An Interactive Evolutionary Multi-objective Optimization Algorithm with a Limited Number of Decision Maker Calls

Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, Kalyanmoy Deb

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

Abstract

This paper presents a preference-based method to handle optimization problems with multiple objectives. With an increase in the number of objectives the computational cost in solving a multi-objective optimization problem rises exponentially, and it becomes increasingly difficult for evolutionary multi-objective techniques to produce the entire Pareto-optimal front. In this paper, an evolutionary multi-objective procedure is combined with preference information from the decision maker during the intermediate stages of the algorithm leading to the most preferred point. The proposed approach is different from the existing approaches, as it tries to find the most preferred point with a limited budget of decision maker calls. In this paper, we incorporate the idea into a progressively interactive technique based on polyhedral cones. The idea is also tested on another progressively interactive approach based on value functions. Results are provided on two to five-objective unconstrained as well as constrained test problems.

Keywords:
Evolutionary multi-objective optimization, multiple criteria decision-making, interactive multi-objective optimization.

1. Introduction

The evolutionary multi-objective optimization (EMO) algorithms have demonstrated their ability in solving complicated multiple objective problems [1, 2]. They have been successful in handling two to three objective test problems, but thereafter, the deterioration in performance becomes noticeable [3, 4, 5] both in terms of convergence\(^2\) and diversity\(^3\) [1]. The deterioration in performance while solving problems with larger number of objectives is primarily due to stagnation in

---

\(^1\)Also Deb@iitk.ac.in, Department of Mechanical Engineering, Indian Institute of Technology Kanpur, PIN 208016, Kanpur, India

\(^2\)In evolutionary multi-objective optimization, convergence refers to the proximity of the solutions to the Pareto-optimal frontier.

\(^3\)In evolutionary multi-objective optimization, diversity refers to the spread of solutions approximating the Pareto-optimal frontier.
search as the Pareto-dominance loses its discriminatory potential in higher dimensions. Moreover, the requirement of an exponentially increasing population size to explore the Pareto-optimal front leads to a huge computational expense. Difficulty in visualization of the objective space further leads to additional challenges related to performance evaluation of the algorithm as well as decision making. These difficulties are inherent to an optimization problem with a larger number of objectives, and efficient procedures are required. 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 multiple objectives.

The EMO algorithms aim for well spread solutions close to the Pareto-optimal front for two to three objective problems. The decision maker (DM) is expected to choose the most suitable point from an array of approximately Pareto-optimal points found by the EMO algorithm. However, in this paper we propose to integrate the DM with the optimization run of an EMO algorithm in a way such that the preferences of the DM can be incorporated into the intermediate generations of the algorithm. Such an integration leads to progress towards the most preferred point. 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 DM. Such a procedure, where a DM is involved in the intermediate generations of an EMO algorithm, is called a progressively interactive EMO approach (PI-EMO) [6, 9]. A progressively interactive approach is a DM-oriented approach, which allows the DM to guide the algorithm towards the most preferred point. The working of such an approach can be observed from Figure 1 for a two-objective maximization problem. The advantage associated with seeking the most preferred point, instead of the entire Pareto-optimal front, is that it saves us from the intricacies involved in exploring the entire multi-dimensional front.

This paper proposes a simple scheme, which could be integrated into any progressively interactive EMO algorithm to approach the most preferred point in limited number of interactions with the decision maker. The budget of DM calls is taken as input at the start of the optimization run, and then the decision maker is invited to provide preference statements whenever required. The concept is integrated with a progressively interactive EMO approach based on polyhedral cones (PI-EMO-PC) [10], which is elaborately discussed in the paper. The scheme of limited budget of DM calls is generic, and is also demonstrated on another progressively interactive EMO ap-
2. A Survey of Preference-based Evolutionary Methods

A multi-objective optimization problem inherently consists of two tasks, namely, search and decision making. These two tasks can be combined in various ways to generate procedures, which can be classified into three broad categories, i.e. apriori approach, aposteriori approach and interactive approach. In this section, we provide a review for the methods falling in each of these categories. Discussion about preference-based methods can also be found in the review papers by Rachmawati et al. [14] and Branke et al. [8].

In the apriori approach, preferences are elicited before the start of the algorithm; then the optimization task is executed by incorporating the preference information, and the most preferred solution is identified. Biased niching based EMO [21], the reference direction based EMO [24], reference point-based EMO approaches [22, 23], and the light beam approach based EMO [25] represent some of the efforts in the direction of utilizing preference information before the start of an EMO algorithm. Once the information is available in the form of a reference direction or reference point, the algorithm finds the most-preferred point without any further interaction with the DM. Deb et al. [22] used the concept of the reference point, but did not apply an Achievement Scalarizing Function; they rather used a weighted Euclidean distance to rank population members. In a later study, Thiele et al. [23] implemented a similar idea using the Achievement Scalarizing Function. Another simple algorithm based on this approach is to modify dominance based on ranks obtained for a few alternatives from the DM. Such an approach was used by Greenwood et al. [26] in their study. Tiwari et al. [32] used pre-determined preference cones in an evolutionary algorithm to converge to a part of the Pareto-optimal frontier. They considered two objective test problems in their study.

Information in an apriori approach is elicited towards the beginning, therefore, the solution obtained after executing the algorithm is usually not the best solution and may not even be close to the most preferred solution. The preference structure of the DM at the beginning might be different from the preference structure at the Pareto-optimal front. Therefore, the approach is highly error prone, as even slight deviations in providing preference information at the beginning may lead to entirely different solutions.

Most of the evolutionary multi-objective algorithms [17, 33] which aim to find the entire frontier, are classic examples of the aposteriori approach. In such methods, the decision making aspect is ignored and the entire Pareto-optimal frontier is generated before incorporating the DM. However, as already mentioned, there are enormous difficulties in finding the entire Pareto-optimal front for a problem having a large number of objectives. Choosing the most preferred solution from this front makes the problem even more challenging.

Realizing the various difficulties associated with the above two approaches, in recent years there has been interest towards development of interactive EMO algorithms, particularly for problems having a large number of objectives. A variety of interactive methods have been presented in the literature, but our focus is on progressively interactive techniques. As already mentioned,
progressively interactive methods converge towards a particular region of the frontier by incorporating preferences obtained from the DM in the dominated regions of the objective space. Preference elicitation is performed during the course of optimization such that a progress towards the most preferred point is made. Some of the recent work in the direction of progressively interactive techniques are [18, 19, 20, 9, 29, 6]. Next, we briefly highlight the salient features of these studies.

Phelps and Koksalan [18] periodically accept preferences from the DM and construct a linearly weighted sum of objectives, which is optimized in the subsequent generations using an evolutionary algorithm. Fowler et al. [19] send a few solutions to the DM for ranking, and construct a convex preference cone, which is used to rank the members not considered by the DM. The study assumes a quasi-concave preference structure for the DM, and presents the results on multi-dimensional knapsack problems. Jaszkiewicz [20] also uses linear value functions, however, his strategy is to select a set of compatible linear value functions from randomly generated linear value functions. The selected value functions are then used within the EMO algorithm to explore the preferred regions on the frontier. In fact, many algorithms use linear value functions, but they have limitations in handling problems where the most preferred point lies on a non-convex part of the Pareto-optimal front. Branke et al. [9] implemented the GRIP [27] methodology, where the preference information from the DM is used to construct all possible additive value functions conforming to the preferences. This guides the search of the EMO algorithm towards interesting regions of the frontier. They extended the idea in a later study [28]. In their approach, the DM is frequently invited to provide preferences, and the implementation has been done on two-objective test problems. It has not been demonstrated in the study, whether the method scales efficiently when handling higher number of objectives. A study by Koksalan et al. [29] defines a territory around preferred solutions elicited from the DM, and generates more solutions around them, obtaining a denser coverage of the interesting regions. Interaction with the DM is again performed, and based on this idea the algorithm gets close to the most preferred point. Some other studies, which utilize preference statements in an EMO interactively, are [30, 31]. Kaliszewski et al. [30] have developed a decision support scheme which could be used with an EMO. It operates by defining upper and lower bounds for solutions, and the process continues until the bounds are tight. Said et al. [31] use a variant of the Pareto-dominance criterion, which has the ability to create a strict partial order. This helps in guiding the search towards the preferred parts of the Pareto optimal frontier. The DM’s preferences are elicited as a set of aspiration levels.

In a recent study [6], the authors have proposed a progressively interactive EMO method, which uses implicitly defined value functions (PI-EMO-VF). In this approach the information from the DM is elicited and used to construct a polynomial value function. At different stages of the algorithm, the implicitly constructed value function maps the preference information provided by the DM, and is subsequently used for making decisions. Another recently suggested methodology [10] eliminates the process of constructing a value function by using polyhedral cones. This method uses preference information from the DM in the form of the best point from a given set. This information is used to formulate a polyhedral cone, which is used to modify the domination criterion, and drive the EMO procedure towards the region of interest.
3. PI-EMO Framework and Fixed Budget of DM Calls

Most of the progressively interactive methods follow a similar framework. In this section, we describe a general framework for such methods, and discuss what makes these methods different. Flowchart in Figure 2 shows the steps followed by a progressively interactive approach. For most of the progressively interactive methods, an initial set of solutions is generated either randomly or using evolutionary search with a standard domination criterion. Once an initial set of solutions is available, DM preferences are elicited on a selected set of solutions to construct a domination criterion, which helps an evolutionary algorithm perform a focussed search in the region of interest. The DM preferences are elicited alternately with the evolutionary search until the termination criterion is met. The algorithms usually produce a single most preferred point as the final outcome. Each step in the approach can be performed in different ways, for example, preference information can be accepted in terms of best point from a given set, or rank ordering a set of points, or pairwise comparisons of given pairs. Similarly, domination condition can be constructed in various ways for example in the context of this paper it may be a polyhedral-based domination criterion or a value function-based domination criterion. Other things which make algorithms different are, how long the evolutionary search is performed and when are the preferences elicited. Therefore, the existing approaches differ from each other in one or more of the following ways:

1. How are the preference statements elicited?
2. When are the preference statements elicited?
3. How are the preference statements converted into an effective domination criterion?

The main focus of this paper is to tackle point 2 from the above list in an effective manner, so that a limited number of DM calls can be efficiently utilized. Now, we provide a brief description how a fixed budget DM call scheme can be implemented for multi-objective problems, and then show its working on a specific progressively interactive approach in the next section.

The proposed fixed budget scheme requires the ideal point \( \bar{I} \) to start. The ideal point is determined by maximizing each of the objectives individually. An individual maximization of the objectives gives at most \( M \) points \( \bar{P}_1, \bar{P}_2, \ldots, \bar{P}_M \). A two-objective case has been shown in Figure 3 where maximization of the objectives gives 2 points \( \bar{P}_1 \) and \( \bar{P}_2 \). Once the ideal point is...
known, the initial random population is created. The point \((P_{ib})\) closest \(^6\) to the ideal point \((I)\) is chosen and its distance from the ideal point is denoted as \(D_I = \|I - P_{ib}\|\). \(\vec{D}_I = \vec{I} - \vec{P}_{ib}\) denotes the vector from \(P_{ib}\) to \(I\). This distance \(D_I\) is divided into \(d_I = \frac{D_I}{n_{DM}}\) equal parts (step-size). This is done to make sure that the preference information from the DM is elicited only after fixed progress has been made. The distance traversed is measured by projecting the progress of one of the points (discussed later) from the population on the vector \(\vec{D}_I\).

Thereafter, an evolutionary search is alternated with decision making instances, until the termination criterion is met. This simple mechanism ensures that the decision maker is invited only after a significant progress is made, and also guarantees that the search would terminate in less than or equal to the maximum allowed number of DM calls. It is also possible to implement the scheme by using a variable step-size instead of keeping it fixed. It might be interesting to study the performance of the scheme with larger step-sizes in the beginning and smaller step sizes in the end. In this paper, we perform the study with a fixed step-size, and then show the advantages of variable step-size towards the end.

When the above scheme is integrated with a standard EMO algorithm (such as NSGA-II \([17]\), SPEA-II \([33]\) and others), and the domination criterion is based on the preferences of the decision maker, the approach leads to the most preferred solution in a fixed budget of DM calls. In this paper, we incorporate the fixed budget scheme in a progressively interactive approach based on polyhedral cones. The approach approximates the preferred region of the decision maker by utilizing the preference statements of the decision maker without requiring any significant parameter to be set, apart from the standard parameters used in evolutionary algorithms. The steps involved in the algorithm are discussed in detail in the next section. Later, we also test the fixed budget scheme on the value function-based approach and compare the performance with the polyhedral cone-based approach.

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

In this section, we propose a progressively interactive EMO algorithm (PI-EMO-PC) \(^7\) which uses the notion of polyhedral cones and sets to converge towards the most preferred point of an \(M\) objective problem. Though the polyhedral cones and sets provide only an approximation of

---

\(^6\)Euclidean distance from the ideal point is computed in the objective space for each of the randomly generated points. The point with the shortest Euclidean distance is denoted as \(P_{ib}\).

\(^7\)For an earlier version, see \([10]\), where there is no mechanism to control the number of interactions with the DM.
the preferred region, they are sufficient to guide an EMO algorithm towards the most preferred point by eliciting preference information from the DM at various stages of the algorithm. Before discussing the step-wise procedure for the algorithm, we 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 \( A \in \mathbb{R}^{L \times M} \) is a matrix, then a polyhedral cone \( C(A) \subset \mathbb{R}^M \) determined by \( A \) is defined as:

\[
C(A) := \{ c \in \mathbb{R}^M : Ac \geq 0 \} \tag{1}
\]

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

\[
C(A, b) := \{ c \in \mathbb{R}^M : Ac \geq b \} \tag{2}
\]

A polyhedral cone is a special kind of cone which is defined by a solution set of a homogeneous system of linear inequalities. In \( M \) dimensions polyhedral cones may be constructed with any finite number of bounding hyperplanes. However, in the context of this paper we consider only 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.

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 population, we utilize it to create a polyhedral cone and set. Since we aim to produce a single point on the Pareto-front, close to the most preferred point, we require two important features in the preference-based approach. The first feature being identification of a proper search direction, and the second feature being identification of the preferred region. These two features are important for a preference-based approach to converge to one of the preferred points on the Pareto-optimal front. In the proposed approach, we use a polyhedral cone at each DM call, and choose a search direction based on the cone to perform a local search (LS). A preferred region is also determined for subsequent exploration by EMO.

Now, we discuss our PI-EMO-PC procedure in the form of a sequence of steps, integrated with the fixed budget scheme:

**Step 0:** Determine the ideal point \( \vec{I} \) by maximizing each of the objectives individually using any single-objective optimization algorithm.

**Step 1:** Initialize a population \( Par_0 \), and set population iteration counter \( t = 0 \). Initialize archive set \( ^8 A \). Determine the point closest \( (P^{ib}) \) to the ideal point \( \vec{I} \) and find the vector \( \vec{D_I} = \vec{I} - P^{ib} \). Store the distance \( d_I = \frac{D_I}{T_{DM}} \), where \( T_{DM} \) is the maximum number of DM calls.

---

\(^8\)The initial size of the archive \( (A) \) is 0. The maximum size the archive can have is \( |A|^{max} \), a subjectively determined parameter.
**Step 2:** Increment the counter as \( t \leftarrow t + 1 \) and execute the EMO algorithm generation with the usual Pareto-domination criterion. The best member \( P^{cb}_{\ast} \) in the current population is the one closest to the ideal point which dominates \( P^{ib}_{\ast} \). All the feasible non-dominated solutions found at the end of the generation are added to the archive \( A \).

**Step 3:** If projected distance of \( P^{cb}_{\ast} - P^{ib}_{\ast} \) on \( D_I \) 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 DM to choose the best solution \( A^{\ast \ast}_{c} \), from the archive \( A_{t} \) using VIMDA[37]. Here \( c \) refers to the DM call. Choose the end points\(^9\) from the non-dominated front of the current parent population \( P_{ar_{t}} \) as rest of the solutions. This makes the chosen number of points for polyhedral cone construction 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 4.3. Set the function evaluation counter \( f = 0 \).

**Step 5:** An offspring population, \( Off_{t} \), is created from the parent population, \( P_{ar_{t}} \), by executing an iteration of the EMO algorithm. Parent selection in the EMO algorithm is performed using modified domination criterion (discussed in Section 4.5), and then crossover-mutation operators are used to generate the offsprings. Increment the function evaluation counter \( f \) by number of function evaluations done in this step. Update the archive by including the offspring population into it and then removing the dominated solutions from the archive. If the archive size exceeds \( |A|^{\max} \), k-mean clustering is used to keep the diverse set of \( |A|^{\max} \) clusters; rest of the solutions are deleted.

**Step 6:** From parent population, \( P_{ar_{t}} \), and offspring population, \( Off_{t} \), a new population, \( P_{ar_{t+1}} \), is determined, using modified domination criterion 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 proceed to Step 7.

**Step 7:** Increment \( c \) by 1 and call the DM to choose the best solution \( A^{\ast \ast}_{c} \), from the archive \( A_{t} \). The previous best solution chosen by the DM is stored as \( A^{\ast \ast}_{c-1} \). The polyhedral cone is constructed and the search direction is determined. Projected distance of vector \( A^{\ast \ast}_{c} - A^{\ast \ast}_{c-1} \) on \( D_I \) is computed and stored as \( d_A \).

**Step 8:** If the projected distance \( d_A \) is less than \( d_I \), a local search is performed with \( A^{\ast \ast}_{c} \) as the reference point along the search direction. The best solution, \( A^{\ast \ast}_{c} \), chosen by the DM is updated at each iteration of the local search. The local search is stopped if \( |A^{\ast \ast}_{c} - A^{\ast \ast}_{c-1}| > d_I \), 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.

\(^9\)In a maximization problem, end points in a set of non-dominated solutions are those members, which have maximum value for one of the objectives. However, it may not be trivial. Refer Section 4.3 for further details.
The PI-EMO-PC algorithm requires the parameters $|A|^{\max}$ and $T_{DM}$ in addition to the EMO algorithm’s parameters to be specified.

4.2. Fine-tuning

While solving a problem it will often occur that not the entire budget of DM calls is utilized, i.e. \( c < T_{DM} \) at the end of the algorithm. In this case we recommend to move back to \( A_{\alpha}^{\text{best}} \), where \( \alpha = c - \left\lfloor \frac{T_{DM} - c}{2} \right\rfloor \), 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 are used to start the fine-tuning algorithm with a smaller \( d_I \) value.

The above fine-tuning procedure can also be utilized if the DM is unhappy with the final solution. In that case the DM specifies \( T'_{DM} \) and the algorithm moves back to \( A_{\alpha}^{\text{best}} \) where \( \alpha = c - \left\lfloor \frac{T'_{DM}}{2} \right\rfloor \) and \( d_I \) is updated as \( d_I = \frac{d_I}{2} \).

4.3. Polyhedral Cone

At an instance of a DM call, the DM is provided with the archive set from which the best point is chosen. The chosen point forms the apex of the polyhedral cone in \( M \)-dimensional space, and we require \( M \) additional points to construct the complete cone. The end points of the non-dominated part of the parent population at that instant are chosen as the remaining points for constructing the bounding hyperplanes of the polyhedral cone. In most of the cases, the number of end points will be equal to \( M \), and including the apex we will have \( M + 1 \) points for constructing the polyhedral cone. However, there could be a few anomalies, where the total number of points is not equal to \( M + 1 \); we handle such cases later in this sub-section.

Choosing a set of \( M - 1 \) points from the set of \( M \) end points and the apex gives us one of the bounding hyperplanes for the polyhedral cone. By considering all possible combinations, we are able to construct \( M \) different bounding hyperplanes for the polyhedral cone. All the chosen points are non-dominated with respect to each other, therefore if we choose any of the \( M \) bounding hyperplanes, then one of the two normals to that plane has non-negative components. This property is important when we decide the search direction for local search in the next sub-section.

For instance, Figure 4 and Figure 5 show the polyhedral cones in two and three dimensions. The equation of each hyperplane can be written as \( \mathbf{p}_i^\top \mathbf{f} + q_i = 0, i \in \{1, \ldots, M\} \). If a given point \( \mathbf{f}^{(0)} = (f_1^{(0)}, \ldots, f_M^{(0)}) \) in the objective space has \( \mathbf{p}_i^\top \mathbf{f}^{(0)} + q_i > 0 \ \forall \ i \in \{1, \ldots, M\} \), then the point lies inside the polyhedral cone; otherwise it lies outside. In Figure 4, the shaded region has \( \mathbf{p}_i^\top \mathbf{f} + q_i < 0 \) for at least one \( i \in \{1, \ldots, M\} \) and the unshaded region has \( \mathbf{p}_i^\top \mathbf{f} + q_i > 0 \ \forall \ i \in \{1, \ldots, M\} \). Similarly, in Figure 5, the unshaded polyhedral cone represents \( \mathbf{p}_i^\top \mathbf{f} + q_i > 0 \ \forall \ i \in \{1, \ldots, M\} \).

There could be a scenario, where we do not have \( M + 1 \) points to construct the polyhedral cone. Such a scenario could occur under two circumstances. Firstly, if the DM chooses an end point along a particular objective as the best point, and secondly, if there is an end point which is common along two or more objectives. To tackle such a situation, we propose the creation of artificial points in order to construct the polyhedral cone. If the DM chooses an end point (say \( (f_1^{(0)}, \ldots, f_i^{(0)}, \ldots, f_M^{(0)}) \)) along a particular objective \( i \) as the best point, then we create an artificial point by averaging the best points along other objectives.
point as \( f(0)_{1}, \ldots, f(0)_{i}, \ldots, f(0)_{M} \) \( - \) \( \delta_{1}, \ldots, \delta_{i}, \ldots, \delta_{M} \), with \( \delta_{i} = 0 \) and \( \delta_{j} \neq i = 10^{-5} \). The created artificial point is chosen as the new end point. If a single point happens to be the end point for two or more objectives, then for each of the objectives we create an artificial point as an end point using a similar strategy.

A different scenario could be that we have more than one end-point corresponding to a particular objective. This would rarely happen, but under such a situation one of those points is chosen randomly as the end point corresponding to that objective.

4.4. Local Search (LS)

Once the polyhedral cone is determined, it provides an idea for a search direction. The non-negative normal unit vectors \( \hat{V}_{i} \) of all the \( M \) hyperplanes can be summed up to get a search direction \( \hat{W} = \sum_{i=1}^{n} \hat{V}_{i} \). \( \hat{W} \) is the unit vector along \( \hat{W} \) and \( W_{i} \forall i \in \{1, \ldots, M\} \) represent the components of the vector \( \hat{W} \). Since all the unit vectors are non-negative, therefore the final search direction would not lead to deterioration in any of the objectives. 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 \([34]\) for the best point from the archive \( \hat{A}_{c}^{best} \). In the following equation, \( z^{b} = \hat{A}_{c}^{best} \):

\[
\text{Maximize } \left( \frac{M}{\min_{i=1}^{M} \frac{f_{i}(x) - z_{i}^{b}}{W_{i}}} \right) + \rho \sum_{j=1}^{M} \frac{f_{j}(x) - z_{j}^{b}}{W_{j}}.
\]

subject to \( x \in S \).

In the above formulation, \( S \) denotes the feasible decision variable space of the original problem. The second term has a small constant \( \rho (= 10^{-10} \) is suggested), which prevents the method from converging to a weakly Pareto-optimal point. The sequential quadratic programming (SQP) optimization method is used to solve the above problem and the intermediate solutions in the objective space \( (z^{(i)}, i = 1, 2, \ldots) \) are recorded. If at any intermediate point, \( \frac{(z^{(i)} - \hat{A}_{c}^{best})^{T} \cdot \hat{D}_{i}}{D_{i}} \) is larger than \( d_{i} \), the Achievement Scalarizing Function optimization is stopped and we continue with the EMO algorithm. In this case, we replace \( \hat{A}_{c}^{best} \) with \( z^{(i)} \) in the archive set, and update the archive set \( A_{t} \), by
deleting the dominated members. $A_{\text{best}}^t$ replaces the member closest to it in the parent population $P_{\text{ar}}$. Figure 6 depicts this scenario. On the other hand, if at the end of the SQP run, the final SQP solution (say, $z^T$) does not meet the criterion \( \frac{(Z^{(s)} - A_{\text{best}}^{s-1})_t}{D_t} > d_I \), we terminate the EMO algorithm and declare $z^T$ as the final preferred solution. This situation indicates that local search is unable to find 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. Figure 7 shows such a situation, warranting a termination of the PI-EMO procedure.

![Figure 6: Local search, when far away from the front, finds a point which meets the distance criterion. Hence, no termination of the PI-EMO-PC.](image)

![Figure 7: Local search does not find a point which meets the distance criterion. Hence, the PI-EMO-PC is terminated.](image)

4.5. Modified Domination Criterion

The polyhedral set, obtained as a result of the constructed polyhedral cone, has been used to modify the domination criterion in order to emphasize and create preferred solutions. The polyhedral set from the most recent decision-making interaction is given by $p_i^T f + q_i \geq 0 \forall i \in \{1, \ldots, M\}$. The bounding hyperplanes of the polyhedral set are used to eliminate regions, and focus the search entirely on the region preferred by the DM. The $i^{th}$ hyperplane of the polyhedral set divides the region into two half-spaces, $p_i^T f + q_i \geq 0$ and $p_i^T f + q_i < 0$. The region represented by $p_i^T f + q_i < 0$ is assumed to be less preferred over region represented by $p_i^T f + q_i \geq 0$. Therefore, the intersection of the regions represented by $p_i^T f + q_i \geq 0 \forall i \in \{1, \ldots, M\}$ gives the preferred region in which we perform a focused search after interaction with the DM. Based on this idea we develop a modified domination criterion in this sub-section.

Once the polyhedral set is known, any two feasible solutions, $x^{(1)}$ and $x^{(2)}$, are compared with their corresponding objective function values, $f^{(1)}$ and $f^{(2)}$, by using the following modified domination criterion:

1. If $p_i^T f^{(1)} + q_i \geq 0$ and $p_i^T f^{(2)} + q_i \geq 0 \forall i \in \{1, \ldots, M\}$, then the two points are compared based on the usual Pareto-domination principle.
2. If $p_i^T f^{(1)} + q_i < 0$ and $p_i^T f^{(2)} + q_i < 0$ for at least one $i \in \{1, \ldots, M\}$, then the two points are compared based on the usual Pareto-domination criterion.
3. If $p_i^T f^{(1)} + q_i \geq 0 \forall i \in \{1, \ldots, M\}$, and $p_i^T f^{(2)} + q_i < 0$ for at least one $i \in \{1, \ldots, M\}$, then the former dominates the latter.
Figure 8 illustrates the region dominated by two points $A$ and $B$. The set formed by the linear equations has been shown. Point $A$ lies in the region in which $p_i^T f + q_i < 0$ 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 Pareto-domination criterion. However, point $B$ lies in the region $p_i^T f + q_i > 0 \ \forall \ i \in \{1, \ldots, M\}$. For this point, the dominated region is different from that which would be obtained using the usual Pareto-domination criterion. In addition to the usual region of dominance, the dominated region includes all points which have $p_i^T f + q_i < 0$ for at least one $i \in \{1, \ldots, M\}$.

The constrained handling mechanism as defined in [17] is used. When two solutions under consideration for a domination check are feasible, then the above modified domination criterion 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.

5. 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 EMO algorithm. Firstly a few generations of the algorithm, as discussed in the steps above, are performed according to the usual NSGA-II algorithm [17] and the archive set is maintained. Thereafter, the NSGA-II algorithm is modified by using the modified domination criterion (discussed in Section 4.5) 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 because of its poor performance [11] in higher dimensions. It 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. Archiving makes sure that most of the non-dominated solutions that are generated at the intermediate steps of the algorithm are never lost, even if the DM makes an error while providing preference information at a particular DM call.
For local search (discussed in Section 4.4), the SQP code of KNITRO [35] software has been used in this paper to solve the single objective optimization problem. The SQP algorithm is terminated either when \( \frac{(z_i - A^i_c) \cdot \vec{D}_I}{D^I} > d_I \) or when the Karush-Kuhn-Tucker (KKT) error measure is less than or equal to \( 10^{-6} \). If the local search terminates due to the KKT measure, then the overall PI-EMO-PC algorithm gets terminated.

As already discussed, we need to find out the ideal point before the start of the algorithm. We use the parent centric crossover (PCX) based single-objective algorithm [15] for the single-objective optimization. The algorithm is executed for a maximum of 2000 function evaluations for each of the objectives ignoring all the other objectives, but keeping the constraints. These function evaluations are counted separately from the function evaluations required to solve the problem. While presenting the results later on, we assume that the ideal points are given. One could use any single objective optimization algorithm [12, 13] to find an approximation to the ideal point. If the number of function evaluations is very high, then the ideal point can be replaced with an aspiration point elicited from the DM.

6. 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. For all the runs in this study we have used the following parameter values.

1. Number of DM calls: \( T_{DM} = 10, 20 \).
2. Crossover Prob. = 0.9, Distribution Index (SBX operator) = 15.
3. Mutation Prob. = 0.1, Distribution Index (Polynomial Mutation) = 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.

After presenting the results obtained with this setting, we perform a parametric study with \( T_{DM} \) and \( |A|^{max} \) in a later section.

In this section, we consider 5 different test problems given in Table 1, which offer different kinds of difficulties to the algorithm. The test-set consists of unconstrained as well as constrained test problems, with number of objectives from 2 to 5. In order to evaluate our methodology, we resort to a value function emulating a consistent DM. A value function representing a consistent and rational DM has a fixed most preferred point, and optimizing this value function always leads to the same point on the frontier. No information from the value function is used in the algorithm other than deciding the best solution from the given set. Since the most preferred point \( z^* \) is known for the chosen value function, it allows us to compare the solution obtained from PI-EMO-PC with \( z^* \).

6.1. Two-Objective Unconstrained Test Problem

Problem 1 is a modified formulation of the ZDT1 test problem and has 30 variables. This problem is modified to have a non-convex Pareto front. This test problem assesses the algorithm’s ability in handling problems with a non-convex Pareto front.
The contours of the value function for this problem are shown in Figure 9 along with the most preferred point corresponding to the value function.

Table 2 shows best, median and worst accuracy and the number of overall function evaluations recorded from 21 runs. The overall function evaluations consist of the function evaluations used by the evolutionary approach, and the function evaluations used by the local search. The accuracy measure shown in Table 2 is computed based on Euclidean distance of the obtained solution from the exact most preferred point \( (z^*) \). 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 Table 2 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. Table 2 indicates that the proposed PI-EMO-PC procedure is able to find a solution close to the exact most preferred solution.

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.

<table>
<thead>
<tr>
<th></th>
<th>$T_{DM} = 10$</th>
<th></th>
<th>$T_{DM} = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Median</td>
<td>Worst</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.0008</td>
<td>0.0205</td>
<td>0.2076</td>
</tr>
<tr>
<td># of Function Evals.</td>
<td>5681</td>
<td>6472</td>
<td>7619</td>
</tr>
</tbody>
</table>

6.2. Three-Objective Unconstrained Test Problem

The DTLZ2 test problem [3] is a scalable test problem where all points in the objective space have positive coordinates and are bounded by two spherical surfaces. In this paper, we maximize each of the objectives of the DTLZ2 test problem, therefore the outer spherical surface becomes the Pareto-optimal front.

The PI-EMO-PC is executed 21 times with different initial population each time. In Table 3, the accuracy achieved and the number of overall function evaluations required 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 $z^*$. Figure 11 shows the population at the final generation of a typical PI-EMO-PC run.

6.3. Three-Objective Constrained Test Problem

Now, we consider a three-objective constrained test problem. The chosen test problem is DTLZ8 [3] which is scalable to any number of objectives. As suggested in [3], we choose the
Table 3: 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.

<table>
<thead>
<tr>
<th></th>
<th>( T_{DM} = 10 )</th>
<th></th>
<th>( T_{DM} = 20 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Median</td>
<td>Worst</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.0041</td>
<td>0.0984</td>
<td>0.7732</td>
</tr>
<tr>
<td># of Function Evals.</td>
<td>3752</td>
<td>4253</td>
<td>4891</td>
</tr>
</tbody>
</table>

Figure 11: Final population members after termination of the algorithm for three-objective modified DTLZ2 problem.

Figure 12: Final population members after termination of the algorithm for three-objective modified DTLZ8 problem.

number of variables as 30 \((n = 10M)\) for three objectives. The original problem is a minimization problem, but have converted it into a maximization problem.

The PI-EMO-PC is executed 21 times with random initial population and the results are presented in Table 4. Figure 12 shows the final population members at the final generation of a typical PI-EMO-PC run.

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 DTLZ8 problem.

<table>
<thead>
<tr>
<th></th>
<th>( T_{DM} = 10 )</th>
<th></th>
<th>( T_{DM} = 20 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Median</td>
<td>Worst</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.0008</td>
<td>0.0203</td>
<td>0.0647</td>
</tr>
<tr>
<td># of Function Evals.</td>
<td>5258</td>
<td>6013</td>
<td>7336</td>
</tr>
</tbody>
</table>

6.4. Five-Objective Unconstrained Test Problem

Next, 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, with the outer surface being the Pareto-optimal front. The results are presented in Table 5.

For this test problem we have shown a convergence plot in Figure 13 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
Table 5: 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.

<table>
<thead>
<tr>
<th></th>
<th>$T_{DM} = 10$</th>
<th></th>
<th>$T_{DM} = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Median</td>
<td>Worst</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.0052</td>
<td>0.1701</td>
<td>1.1449</td>
</tr>
<tr>
<td># of Function Evals.</td>
<td>6653</td>
<td>8504</td>
<td>9819</td>
</tr>
</tbody>
</table>

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 DM (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 gets close to the most preferred point. The algorithm run gains an approximate improvement of $d_I$ from one DM call to another, and the fine-tuning gains an approximate improvement of $d_I/2$ from one DM call to another.

![Figure 13: Convergence plot for the 5 objective DTLZ2 test problem. X-axis represents the number of DM calls and the Y-axis represents the distance from the most preferred point.](image)

![Figure 14: Final population members after termination of the algorithm for five-objective DTLZ9 problem.](image)

6.5. Five-Objective Constrained Test Problem

We now consider the DTLZ9 [3] 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 have converted it into a maximization problem.

Table 6 provides the results obtained from multiple runs of the PI-EMO-PC procedure. Figure 14 shows the final population members at the final generation of a typical PI-EMO-PC run. Since the first four objectives are expected to be equal on the frontier, we plot the figure with $f_1$ against $f_5$. The other objectives provide a similar plot when represented against $f_5$.

For all the test problems, the best, median and worst results suggest that the algorithm is able to approximate the most preferred point with a high accuracy. The obtained accuracy is found to
Table 6: 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.

<table>
<thead>
<tr>
<th></th>
<th>$T_{DM} = 10$</th>
<th></th>
<th>$T_{DM} = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Median</td>
<td>Worst</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.0020</td>
<td>0.0433</td>
<td>0.2094</td>
</tr>
<tr>
<td># of Function Evals.</td>
<td>6296</td>
<td>7836</td>
<td>9290</td>
</tr>
</tbody>
</table>

Figure 15: Box plot for accuracy distribution for $T_{DM} = 10, 20$ when PI-EMO-PC is executed on 100 random most preferred points for each test problem.

increase with an increase in the budget of decision maker calls.

6.6. Sensitivity to the Location of the Most Preferred Point

In order to test the sensitivity of the algorithm to the location of the most preferred point on different parts of the frontier, we randomly choose a most preferred point (say $z$) on the frontier and construct a value function as $V(f) = \sum_{i=1}^{M} \frac{1}{\sum_{i=1}^{M} (f_i - a_i)^2}$. For a given $z$, we choose $a = z + 0.1 \times \hat{p}$, where $\hat{p}$ is a unit vector perpendicular to the Pareto-optimal front at point $z$. The obtained value function is a distance-based function and not a true value function, as it might assign a higher value to a dominated point. However, the property of this function is that it assumes a maximum at the point $z$. If only non-dominated points are given for ranking, it serves the purpose of a value function and ranks them in the order of their proximity to the most preferred point. For each test problem we generate 100 random most preferred points, and their corresponding value functions. We execute the PI-EMO-PC algorithm for each of the most preferred points, and report the accuracy distribution in Figure 15. From the distribution we observe that the algorithm is able to converge close to the most preferred point, irrespective of the location of the most preferred point on the Pareto-optimal front.
7. Parametric Study

The algorithm uses two important parameters, $T_{DM}$, which is the maximum number of times a DM 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 $T_{DM}$. The effect of maintaining a small or a large archive has also been studied and the results are reported.

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 accuracy is represented as the Euclidean distance between the solution obtained through PI-EMO-PC algorithm and the most preferred point. The smaller the Euclidean distance, the higher is the accuracy. It can be observed from the two figures that an increase in preference information in terms of number of DM calls leads to improvement in accuracy. The function evaluations are found to be higher for higher DM calls, which can be attributed to two causes. Firstly,
higher number of local searches are performed at the intermediate steps of the algorithmic run leading to higher local search function evaluations. Secondly, the algorithm moves in a zig-zag path with small steps towards the most preferred point making the overall search path longer. 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 an increase in the maximum archive size. As the maximum archive size increases there is a sharp increase in accuracy as well. However, beyond the maximum archive size of $10^N$, there is no significant improvement with further increase in the maximum archive size.

8. Comparison with an EMO approach

The number of function evaluations required for two and three-objective problems is much less than what would be required if an EMO algorithm were used to generate the frontier with a similar accuracy. In five dimensions, an EMO algorithm (including NSGA-II) will fail [3] to produce a solution with a similar accuracy. In order to confirm this, we have performed a comparison of the proposed preference-based approach against an EMO algorithm (NSGA-II), which does not involve any DM interaction and uses the usual Pareto-domination criterion. The non-dominated solutions obtained by the EMO approach are taken and the distance from the point closest to the most preferred point is reported as the accuracy of the approach. Of course having a set of non-dominated solutions has its own advantages, however, here we would like to draw a comparison as to how close can an EMO algorithm get to the most preferred point. One could claim that increasing the population size of the algorithm could provide a better accuracy, but on the other hand it leads to a higher number of function evaluations. In this simulation, we have chosen different number of generations and population sizes for each problem. Based on a few experiments appropriate population sizes and number of generations were chosen to ensure that the EMO algorithm has converged. The comparison results are presented in Table 7. From Table 7, one could readily observe the deterioration in accuracy with increasing number of dimensions. Moreover, the accuracy in the case of NSGA-II has been achieved only after a high number of function evaluations. On the other hand, the proposed preference-based approach is able to achieve a higher accuracy with much fewer function evaluations for most of the problems. Most of the EMO algorithms require a high population size to generate the entire frontier, which leads to high function evaluations. However, PI-EMO-PC approach requires a much smaller population size, as it performs a focused search and generates a single point on the Pareto-optimal frontier. Therefore, integration of preference information within an EMO algorithm is a viable way to handle such problems. The preference information helps in a number of ways, such as reducing the computational expense, leading to an enhanced accuracy and rendering these problems solvable.

9. Implementation of Fixed Budget of DM Calls Scheme on PI-EMO-VF

In this section, we evaluate the proposed fixed budget of DM calls scheme on the PI-EMO-VF approach [6, 7]. The PI-EMO-VF approach is modified, such that it stores all the non-dominated members in the archive. A single point from the archive is chosen by the DM during any DM call, and a value function is constructed with this information. We implement the fixed budget
Table 7: Comparison of the PI-EMO-PC approach with $T_{DM} = 20$ against EMO approach (NSGA-II) with usual Pareto-domination criterion and no preference information. Both the approaches have been executed 21 times.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pop. Size</td>
<td>100</td>
<td>200</td>
<td>200</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>No. of Gens.</td>
<td>200</td>
<td>500</td>
<td>500</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Med. Fun. Evals.</td>
<td>20,000</td>
<td>100,000</td>
<td>100,000</td>
<td>500,000</td>
<td>500,000</td>
</tr>
<tr>
<td>Med. Accuracy</td>
<td>0.0083</td>
<td>0.1354</td>
<td>0.1125</td>
<td>0.5351</td>
<td>0.1554</td>
</tr>
<tr>
<td>PI-EMO-PC ($T_{DM} = 20$)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fun. Evals.</td>
<td>7152</td>
<td>6252</td>
<td>10,543</td>
<td>11,324</td>
<td>11,468</td>
</tr>
<tr>
<td>Med. Accuracy</td>
<td>0.0125</td>
<td>0.0782</td>
<td>0.0006</td>
<td>0.0555</td>
<td>0.0174</td>
</tr>
</tbody>
</table>

scheme, and also incorporate fine tuning in the PI-EMO-VF approach. In other words, we replace the polyhedral cones in the algorithm described in this paper with value functions, and evaluate the performance. The median results obtained by the value function based approach are reported in Table 8, and a comparison has been drawn against PI-EMO-PC. It can be observed that both methods perform similarly under similar settings.

Table 8: Comparison of the PI-EMO-PC against PI-EMO-VF under similar settings. Both the approaches have been executed 21 times. ($T_{DM} = 20$)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PI-EMO-VF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Med. Fun. Evals.</td>
<td>6322</td>
<td>5783</td>
<td>12,743</td>
<td>10,842</td>
<td>9,842</td>
</tr>
<tr>
<td>Med. Accuracy</td>
<td>0.0361</td>
<td>0.0526</td>
<td>0.0013</td>
<td>0.1426</td>
<td>0.0124</td>
</tr>
<tr>
<td>PI-EMO-PC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Med. Fun. Evals.</td>
<td>7152</td>
<td>6252</td>
<td>10,543</td>
<td>11,324</td>
<td>11,468</td>
</tr>
<tr>
<td>Med. Accuracy</td>
<td>0.0125</td>
<td>0.0782</td>
<td>0.0006</td>
<td>0.0555</td>
<td>0.0174</td>
</tr>
</tbody>
</table>

10. Variable Step-size Study on PI-EMO-PC

In this section, we perform a study for variable step-size, where we compare the fixed step-size against other possibilities, i.e. reducing step-size and increasing step-size. For reducing step-size we consider two scenarios, a linearly reducing step-size and an exponentially reducing step-size. Similarly, for increasing step-size we consider two scenarios, a linearly increasing step-size and an exponentially increasing step-size. If the estimated distance to be traversed is $D_I$, then the individual steps ($d_i$) taken by the algorithm should sum up to $D_I$, as shown in the following
$$\sum_{i=1}^{T_{DM}} d_i = D_I$$

In case of fixed step-size, $$\sum_{i=1}^{T_{DM}} d_i = D_I$$ which gives $$d_i = d_I = \frac{D_I}{T_{DM}}$$. In case of linearly reducing step-size we have $$\sum_{i=1}^{T_{DM}} (T_{DM} - i + 1)d_i = D_I$$, which gives $$d_i = (T_{DM} - i + 1)d_I = (T_{DM} - i + 1)\frac{2D_I}{T_{DM}(T_{DM}+1)}$$. Similarly, in case of exponentially reducing step size we have $$\sum_{i=1}^{T_{DM}} 2^{T_{DM}-i}d_i = D_I$$, which gives $$d_i = 2^{(T_{DM}-i)}d_I = 2^{(T_{DM}-i)}\frac{D_I}{\sum_{j=1}^{T_{DM}} 2^{T_{DM}-j}}$$. The step-sizes for the increasing case may be computed in a similar manner. Next, we provide the results in Table 9 for the variable step-size study performed on all the test problems. The table provides the median accuracy obtained for all the test problems. It can be observed that exponentially decreasing step-size performs the best followed by linearly decreasing step-size. Increasing step-size approach performs the worst, which could be understood intuitively. Decreasing step-size takes larger steps in the beginning and smaller steps towards the end, which means that the DM is called fewer times when the algorithm is far away from the frontier, and more frequent calls are made when the algorithm approaches the frontier. This makes it perform the best when compared with other cases. For brevity, we skip the table for number of function evaluations, as they were found to be of similar magnitude for all the step-size cases for the respective test problems.

Table 9: Median accuracy obtained for various step-size cases with $$T_{DM} = 10$$.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M-ZDT1 (2)</td>
<td>Fixed</td>
<td>0.0205</td>
<td>0.0179</td>
<td>0.0167</td>
<td>0.1247</td>
<td>0.1549</td>
</tr>
<tr>
<td>M-DTLZ2 (3)</td>
<td>Lin. Dec.</td>
<td>0.0984</td>
<td>0.0733</td>
<td>0.0647</td>
<td>0.2643</td>
<td>0.3366</td>
</tr>
<tr>
<td>M-DTLZ8 (3)</td>
<td>Exp. Dec.</td>
<td>0.0203</td>
<td>0.0176</td>
<td>0.0153</td>
<td>0.2983</td>
<td>0.3847</td>
</tr>
<tr>
<td>M-DTLZ2 (5)</td>
<td>Lin. Inc.</td>
<td>0.1701</td>
<td>0.1492</td>
<td>0.1344</td>
<td>0.4122</td>
<td>0.4478</td>
</tr>
<tr>
<td>M-DTLZ9 (5)</td>
<td>Exp. Inc.</td>
<td>0.0433</td>
<td>0.0378</td>
<td>0.0298</td>
<td>0.3311</td>
<td>0.4123</td>
</tr>
</tbody>
</table>

11. Estimating contributions of individual components

In this section, we aim to assess the importance of individual components in the algorithm, like the fine-tuning strategy, storing all solutions in an archive, and choice of fixed step-size. To assess the accuracy advantage because of a particular component, we execute the PI-EMO-PC approach by using all the components, and then compare the accuracy by eliminating one of the components. That is, in order to assess the importance of fine-tuning we execute the algorithm without fine-tuning and then record the accuracy loss. Similarly, in order to assess the importance of storage of solutions in an archive, we execute the algorithm without an archive and measure the accuracy loss. For assessing the importance of step-size, we execute the algorithm with a random step-size and once again record the loss in accuracy. For obtaining a random step-size,
we choose a step-size randomly between 0 and $2d_I$, such that the expected step-size is equal to $d_I$.

The accuracy loss has been presented in terms of percentage as follows:

$$\text{Accuracy Loss} = \frac{A_c - A_c^{(-1)}}{A_c} \times 100$$

In the above equation, $A_c$ represents the accuracy obtained from PI-EMO-PC executed with all the components, and $A_c^{(-1)}$ represents the accuracy of PI-EMO-PC approach without one particular component. The results have been presented in Table 10. It can be observed that step-size and storing solutions in an archive have a large contribution towards accuracy. Fine-tuning leads to a relatively small contribution towards accuracy.

Table 10: Accuracy loss attributed to absence of an individual component of PI-EMO-PC with $T_{DM} = 10$.

<table>
<thead>
<tr>
<th>Problems (Obj.)</th>
<th>Without Fine-tuning</th>
<th>Without Archive-storage</th>
<th>Without Step-size</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-ZDT1 (2)</td>
<td>15.35%</td>
<td>19.85%</td>
<td>17.64%</td>
</tr>
<tr>
<td>M-DTLZ2 (3)</td>
<td>8.74%</td>
<td>16.54%</td>
<td>14.63%</td>
</tr>
<tr>
<td>M-DTLZ8 (3)</td>
<td>9.65%</td>
<td>21.44%</td>
<td>16.24%</td>
</tr>
<tr>
<td>M-DTLZ2 (5)</td>
<td>6.34%</td>
<td>17.64%</td>
<td>19.27%</td>
</tr>
<tr>
<td>M-DTLZ9 (5)</td>
<td>8.12%</td>
<td>23.94%</td>
<td>14.45%</td>
</tr>
</tbody>
</table>

12. Additional Discussion

The proposed fixed budget scheme is generic and applicable to all kinds of multi-objective problems, irrespective of the kind of Pareto-frontier. The algorithm begins with an ideal point and computes the step-size between two DM calls based on the distance of the ideal point from the initial population members. The scheme assumes the ideal point as the target point and computes the step-sizes based on distance from it. In this way we are always over-estimating the distance of the actual most preferred point from the initial population members. If the Pareto-front is convex, the over-estimation might not be large, however, if the Pareto-front is concave, we might be highly overestimating the distance of the most preferred point. In cases with high overestimation, the algorithm will terminate with few DM calls and then the fine-tuning step will be used to utilize the remaining DM calls. In the current version of the algorithm, the fine-tuning is performed by retreating away from the frontier and then restarting the search with smaller step-sizes. In future work, we would like to improve the fine-tuning step by allowing a search on the Pareto-optimal frontier, instead of retreating away from the frontier. In the proposed approach, we fix the step-size between two DM calls, however, this might not always be a good strategy as suggested by our variable step-size study. The algorithm involves a local search step for which we are currently using an SQP-based algorithm. One could replace the SQP-based algorithm with any other fast black-box algorithm (for example; a greedy genetic algorithm).
13. Conclusions

In this paper, a progressively interactive EMO algorithm has been proposed which uses the idea of polyhedral cones and polyhedral sets to move towards the most preferred point. Before the start of the optimization the DM is expected to provide information about the maximum number of times she will be available to provide preference information. This is called the budget of DM calls, and the algorithm tries to find the most preferred point within this budget by accepting preference information in the intermediate steps of the algorithm. Preference information is accepted in terms of the best member from the given archive set. Eliciting preferences progressively provides the algorithm information to focus the search on a specific area of the search space. 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 DM. It brings the DM and the algorithm together, providing more control to the DM over the optimization process. The procedure successfully handles two to five-objective constrained and unconstrained test problems. A parametric study has been done for the parameters, maximum number of DM calls, and maximum archive size. It has been shown that the budget of DM calls is critical in defining the accuracy of the optimization, with higher values leading to better accuracy. The performance of the algorithm shows that such progressively interactive techniques are capable of handling optimization problems with a relatively large number of objectives efficiently.

14. Acknowledgments

The authors wish to thank the Academy of Finland (Grants: 133387 and 253583), Wallenberg Foundation and Liikesivistysrahasto for supporting this study.

References


