] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley, 2001.

[2] C. A. C. Coello, D. A. VanVeldhuizen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston, MA: Kluwer, 2002.

[3] A. Sinha, A. Srinivasan and K. Deb, "A Population-Based, Parent Centric Procedure for Constrained Real-Parameter Optimization," *IEEE Congress on Evolutionary Computation*, 2006.

[4] K. Deb, A. Sinha, P. Korhonen and J. Wallenius, "An Interactive Evolutionary Multi-Objective Optimization Method Based on Progressively Approximated Value Functions," *IEEE Transactions on Evolutionary Computation*, 2010, vol. 14, no. 5, pp. 723–739.

[5] A. Sinha, K. Deb, P. Korhonen and J. Wallenius, "Progressively Interactive Evolutionary Multi-Objective Optimization Method Using Generalized Polynomial Value Functions," *IEEE Congress on Evolutionary Computation*, 2010.

[6] A. Sinha, K. Deb, P. Korhonen and J. Wallenius, "An Interactive Evolutionary Multi-Objective Optimization Method Based on Polyhedral Cones," in *Proceedings of Learning and Intelligent Optimization Conference (LION-2010)*, 2010.

[7] J. Branke, S. Greco, R. Slowinski, and P. Zielniewicz, "Interactive evolutionary multiobjective optimization using robust ordinal regression," in *Proceedings of the Fifth International Conference on Evolutionary Multi-Criterion Optimization (EMO-09)*. Berlin: Springer-Verlag, 2009, pp. 554–568.

[8] P. Korhonen, H. Moskowitz, and J. Wallenius, "A progressive algorithm for modeling and solving multiple-criteria decision problems," *Operations Research*, vol. 34, no. 5, pp. 726–731, 1986.

[9] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[10] S. Phelps and M. Koksalan, "An interactive evolutionary metaheuristic for multiobjective combinatorial optimization," *Management Science*, vol. 49, no. 12, pp. 1726–1738, December 2003.

[11] A. Jaszkiewicz, "Interactive multiobjective optimization with the pareto memetic algorithm," *Foundations of Computing and Decision Sciences*, vol. 32, no. 1, pp. 15–32, 2007.

[12] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D.

Papailiou, and T. Fogarty, Eds. Athens, Greece: International Center for Numerical Methods in Engineering (CIMNE), 2001, pp. 95–100.

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

[14] A. P. Wierzbicki, “The use of reference objectives in multiobjective optimization,” in *Multiple Criteria Decision Making Theory and Applications*, G. Fandel and T. Gal, Eds. Berlin: Springer-Verlag, 1980, pp. 468–486.

[15] R. H. Byrd, J. Nocedal, and R. A. Waltz, *KNITRO: An integrated package for nonlinear optimization*. Springer-Verlag, 2006, pp. 35–59.

[16] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multi-objective optimization,” in *Evolutionary Multiobjective Optimization*, A. Abraham, L. Jain, and R. Goldberg, Eds. London: Springer-Verlag, 2005, pp. 105–145.

[17] J. W. Fowler, E. S. Gel, M. Koksalan, P. Korhonen, J. L. Marquis and J. Wallenius, “Interactive Evolutionary Multi-Objective Optimization for Quasi-Concave Preference Functions,” in *European Journal of Operational Research*, vol. 206 , no. 2 , pp. 417–425, 2009.

[18] P. Korhonen and J. Karaivanova “An Algorithm for Projecting a Reference Direction onto the Nondominated Set of Given Points,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 29, pp. 429–435, 1999.

[19] Y. Sawaragi, H. Nakayama, T. Tanino, “Theory of Multiobjective Optimization,” in *Academic Press, Orlando*, 1985.

[20] P.L. Yu, “Cone convexity, cone extreme points and nondominated solutions in decision problems with multiobjectives,” *Journal of Optimization Theory and Applications*, vol. 14, pp. 319–377, 1974.

[21] P.L. Yu, “Multiple-Criteria Decision Making: Concepts, Techniques, and Extensions,” *Plenum Press, New York*, 1985.

[22] V.D. Noghin, “Relative importance of criteria: A quantitative approach,” *Journal of Multi-criteria Decision Analysis*, vol. 6, pp. 355–363, 1997.

[23] M.M. Wiecek, “Advances in Cone-Based Preference Modeling for Decision Making with Multiple Criteria,” *Decision Making in Manufacturing and Services*, vol. 1, no. 1-2, pp. 153–173, 2007.