An Interactive Evolutionary Multi-Objective Optimization Method Based on Progressively Approximated Value Functions

Kalyanmoy Deb[*], Ankur Sinha[†], Pekka Korhonen[‡] and Jyrki Wallenius[§]

KanGAL Report Number 2009005

Abstract

This paper suggests a preference based methodology, which incorporates an evolutionary multi-objective optimization algorithm to lead a decision-maker to the most preferred solution of her or his choice. The progress towards the most preferred solution is made by accepting preference based information progressively from the decision maker after every few generations of an evolutionary multi-objective optimization algorithm. This preference information is used to model a strictly increasing value function, which is used for the subsequent iterations of the EMO algorithm. In addition to the development of the value function which satisfies DM's preference information, the proposed progressively interactive EMO (PI-EMO) approach utilizes the constructed value function in directing EMO algorithm's search to more preferred solutions. This is accomplished using a preference-based domination principle and utilizing a preference based termination criterion. Results on two to five-objective optimization problems using the progressively interactive NSGA-II approach shows the simplicity of the proposed approach and its future promise. A parametric study involving the algorithm's parameters reveals interesting insights of parameter interactions and indicates useful parameter values. A number of extensions to this study are also suggested.

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

1 Introduction

In evolutionary multi-objective optimization (EMO), the target has usually been to find a set of well-converged and well-diversified Pareto-optimal solutions [1, 2]. Once an optimization run is started, usually no further information is taken from the decision maker (DM). In an *aposteriori EMO approach*, after a set of approximate Pareto-optimal solutions has been found, a decision-making event is executed by taking preference information from a DM to choose the most preferred solution. As discussed elsewhere [3, 4], this is not a particularly good idea for handling a large number of objectives (practically, more than four). Firstly, the usual domination principle allows a majority of the population members to become non-dominated to each other, thereby not allowing much room for introducing new solutions in a finite

[*]Kalyanmoy Deb is a Deva Raj Chair Professor at Indian Institute of Technology Kanpur, PIN 208016, India (deb@iitk.ac.in), Also a Professor at Department of Business Technology, Helsinki School of Economics, PO Box 1210, FIN-00101, Helsinki, Finland (kalyanmoy.deb@hse.fi)

[†]Ankur Sinha is a doctoral student at the Department of Business Technology, Helsinki School of Economics, PO Box 1210, FIN-00101, Helsinki, Finland (ankur.sinha@hse.fi)

[‡]Pekka Korhonen is a Professor at the Department of Business Technology, Helsinki School of Economics, PO Box 1210, FIN-00101, Helsinki, Finland (pekka.korhonen@hse.fi)

[§]Jyrki Wallenius is a professor at the Department of Business Technology, Helsinki School of Economics, PO Box 1210, FIN-00101, Helsinki, Finland (jyrki.wallenius@hse.fi)

population. This slows down the progress of an EMO algorithm. Secondly, the representation of a high-dimensional Pareto-optimal front requires an exponentially large number of points, thereby requiring a large population size in running an EMO procedure. Thirdly, the visualization of a high-dimensional front becomes a non-trivial task for decision-making purposes.

To alleviate the above problems associated with the aposteriori EMO approach in handling a large number of objectives, some EMO researchers have adopted a particular multiple criteria decision-making (MCDM) approach (*apriori approach*) and attempted to find a crowded set of Pareto-optimal points near the most preferred solution. The cone-domination based EMO [5, 1], biased niching based EMO [6], reference point based EMO approaches [7, 8], the reference direction based EMO [9], the light beam approach based EMO [10] are a few attempts in this direction. Also, Greenwood et al. [18] derived a linear value function from a given ranking of a few alternatives and then employed an EMO algorithm to find points which are preferred with respect to the constructed linear value function. In Greenwood's method, the preference information is used prior to employing the EMO algorithm, thus this qualifies as another apriori method. For a recent survey, see [11]. These studies have clearly shown that it is difficult for an EMO algorithm alone to find a good spread of solutions in 5 or 10-objective problems. When solutions around a specific Pareto-optimal point (or around a region) are the target, MCDM-based EMO approaches suggested in these studies can find satisfactory solutions. However, in these approaches, the decision maker interacts only at the beginning of an EMO run. The decision maker provides preference information such as one or more reference point(s), one or more reference directions, one or more light beam specifics, etc. An EMO algorithm then targets its population to converge near the specific solutions on the Pareto-optimal front.

The above MCDM-based EMO approaches can also be used in an iterative manner with a DM, similar to the way suggested elsewhere [12, 13]. In an *semi-interactive EMO approach*, some preference information (in terms of reference points or reference directions or others) can be obtained from the DM and an MCDM-based EMO algorithm can be employed to find a set of preferred Pareto-optimal solutions. Thereafter, a few representative preferred solutions can be shown to the DM and a second set of preference information in terms of new reference points or new reference directions can be obtained and a second MCDM-based EMO run can be made. This procedure can be continued till a satisfactory solution is found. This principle has been utilized with the reference direction [9] and light beam approaches [10] to solve some engineering design problems.

However, the integration of preference information within an EMO algorithm can be made in a more effective manner, as shown in a recent study [14]. Instead of keeping the DM waiting, to complete an EMO run (either to find a complete Pareto-optimal front in the aposteriori approach or to find a preferred set of Pareto-optimal solutions based on an MCDM principle in an apriori approach), the DM can be involved to periodically provide preference information as the EMO iterations are underway. This will be a less time-consuming and simultaneously more flexible approach than the previously suggested ones. In such a *progressively interactive EMO approach* (PI-EMO), the DM is allowed to modify her/his preference structure as new solutions evolve. Since the DM gets more frequent chances to provide new information, the overall process is more DM-oriented. The DM may feel more in-charge and more involved in the overall optimization-cum-decision-making process.

In this paper, we have suggested a simplistic framework of a PI-EMO approach based on a couple of earlier progressive multi-criterion decision-making approaches [15, 16]. Periodically, the DM is supplied with a handful of currently non-dominated points and is asked to rank the points from best to worst. From hereon we refer to this instance as a 'DM call'. Based on this preference information, an optimization problem is formulated and solved to find a suitable value function, which optimally captures DM's preference information. From this iteration till the next DM call, the derived value function is utilized to drive the EMO algorithm in major ways: (i) in determining termination of the overall procedure and (ii) in modifying the domination principle, which directly affects EMO algorithm's convergence and diversity-preserving operators. The PI-EMO concept is integrated with the well-known NSGA-II algorithm [17] and the working of the algorithm is demonstrated on three problems involving two to five objectives. A parameter sensitivity study is also performed to analyze their influence on the working of the overall algorithm.

Thereafter, the sensitivity of the proposed PI-NSGA-II procedure on the inconsistencies in decision-maker responses is studied. Finally, a number of important and immediate future studies are listed and conclusions are drawn.

2 Past Studies on progressively interactive EMO

There exist a plethora of studies involving aposteriori and apriori EMO approaches. Most methodologies borrow the core decision-making idea from the MCDM literature and integrate it with an EMO algorithm. Since the focus of this study is not on discussing the aposteriori or the apriori EMO approaches, but to concentrate on procedures requiring more frequent involvements of a DM with an EMO algorithm, we do not provide a review of aposteriori and apriori approaches, except to encourage the readers to look at a recent survey [11].

Towards the methodologies involving a progressive use of preference information by involving a decision-maker in an evolutionary multi-objective optimization framework, there are not many studies yet. Some recent studies periodically presented to the DM one or more pairs of alternative points found by an EMO algorithm and expected the DM to provide some preference information about the points. The information is then used to derive a weighted value function, which is linear. Phelps and Köksalan [19] optimized the constructed linearly weighted sum of objectives in subsequent iterations using an evolutionary algorithm. In their technique, if the actual value function is non-linear, the method may not be able to find a linear approximation and may generate an infeasible solution. This creates a need to reformulate the optimization problem by deleting constraints one at a time. Fowler et al. [33] have developed an interactive EMO approach based on the idea of using convex preference cones. They use such cones to partially order the population members and further use the order as the fitness function. They have tested their algorithm on multi-dimensional (upto 4 dimensions) knapsack problems. Jaszkiewicz [20] selected a set of linear value functions (based on weighted sum of objectives) from a set of randomly created linear value functions, conforming to the preference information supplied by the DM by pairwise comparisons. EMO algorithm's search is then continued with these selective weight vectors. Although the assumption of linear value functions facilitates a quick and easy determination of the value function representing DM's preference information, linear value functions have limitations in handling non-linear problems, particularly where the most preferred point lies on a non-convex part of the Pareto-optimal front. Nevertheless, each interactive EMO idea suggested in the above-mentioned studies remains as the main hallmark of these studies.

Branke et al. [14] implemented the GRIP [21] methodology in which the DM compares pairs of alternatives and the preference information thus obtained is used to find all possible compatible additive value functions (not necessarily linear). An EMO algorithm (NSGA-II) then used a preference-based dominance relationship and a preference-based diversity preserving operator to find new solutions for the next few generations. Their procedure recommended to make a single pair of comparison after every few generations in order to develop the preference structure. Since this procedure generates not enough preference information after every call of the DM, the EMO algorithm is likely to keep a wide variety of points from across the Pareto-optimal front in the population. The authors have demonstrated their procedure on a two-objective test problem. To obtain a narrow range of points close to the true preferred Pareto-optimal point, they had to call the DM at every generation of the EMO algorithm. It is not clear how the procedure will perform in higher objective problems, where dominance-based approaches are too slow and a reasonably high level of preference information would be needed to make a fast and focused search using an EMO algorithm. However, the use of preference information in EMO algorithm's operations remains a significant contribution of this study.

Korhonen, Moskowitz and Wallenius [15] suggested a progressive and interactive multi-objective optimization and decision-making algorithm in which the DM is presented with a set of alternatives and is asked to make a set of binary comparisons of the alternatives. From this information, a linear programming problem is solved to identify a class of value functions in which the DM's preference information falls. They considered three classes of value functions for further processing: (i) linear, (ii) quasi-concave and (iii) non-quasi-concave. Based on this classification, a dominance structure is defined and either by search or from an existing sample of alternatives, the expected probabilities of finding new and better alternatives are determined. If there is a reasonable probability of finding better points, the algorithm is continued, otherwise the currently judged most preferred point is reported. An extension of this study [16] used a sampling based statistical procedure to compute expected probabilities of finding better solutions. It is clear that the algorithm is likely to perform better if the sampling procedure is replaced by an evolutionary multi-objective optimization algorithm for finding new points. After every decision-making event, an EMO algorithm can be employed for a few generations to find a better population of points, if available. Motivated by this study and recognizing the need for a simple interactive preference-based approach involving a DM in an EMO framework, we launch this particular study.

3 Proposed Progressively Interactive EMO (PI-EMO)

In this section, we propose a value function based interactive EMO algorithm, where an approximate value function is generated progressively after every few generations. A standard EMO algorithm (such as NSGA-II [17], SPEA2 [22] and others) works with a population of points in each iteration and prefers a sparsed set of non-dominated points in a population so that the algorithm progresses towards the Pareto-optimal front and aims at finding a representative set over the entire front. However, in our proposed approach, we are interested in utilizing DM's preference information repeatedly as the algorithm progresses and in directing the search on the corresponding preferred region of the Pareto-optimal front iteratively.

For this purpose, after every τ generations of an EMO algorithm, we provide the decision-maker with η (≥ 2) well-sparsed non-dominated solutions from the current set of non-dominated points and expect the decision-maker to provide some preference information about superiority or indifference of one solution over the other. In the ideal situation, the DM can provide a complete ranking (from best to worst) of these solutions, but partial preference information is also allowed. With the given preference information, we then construct a strictly increasing polynomial value function. The construction procedure involves solving a single-objective optimization problem. Till the next τ generations, we use the constructed value function to direct the search for additional such preferred solutions. A termination condition is also set up based on the expected progress, which can be made with respect to the constructed value function. In the following, we provide a step-by-step procedure of the proposed progressively interactive EMO (PI-EMO) methodology:

Step 1: Initialize a population Par_0 and set iteration counter $t = 0$. Domination of one solution over another is defined based on the usual definition of dominance [23] and an EMO algorithm is executed for τ iterations. The value of t is incremented by one after each iteration.

Step 2: If ($t \mod \tau = 0$), cluster the current non-dominated front to choose η widely distributed points: otherwise, proceed to Step 5.

Step 3: Obtain decision-maker's preferences on η points. Construct a value function $V(\mathbf{f})$ from this information by solving an optimization problem (VFOP), described in Section 3.1. If no feasible value function is found satisfying all DM's preference information, we move to Step 5 and use the usual domination principle in EMO operators.

Step 4: A termination check (described in Section 3.2) is performed based on the expected improvement in solutions from the currently judged best solution based on the value function $V(\mathbf{f})$. If the expected improvement is not significant (with respect to a parameter d_s), the algorithm is terminated and the current best solution is chosen as the final outcome.

Step 5: The parent population Par_t is used to create a new offspring population Off_t by using a modified domination principle (discussed in Section 3.3) based on the current value function $V(\mathbf{f})$ and EMO algorithm's search operators.

Step 6: Populations Par_t and Off_t are used to determine a new population Par_{t+1} using the current value function and EMO algorithm's diversity preserving operator. The iteration counter is incremented as $t \leftarrow t+1$ and the algorithm proceeds to Step 2.

The above is a generic progressively interactive PI-EMO procedure, which can be combined with any existing EMO algorithm in Step 1 and subsequently in Steps 5 and 6. The PI-EMO algorithm expects the user to set a value of τ, η and d_s.

In Step 2, points in the best non-dominated front are considered and the k-mean clustering algorithm [1, 22] can be used to identify η well-diversified points in the objective space. Other multi-criteria decision making methodologies [35] of selecting points from a set of non-dominated points may also be used.

We now provide the details for the specific procedures used in this study for Steps 3 to 6.

3.1 Step 3: Decision Maker's Preference Information and Construction of a Polynomial Value Function

At an instance of a DM call, η points are presented to the DM. Based on an analysis, the DM is then expected to provide some preference information. One of the usual ways of providing such information is to make pairwise comparisons of given points and suggest one of the two scenarios: (i) a solution is more preferred over the other or (ii) both solutions are incomparable. Based on such a preference statement, it is expected that for some pairs (i, j) of points, the i-th point is found to be preferred over the j-th point, thereby establishing $P_i \succ P_j$ and for some pairs (i, j), they are incomparable ($P_i \equiv P_j$). It is expected that the DM is able to establish at least one pair satisfying $P_i \succ P_j$. Thus, at the end of DM's preference elicitation task, we shall have at least one point which lies in the best category and at least one point which lies in the second-best category. In the 'complete ranking' situation, the DM may provide a complete ranking of η solutions (say, P_1 being the best, P_2 being the second-best and so on till P_η being the least preferred point).

Given such preference information, the task is to construct a polynomial value function satisfying the given preference structure. A similar task has been performed for linear utility functions elsewhere [34, 15]. The study [15] also suggested a procedure for checking if there can exist a quasi-concave utility function satisfying the given ranking, but no specific value function was constructed. Here, we construct a simple mathematical value function to capture the given preference information of η points.

3.1.1 Polynomial Value Function for Two Objectives

A value function is formed based on preference information provided by the decision maker. We first describe the procedure for two objectives and then present the procedure for the generic case. The structure of the value function is fixed as follows:

$$\begin{array}{lll} & V(f_1, f_2) = (f_1 + k_1 f_2 + l_1)(f_2 + k_2 f_1 + l_2), & \\ \text{where} & f_1, f_2 & \text{are the objective functions} \\ \text{and} & k_1, k_2, l_1, l_2 & \text{are the value function parameters} \end{array} \qquad (1)$$

The value function V, for two objectives shown above, is considered to be the product of two linear functions $S_1 : \mathbf{R}^2 \rightarrow \mathbf{R}$ and $S_2 : \mathbf{R}^2 \rightarrow \mathbf{R}$. The parameters k_1, k_2, l_1 and l_2 are unknown and must be determined from the preference information of η points supplied by the decision-maker (DM). For this purpose, we solve the following optimization problem (VFOP):

$$\begin{array}{ll} \text{Maximize} & \epsilon, \\ \text{subject to} & V \text{ is non-negative at every point } P_i, \\ & V \text{ is strictly increasing at every point } P_i, \\ & V(P_i) - V(P_j) \geq \epsilon, \quad \text{for all } (i,j) \text{ pairs} \\ & \qquad \text{satisfying } P_i \succ P_j, \\ & |V(P_i) - V(P_j)| \leq \delta_V, \quad \text{for all } (i,j) \text{ pairs} \\ & \qquad \text{satisfying } P_i \equiv P_j. \end{array} \qquad (2)$$

The first two sets of constraints ensure that the derived value function is non-negative and strictly increasing at all η points. The value function can always be shifted by adding a constant term to the function and thus can also be made negative if required. Without loss of generality, we construct a value function which assigns a positive value to all the data points. These conditions satisfy the quasi-concavity of the value function – a desired property suggested in the economics literature [36]. This property in fact corresponds with the convex towards the origin indifference contours. The third and fourth sets of constraints ensure that the preference order supplied by the decision maker is maintained for respective pairs. In order to implement the first two constraint sets, we first sketch the value function $V(f_1, f_2)$ with the desired properties of being non-negative and strictly increasing. Figure 1 shows a pair of straight lines represented by $V(f_1, f_2) = 0$ at which either (or both) of the two terms S_1 or S_2 is zero. However, if the chosen points P_i $(i = 1, \ldots, \eta)$ are such that both S_1 and S_2 are non-negative at these points, the first set of constraints will be satisfied. A generic iso-value curve for which $S_m > 0$ (for $m = 1, 2$) is also depicted in the figure. Thus, the first set of constraints can be satisfied by simply considering $S_m \geq 0$ for $m = 1, 2$. To impose strictly increasing nature of the value function at the chosen points, we can use $\partial V/\partial f_i \geq 0$ for both objectives. For the two-objective case, these two conditions yield $S_2 + k_2 S_1 \geq 0$ and $k_1 S_2 + S_1 \geq 0$. The fourth constraint set takes into account all pairs of incomparable points. For such pairs of points, we

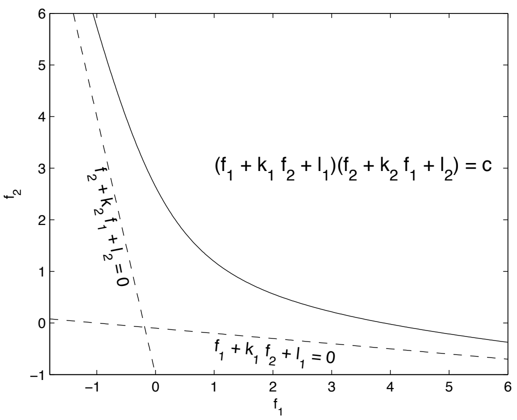

Figure 1: The proposed value function.

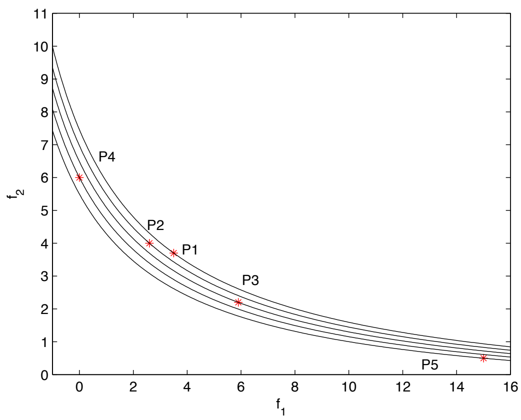

Figure 2: Value function found by optimization.

would like to restrict the absolute difference between their value function values to be within a small range (δ_V). To eliminate having another parameter, we may like to use $\delta_V = 0.1\epsilon$, such that it is at most 10% of the maximum difference in value functions between $\succ$-class of points.

A little thought will reveal that the above optimization problem attempts to find a value function for which the maximum difference in the value function values between the ordered pairs of points is maximum. Considering all the expressions, we have the following optimization problem:

$$
\begin{array}{rl}
\text{Maximize} & \epsilon, \\
\text{subject to} & S_m(P_i) \geq 0, \quad i = 1, 2, \ldots, \eta, \text{ and } m = 1, 2, \\
& S_2(P_i) + k_2 S_1(P_i) \geq 0, \quad i = 1, 2, \ldots, \eta, \\
& k_1 S_2(P_i) + S_1(P_i) \geq 0, \quad i = 1, 2, \ldots, \eta, \\
& V(P_i) - V(P_j) \geq \epsilon, \quad \text{for all } (i, j) \text{ pairs} \\
& \qquad \text{satisfying } P_i \succ P_j, \\
& |V(P_i) - V(P_j)| \leq \delta_V, \quad \text{for all } (i, j) \text{ pairs} \\
& \qquad \text{satisfying } P_i \equiv P_j.
\end{array}
\tag{3}
$$

Figure 2 considers five ($\eta = 5$) hypothetical points ($P_1 = (3.5, 3.7)$, $P_2 = (2.6, 4.0)$, $P_3 = (5.9, 2.2)$, $P_4 = (0.0, 6.0)$, and $P_5 = (15.0, 0.5)$) and a complete ranking of the points (P_1 being best and P_5 being worst). Due to a complete ranking, we do not have the fourth constraint set. The solution to the above optimization problem results in a value function, the contours (iso-utility curves) of which are drawn in the figure. The value function obtained after the optimization is as follows:

$$V(f_1, f_2) = (f_1 + 4.3229)(f_2 + 0.9401).$$

The asymptotes of this value function are parallel to f_1 and f_2 axes. It is interesting to note the preference order and other restrictions are maintained by the obtained value function.

3.1.2 Polynomial Value Function for M Objectives

The above suggested methodology can be applied to any number of objectives. For a general M objective problem the value function can be written as follows:

$$\begin{aligned} V(\mathbf{f}) = \; & (f_1 + k_{11}f_2 + k_{12}f_3 + \ldots + k_{1(M-1)}f_M + l_1)\times \\ & (f_2 + k_{21}f_3 + k_{22}f_4 + \ldots + k_{2(M-1)}f_1 + l_2)\times \\ & \ldots \\ & (f_M + k_{M1}f_1 + k_{M2}f_4 + \ldots + k_{M(M-1)}f_{M-1} + l_M) \end{aligned} \quad (4)$$

In the above formulation it should be noted that the subscripts of the objective functions change in a cyclic manner as we move from one product term to the next. The number of parameters in the value function is M^2. The optimization problem formulation for the value function suggested above contains $M^2 + 1$ variables. The variable ϵ is to be maximized. The second set of constraints (strictly increasing property of V) will introduce non-linearity. To avoid this, we simplify the above constraints by restricting the strictly increasing property of each term S_k, instead of V itself. The resulting constraints then become $k_{ij} \geq 0$ for all i and j combinations. The optimization problem (VFOP) to determine the parameters of the value function can thus be generalized as follows:

$$\begin{aligned} \text{Maximize} \quad & \epsilon, \\ \text{subject to} \quad & S_m(P_i) \geq 0, \quad i = 1, \ldots, \eta \text{ and } m = 1, \ldots, M, \\ & k_{ij} \geq 0, \quad i = 1, \ldots, M, \text{ and } j = 1, \ldots, (M-1), \\ & V(P_i) - V(P_j) \geq \epsilon, \quad \text{for all } (i, j) \text{ pairs} \\ & \qquad \text{satisfying } P_i \succ P_j, \\ & \qquad \text{combinations satisfying } i < j, \\ & |V(P_i) - V(P_j)| \leq \delta_V, \quad \text{for all } (i, j) \text{ pairs} \\ & \qquad \text{satisfying } P_i \equiv P_j. \end{aligned} \quad (5)$$

In the above problem, the objective function and the first two constraint sets are linear, however the third and fourth constraint sets are polynomial in terms of the problem variables. There are a total of $M\eta + M(M-1)$ linear constraints. However, the number of polynomial constraints depends on the number of pairs for which the preference information is provided by the DM. For a 10-objective ($M = 10$) problem having $\eta = 5$ chosen points, the above problem has 101 variables, 140 linear constraints. and at most 10 ($\binom{5}{2}$) polynomial constraints. Since majority of the constraints are linear, we suggest using a sequential quadratic programming (SQP) algorithm to solve the above problem. The non-differentiability of the fourth constraint set can be handled by converting each constraint ($|g(\mathbf{x})| \leq \delta_V$) into two constraints ($g(\mathbf{x}) \geq -\delta_V$ and $g(\mathbf{x}) \leq \delta_V$)). In all our problems, we did not consider the cases involving the fourth constraint and leave such considerations for a later study.

3.2 Step 4: Termination Criterion

Once the value function V is determined, the EMO algorithm is driven by it in the next τ generations. The value function V can also be used for determining whether the overall optimization procedure should be terminated or not. To implement the idea we identify the best and second-best points P_1 and P_2 from the given set of η points based on the preference information. In the event of more than one point in each of the top two categories (best and second-best classes) which can happen when the '$\equiv$'-class exists, we choose P_1 and P_2 as the points having highest value function value in each category, respectively.

The constructed value function can provide information about whether any new point P is better than the current best solution (P_1) with respect to the value function. Thus, if we perform a single-objective search along the gradient of the value function (or ∇V) from P_1, we expect to create better preferred solutions than P_1. We can use this principle to develop a termination criterion.

We solve the following achievement scalarizing function (ASF) problem [25] for $P_1 = \mathbf{z}^b$:

$$\begin{aligned} \text{Maximize} \quad & \left(\min_{i=1}^{M} \frac{f_i(\mathbf{x}) - z_i^b}{\frac{\partial V}{\partial f_i}} \right) + \rho \sum_{j=1}^{M} \frac{f_j(\mathbf{x}) - z_j^b}{\frac{\partial V}{\partial f_j}}. \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{S}. \end{aligned} \tag{6}$$

Here, $\mathcal{S}$ denotes the feasible decision variable space of the original problem. The second term with a small ρ ($= 10^{-10}$ is used here) prevents the solution from converging to a weak Pareto-optimal point. Any single-objective optimization method can be used for solving the above problem and the intermediate solutions ($\mathbf{z}^{(i)}$, $i = 1, 2, \ldots$) can be recorded. If at any intermediate point, the Euclidean distance between $\mathbf{z}^{(i)}$ from P_1 is larger than a termination parameter d_s, we stop the ASF optimization task and continue with the EMO algorithm. In this case, we replace P_1 with $\mathbf{z}^{(i)}$. Figure 3 depicts this scenario. On the other hand, if at the end of the SQP run, the final SQP solution (say, $\mathbf{z}^T$) is not greater than d_s distance away from P_1, we terminate the EMO algorithm and declare $\mathbf{z}^T$ as the final preferred solution. This situation indicates that based on the current value function, there does not exist any solution in the search space which will provide a significantly better solution than P_1. Hence, we can terminate the optimization run. Figure 4 shows such a situation, warranting a termination of the PI-EMO procedure.

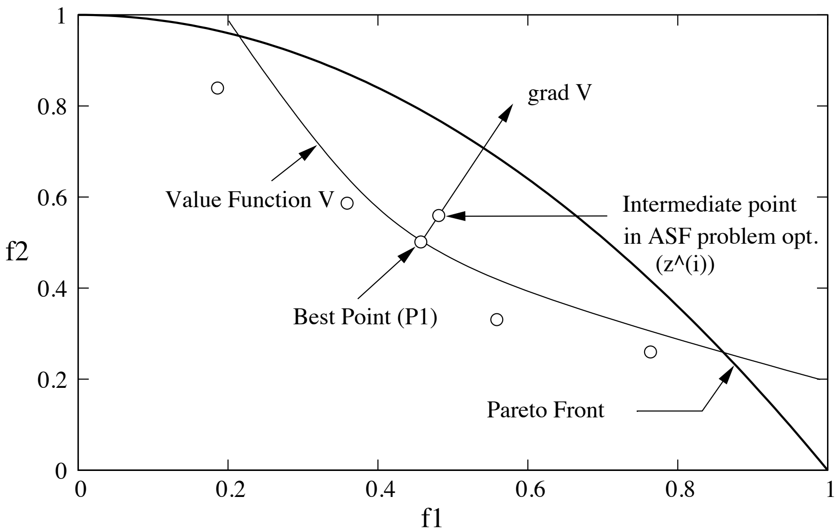

Figure 3: Local search, when far away from the front, finds a better point more than distance d_s away from the best point. Hence, no termination of the P-EMO.

3.3 Steps 5 and 6: Modified Domination Principle

The utility function V can also be used to modify the domination principle in order to emphasize and create preferred solutions.

Let us assume that the value function from the most recent decision-making interaction is V. The value function value for the second-best member (P_2 defined in the previous subsection) from the set of η points given to the DM is V_2. 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:

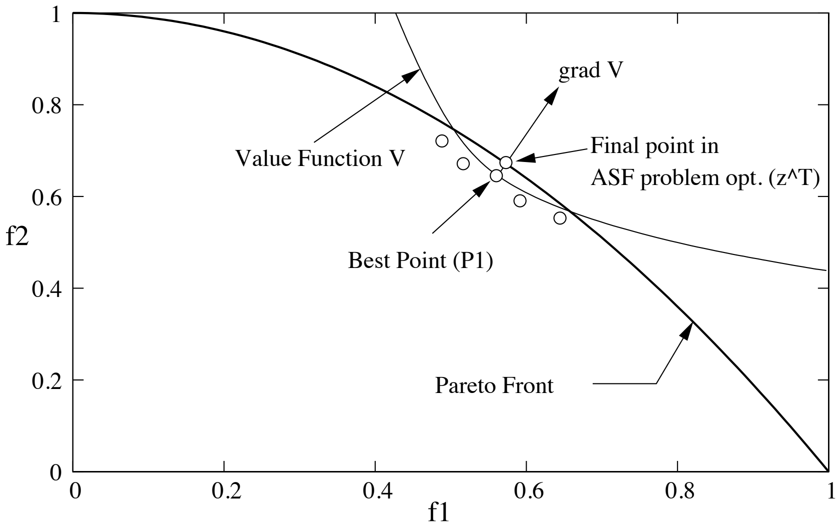

Figure 4: Local search terminates within distance d_s from the best point. Hence, the P-EMO is terminated.

1. If both solutions have a value function value *less* than V_2, then the two points are compared based on the usual dominance principle.

2. If both solutions have a value function value *more* than V_2, then the two points are compared based on the usual dominance principle.

3. If one has value function value more than V_2 and the other has value function value less than V_2, then the former dominates the latter.

Figure 5 illustrates the region dominated by two points A and B. The value function contour having a value V_2 is shown by the curved line. The point A lies in the region in which the value function is smaller than V_2. 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 in which the value function is larger than V_2. 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 a smaller value function value than V_2.

We now discuss the reason for choosing the baseline value function value at P_2 (as opposed to at P_1) for defining the modified dominance criterion above. While providing preference information on η points given to the DM, the DM has the knowledge of η points. Consider the scenario in Figure 6, in which the point $\mathbf{z}^*$ may lie in between P_1 and P_2. If the value function at P_1 is considered as the baseline value for domination, the most preferred point $\mathbf{z}^*$ will get dominated by points like P_1. In higher objective problems, the most preferred point may lie elsewhere and considering V_2 may also be too stringent. To be more conservative, $V(P_\eta)$ can be considered as the baseline value in the modified domination criterion.

The above modified domination principle can be used in both steps 5 and 6 for creating the new population Off_t and for selecting the new population Par_{t+1}.

Although we do not handle constrained problems in this study, the above modified domination principle can be extended for handling constraints. As defined in [17], when both solutions under consideration for a domination check are *feasible*, the above domination principle can simply be used to establish dominance of one over the other. However, if one point is feasible and the other is not, the feasible solution can be declared as dominating the other. Finally, if both points are infeasible, the one having smaller overall constraint violation may be declared as dominating the other. We defer consideration of a constrained PI-EMO to a later study.

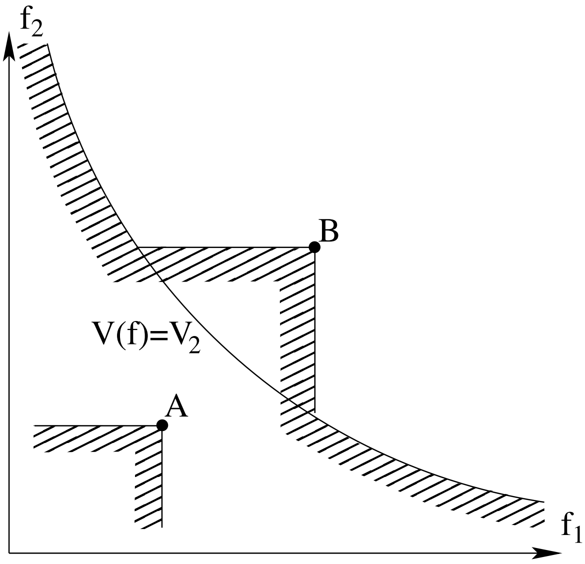

Figure 5: Dominated regions of two points A and B using the modified definition.

4 PI-NSGA-II Procedure

In the PI-NSGA-II procedure, the first τ generations are performed according to the usual NSGA-II algorithm [17]. Thereafter, we modify the NSGA-II algorithm by using the modified domination principle (discussed in Section 3.3) in the elite-preserving operator and also in the tournament selection for creating the offspring population. We also use a different recombination operator in this study. After a child solution $\mathbf{x}^C$ is created by the SBX (recombination) operator [26], two randomly selected population members $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are chosen and a small fraction of the difference vector is added to the child solution (similar in principle to a differential evolution operator [27]), as follows:

$$\mathbf{x}^C = \mathbf{x}^C + 0.1\left(\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\right). \tag{7}$$

The crowding distance operator of NSGA-II has been replaced with k-means clustering for maintaining diversity among solutions of the same non-dominated front.

The value function optimization problem is solved using the SQP code of KNITRO software [28]. The termination is set if the Karush-Kuhn-Tucker (KKT) error measure computed within KNITRO is less than or equal to 10^{-6}.

For termination check (discussed in Section 3.2), we also use the SQP code of KNITRO software and the SQP algorithm is terminated (if not terminated due to d_s distance check from P_1 discussed earlier) when the KKT error measure is less than or equal to 10^{-6}.

5 Results

In this section, we present the results of the PI-NSGA-II procedure on two, three, and five objective test problems. ZDT1 and DTLZ2 test problems are adapted to create maximization problems. In all simulations, we have used the following parameter values:

1. Number of points given to the DM for preference information: $\eta = 5$.
2. Number of generations between two consecutive DM calls: $\tau = 5$.
3. Termination parameter: $d_s = 0.01$.

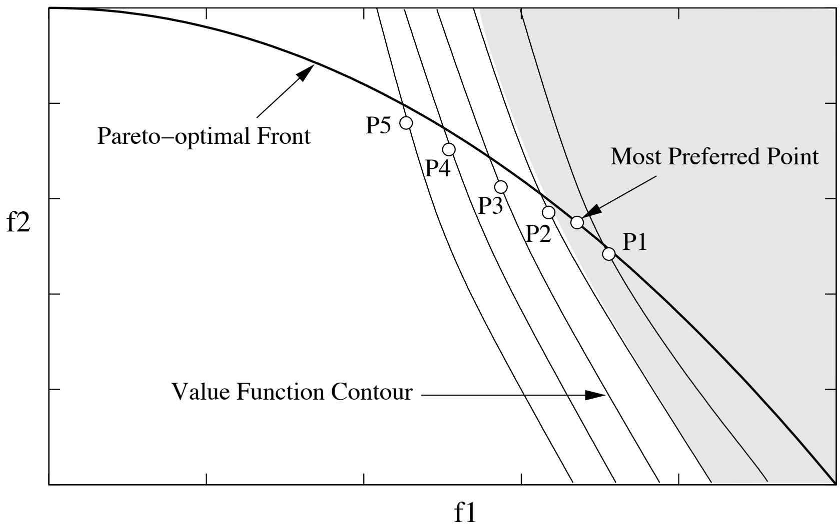

Figure 6: A scenario in which final preferred point may lie between $P1$ and P_2 for a two-objective problem.

4. Crossover probability and the distribution index for the SBX operator: $p_c = 0.9$ and $\eta_c = 15$.

5. Mutation probability: $p_m = 0$.

6. Population size: $N = 10M$, where M is the number of objectives.

In the optimization of the VFOP problem (given in equation 4), we restrict the bounds of parameters as follows: $0 \le (k_1, k_2) \le 1000$ and $-1000 \le (l_1, l_2) \le 1000$. In the next section, we perform a parametric study with some of the above parameters. Here, we present the test problems and results obtained with the above setting.

5.1 Two-Objective Test Problem

Problem 1 is adapted from ZDT1 and has 30 variables [29].

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

The Pareto-optimal front is given by $f_2 = 10 - \sqrt{f_1}$ and is shown in Figure 7. The solutions are $\mathbf{x}_i = 0$ for $i = 2, 3, \ldots, 30$ and $x_1 \in [0, 1]$.

This maximization problem has a non-convex front, therefore if the decision maker is not interested in the end points, the value function has to be non-linear. A linear value function will always lead to the end points of the front. In our simulations, we assume a particular value function which acts as a representative of the DM, but the information is not explicitly used in creating new solutions by the operators of the PI-NSGA-II procedure. In such cases, the most preferred point $\mathbf{z}^*$ can be determined from the chosen value function beforehand, thereby enabling us to compare our obtained point with $\mathbf{z}^*$.

In our study, we assume the following non-linear value function (which acts as a DM in providing a complete ranking of η solutions at every τ generations):

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

This value function gives the most preferred solution as $\mathbf{z}^* = (0.25, 9.50)$. The contours of this value function are shown in Figure 7. Since a DM-emulated value function is used to decide on preference of one point to the other in pairwise comparisons, we shall have complete ranking information of all η points in our study. Thus, we shall not have the fourth set of constraints in determining the value function, as given in equation 4. In a future study, we shall consider partial preference information and its effect on the constructed value function.

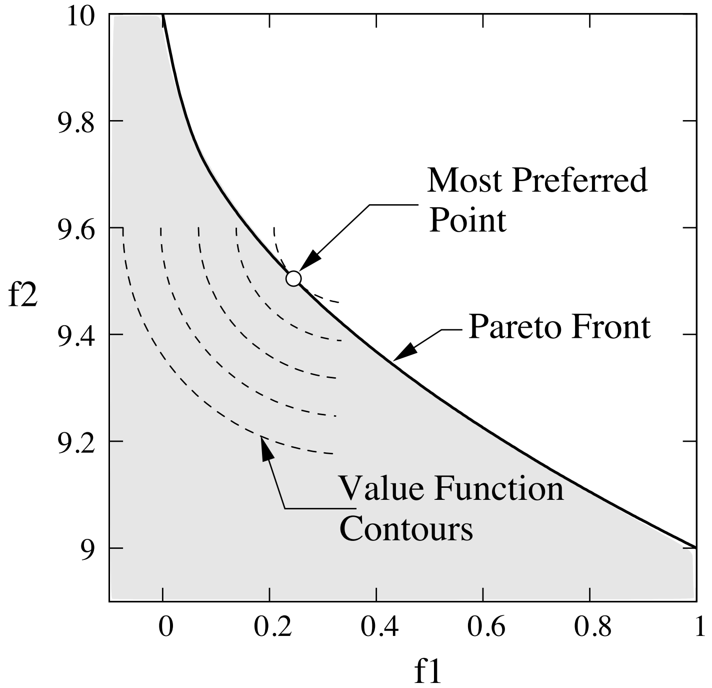

Figure 7: Contours of the chosen value function (acts as a DM) and the most preferred point corresponding to the value function.

Table 1 presents the best, median and worst of 21 different PI-NSGA-II simulations (each starting with a different initial population). The performance (accuracy measure) is computed based on the Euclidean distance of each optimized point with $\mathbf{z}^*$. Note that this accuracy measure is different from the termination criterion used in the PI-NSGA-II procedure. Table 2 shows minimum, median and maximum accuracy,

Table 1: Final solutions obtained by PI-NSGA-II for the modified ZDT1 problem.

	$\mathbf{z}^*$	Best	Median	Worst
f_1	0.2500	0.2498	0.2461	0.2713
f_2	9.5000	9.5002	9.5038	9.4791

the number of overall function evaluations, and the number of DM calls recorded in the 21 runs. The table indicates that the proposed PI-NSGA-II procedure is able to find a solution close to the final preferred solution. Although the overall number of function evaluations depend on the initial population, for a 30-variable problem these numbers are reasonable.

We now show the working of the PI-NSGA-II approach for a particular run, which required 14 DM calls before termination. Figure 8 shows the value functions optimized after various DM calls. The first DM call was made after generation 5. Five chosen points (P1 to P5 shown in shaded circles) from the non-dominated solutions at generation 5 are shown in the figure. The best and second-best points are close to

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

	Minimum	Median	Maximum
Accuracy	0.0001	0.0062	0.0197
Func. Evals.	5,408	7,372	11,809
# of DM Calls	14	19	30

each other. The strictly increasing requirement of the value function imposed in the optimization process creates an almost linear value function as an optimum choice in this case. The corresponding parameter values of the value function are: ($k_1 = 998.189$, $k_2 = 0.049$, $l_1 = 369.532$, and $l_2 = 137.170$). The value functions are drawn at the second-best point. After five more generations, the DM is called to provide preference information the second time. The corresponding value function drawn at the second-best point is shown in the figure. Five points used for preference ranking are shown as diamonds. The figure shows how the PI-NSGA-II procedure finds better and better points and how progressively the DM calls enable the overall procedure to find refined value functions. Eventually, at the 14th DM call, all five solutions come very close to $\mathbf{z}^*$ and the algorithm terminates with the imposed $d_s = 0.01$ condition. The optimal

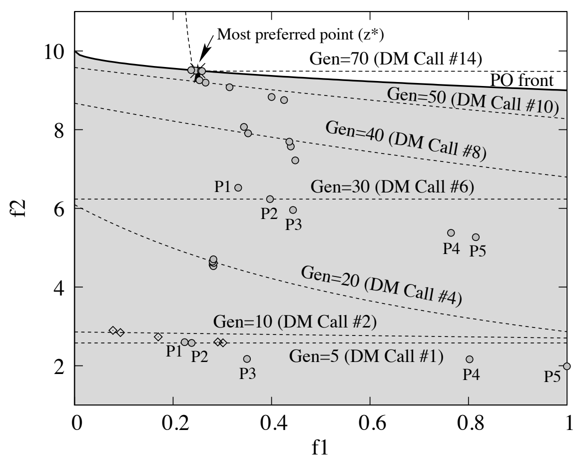

Figure 8: Evolution of value functions after successive DM calls.

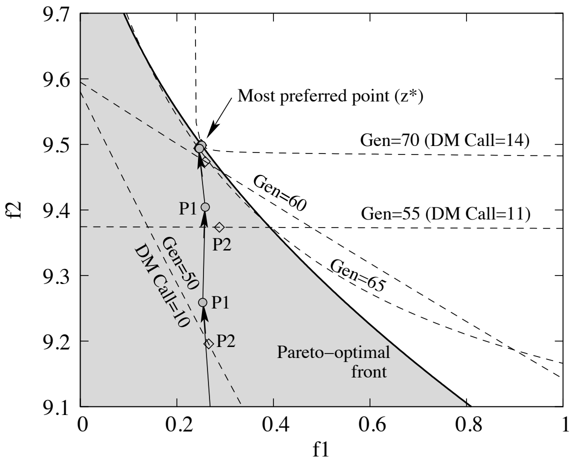

Figure 9: Value functions near the most preferred point.

parameter values fixing the value functions at various DM calls are shown in Table 3. Although no pattern in these parameter values is observed from one DM call to another, every value function thus obtained is strictly increasing and maximizes the maximum difference in value function values between any two chosen points. However, the NSGA-II algorithm with these value functions in five subsequent generations seems to guide the best point towards the most preferred point ($\mathbf{z}^*$) progressively.

Figure 9 shows the value functions from the 10th DM call onwards. In this figure, the value functions are drawn at the second-best point (shown with a diamond) and the corresponding best point is also shown by a shaded circle. It is interesting to observe how the value functions get modified with generations and how the modified value functions help find better non-dominated solutions progressively with the help of modified domination and NSGA-II operators. The final point obtained by the PI-NSGA-II is $(f_1, f_2) = (0.251, 9.499)$, which is very close to the most preferred point $\mathbf{z}^* = (0.25, 9.5)$ corresponding to the optimum of the DM-emulated value function given in equation 9.

Table 3: Optimal parameter values determining the value function and corresponding best point at various DM calls.

DM Call	k_1	k_2	l_1	l_2	$P_1 = (f_1, f_2)$
#1	998.189	0.049	369.532	137.170	(0.223, 2.600)
#2	999.998	19.699	114.161	359.199	(0.078, 2.898)
#3	821.797	0.003	-15.116	770.050	(0.260, 4.321)
#4	1000.000	440.133	87.366	393.896	(0.282, 4.706)
#6	804.650	0.033	-99.871	567.481	(0.332, 6.525)
#8	807.395	105.691	-30.880	365.454	(0.344, 8.066)
#10	403.750	49.007	-30.667	290.960	(0.254, 9.259)
#14	0.007	0.006	-0.308	-9.488	(0.251, 9.499)

5.2 Three-Objective Test Problem

The DTLZ2 test problem [30] is scalable to number of objectives. In the three-objective case, all points (objective vectors) are bounded by two spherical surfaces in the first octant. In the case of minimizing all objectives, the inner surface (close to the origin) becomes the Pareto-optimal front. But here, we maximize each objective of the DTLZ2 problem. Thus, the outer spherical surface becomes the corresponding Pareto-optimal front. An M-objective DTLZ2 problem for maximization is given 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 a three-objective DTLZ2 problem is shown in Figure 10. The points (objective vectors) on the Pareto-optimal front follow the relation: $f_1^2 + f_2^2 + f_3^2 = 3.5^2$. The decision variable values correspond to $x_1 \in [0, 1]$, $x_2 \in [0, 1]$ and $x_i = 0$ or 1 for $i = 3, 4, \ldots, 12$.

To test the working of PI-NSGA-II on this problem, we have replaced the decision maker by using a linear value function (emulating the DM), as follows:

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

This value function produces the most preferred solution on the Pareto-optimal front as $\mathbf{z}^* = (1.25, 1.50, 2.9047)$.

The PI-NSGA-II is run with $N = 10 \times 3$ or 30 population members 21 times, each time with a different random initial population. In terms of the accuracy measure from $\mathbf{z}^*$, Table 4 presents the minimum, median and worst performing runs. Table 5 shows the accuracy, number of overall function evaluations and number of DM calls needed by the procedure. It is clear that the obtained points are close to the most preferred point $\mathbf{z}^*$. Figure 10 shows the population at the final generation of a typical PI-NSGA-II run.

5.3 Five-Objective Test Problem

We now consider the five-objective ($M = 5$) version of the DTLZ2 problem described in the previous subsection. The Pareto-optimal front is described as $f_1^2 + f_2^2 + f_3^2 + f_4^2 + f_5^2 = 3.5^2$.

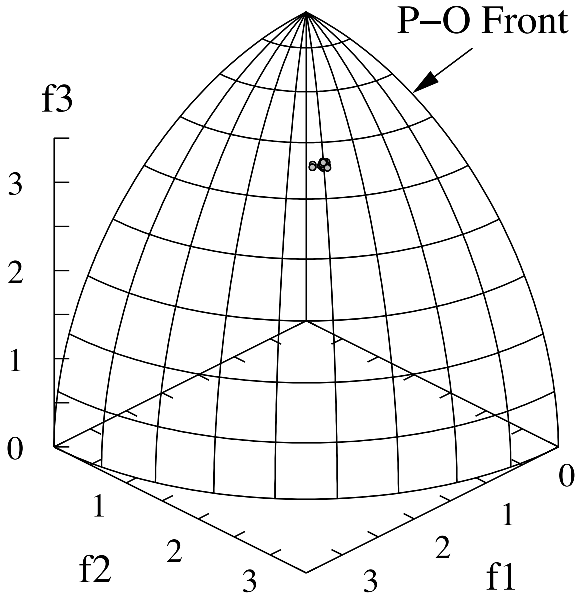

Figure 10: Final population members after termination of the algorithm for three-objective modified DTLZ2 problem. The complete Pareto-optimal surface is marked as ‘P-O front’.

Table 4: Final solutions obtained by PI-NSGA-II for the three-objective modified DTLZ2 problem.

	$\mathbf{z}^*$	Best	Median	Worst
f_1	1.2500	1.2459	1.2197	1.3178
f_2	1.5000	1.5050	1.4888	1.4755
f_2	2.9047	2.9039	2.9233	2.8873

For this problem, we choose a non-linear DM-emulated value function, as follows:

$$\begin{aligned} V(\mathbf{f}) &= 1/\left((f_1-1.1)^2+(f_2-1.21)^2+(f_3-1.43)^2\right.\\ &\quad \left.+(f_4-1.76)^2+(f_5-2.6468)^2\right). \end{aligned} \tag{12}$$

This value function produces the most preferred point as $\mathbf{z}^* = (1.0, 1.1, 1.3, 1.6, 2.4062)$.

Table 6 presents the obtained solutions by PI-NSGA-II with 50 population members. Table 7 shows the accuracy measure, the number of overall function evaluations, and the number of DM calls. Although the points close to the most preferred point are obtained in each run, the higher dimensionality of the problem requires more function evaluations and DM calls compared to two and three-objective test problems. However, the above results are obtained for a strict termination criterion with $d_s = 0.01$. Smaller number of DM calls and evaluations are expected if this termination criterion is relaxed. We discuss these matters in the next section. It is worth mentioning that the application of an EMO (including NSGA-II) will face difficulties in converging to the entire five-dimensional Pareto-optimal front with an identical number of function evaluations, but since here our target is one particular preferred point on the Pareto-optimal front, it is possible to apply a PI-EMO to a five-objective optimization problem.

Table 5: Distance of obtained solution from the most preferred solution, number of function evaluations, and number of DM calls required by PI-NSGA-II on the three-objective modified DTLZ2 problem.

	Minimum	Median	Maximum
Accuracy	0.0008	0.0115	0.0434
Func. Evals.	4,200	6,222	8,982
# of DM Calls	17	25	36

Table 6: Final objective values obtained from PI-NSGA-II for the five-objective modified DTLZ2 problem.

	$\mathbf{z}^*$	Best	Median	Worst
f_1	1.0000	0.9931	0.9785	0.9455
f_2	1.1000	1.1382	1.0502	1.1467
f_3	1.3000	1.3005	1.3382	1.3208
f_4	1.6000	1.5855	1.5947	1.6349
f_5	2.4062	2.4007	2.4199	2.3714

6 Parametric Study

Besides the usual parameters associated with an evolutionary algorithm, such as population size, crossover and mutation probabilities and indices, tournament size etc., in the proposed PI-NSGA-II we have introduced a few additional parameters which may effect the accuracy and number of DM calls. They are the number of points used in obtaining DM's preference information (η), the number of generations between DM calls (τ), termination parameter (d_s), KKT error limit for terminating SQP algorithm in value function optimization and in single-objective optimization used for the termination check, and the parameter ρ used in the ASF function optimization. Of these parameters, the first three have shown to have an effect on the chosen performance measures — accuracy, the number of overall function evaluations, and the number of DM calls. As mentioned earlier, the parameter η is directly related to the maximum number of pairwise comparisons a DM would like to do in a single DM call. Of course, if more points can be compared, a more appropriate value function can be obtained. However, based on a maximum of 10 pairwise comparisons per DM call (in line with the recommendation of 7 ± 2 comparisons as suggested in [31]), we restrict $\eta = 5$ in this study and do not do a parametric study with this parameter. Thus, in this section, we study the effect of two parameters (τ and d_s), while keeping the other PI-NSGA-II parameters identical to that mentioned in the previous section. In each case, we use the same three test problems.

Table 7: Distance of obtained solution from the most preferred solution, function evaluations, and the number of DM calls required by PI-NSGA-II for the five-objective modified DTLZ2 problem.

	minimum	median	maximum
Accuracy	0.0084	0.0240	0.0902
# of Function Eval.	23,126	27,202	41,871
# of DM Calls	57	67	102

6.1 Effect of Frequency of DM Calls (τ)

First we study the effect of τ by considering four different values: 2, 5, 10 and 20 generations. The parameter d_s is kept fixed to 0.01. To investigate the dependence of the performance of the procedure on the initial population, in each case, we run PI-NSGA-II from 21 different initial random populations and plot the best, median and worst performance measures.

We plot three different performance measures — accuracy, number of DM calls and number of function evaluations obtained for the modified ZDT1 problem in Figure 11. It is interesting to note that all three

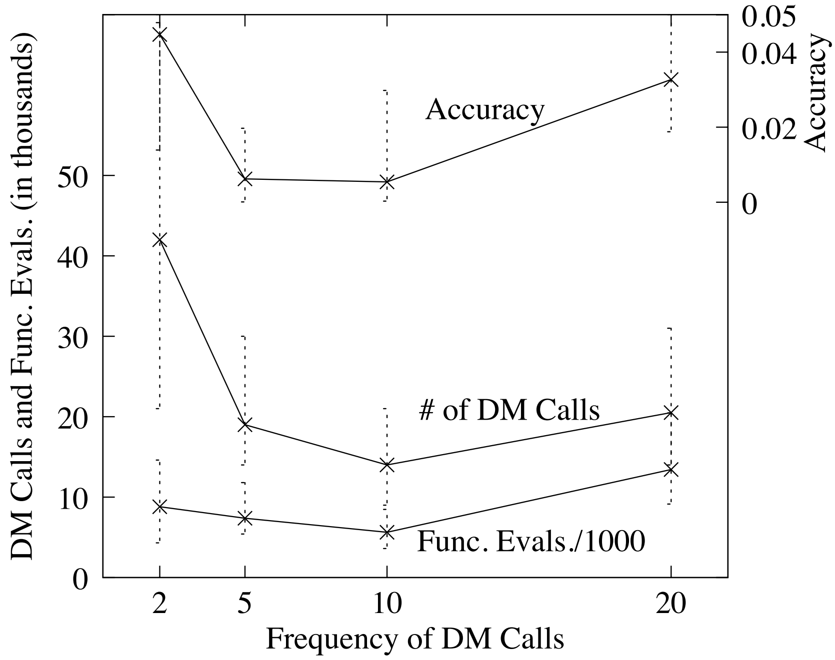

Figure 11: Three performance measures on modified ZDT1 problem for different τ values.

median performance measures are best for $\tau = 10$, although $\tau = 5$ also results in a similar accuracy and the number of DM calls. A small value of τ means that DM calls are to be made more frequently. Clearly, this results in higher number of DM calls, as evident from the figure. Frequent DM calls result in more single-objective optimization runs for termination check, thereby increasing the number of overall function evaluations. On the other hand, a large value of τ captures too little preference information to focus the search near the most preferred point, thereby causing a large number of generations to satisfy termination conditions and a large number of DM calls.

Figure 12 shows the same three performance measures on the three-objective modified DTLZ2 problem. For this problem, the number of DM calls is similar for $\tau = 5$ and 10 generations, whereas accuracy and the number of function evaluations are better for $\tau = 5$ generations. Once again, too small or too large τ is found to be detrimental.

For the five-objective modified DTLZ2 problem, $\tau = 5$ produces optimal median performance on the number of DM calls and accuracy (Figure 13). However, the overall function evaluations is smaller with smaller τ.

Based on these simulation studies on two, three and five-objective optimization problems, one can conclude that a value of τ within 5 to 10 generations is better in terms of an overall performance of the PI-NSGA-II procedure. This range of τ provides a good convergence accuracy, requires less function evaluations, and less DM calls to converge near the most preferred point.

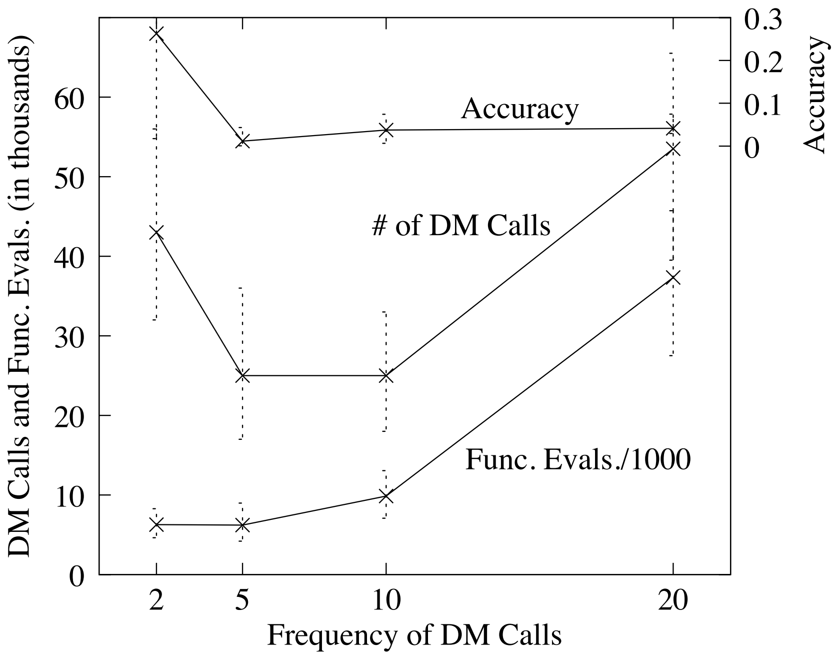

Figure 12: Three performance measures on three-objective modified DTLZ2 problem for different τ values.

6.2 Effect of Termination Parameter d_s

Now, we investigate the effect of the termination parameter d_s on the three performance measures on all three problems. In this study, we fix $\eta = 5$ and $\tau = 5$. Figure 14 shows the positive correlation between accuracy and d_s. As d_s is increased (meaning a relaxed termination), the obtained accuracy (distance from $\mathbf{z}^*$) gets worse. Interestingly, the associated variation in obtained accuracy over number of runs also gets worse. The flip side of increasing d_s is that the number of function evaluations reduces, as a comparatively lesser number of generations are now needed to satisfy the termination condition. Similarly, the number of DM calls also reduces with an increase in d_s.

Similar observations are made for three-objective and five-objective modified DTLZ2 problem, as evident from Figures 15 and 16, respectively.

These results clearly reveal the behavior of our proposed algorithm on the choice of d_s. Unlike in the parametric study of τ, where we observed an optimal range of values of τ for which the performance of PI-NSGA-II is better, here we find a monotonic variation in performance measures with d_s, however with a trade-off between accuracy and the number of DM calls (or, the number of function evaluations). This indicates that d_s need not be chosen as an arbitrarily small value. If approximate solutions are acceptable, they can be achieved with a smaller number of function evaluations and DM calls. Figure 17 shows the trade-off of these quantities for the modified three and five-objective DTLZ2 problems. The nature of the trade-off between accuracy and the number of DM calls indicates that at $d_s = 0.05$ makes a good compromise between these performance indicators for these problems, as a smaller d_s requirement calls for substantially more DM calls, and a larger d_s setting, although reduces the number of DM calls, makes a substantially large deviation from the most preferred solution.

7 Random Error in Preference Information

In the above simulations, a mathematical value function is used to emulate the preference information to be given by a DM. However, in practice, the DM is a human being (or representative of a group of human

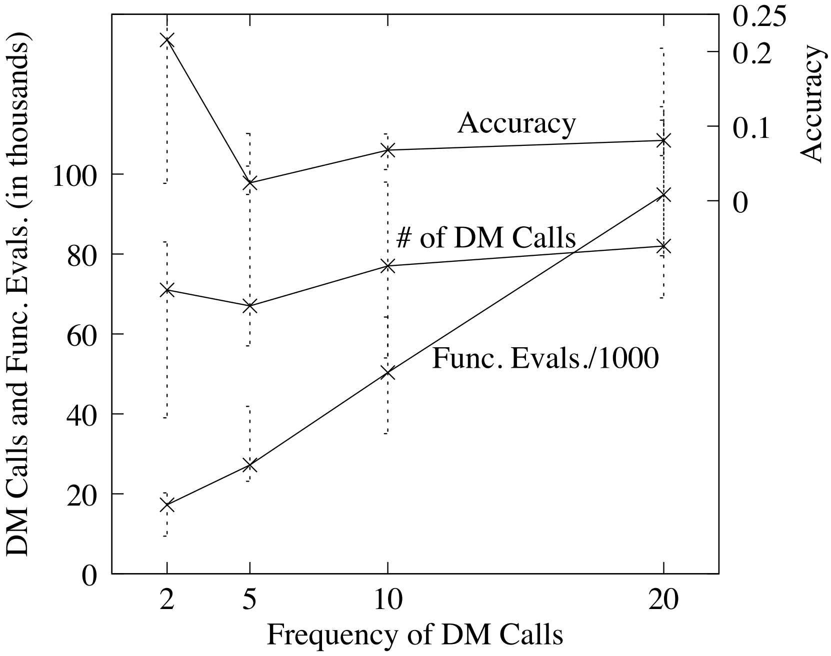

Figure 13: Three performance measures on five-objective modified DTLZ2 problem for different τ values.

beings). There is bound to be some level of inconsistencies in providing preference information from one DM call to another. To simulate the effect of this factor, we consider a DM-emulating value function which is stochastic in nature.

A linear value function similar to the one used before is chosen, but the coefficients of the value function are made stochastic. The stochasticity is reduced with the increase in the number of generations. This has been done to replicate a realistic DM who is likely to make errors during the start of the algorithm when he/she is in the process of learning his/her preferences. Later, the decision maker is likely to make more consistent decisions. Ironically, a large stochasticity in the beginning may also cause an algorithm to get misled from progressing towards the most preferred Pareto-optimal solution. A successful convergence of an algorithm in this case verifies that the algorithm does not get misdirected by inconsistent preference based information during the beginning of the run.

The DM-emulated value function used for the three-objective modified DTLZ2 problem is as follows:

$$V(f_1, f_2, f_3) = \text{noise}(1.25, \sigma) f_1 + \text{noise}(1.50, \sigma) f_2 + \text{noise}(2.9047, \sigma) f_3, \quad (13)$$

where σ is set as $\exp(-t/10)$ (t is the generation counter) and $\text{noise}(\mu, \sigma)$ refers to a random normal distribution with a mean μ and standard deviation σ. This setting ensures that the standard deviation of the noise around the mean reduces as the number of generations of the algorithm increases. With $\sigma = 0$, this value function gives the most preferred point as $\mathbf{z}^* = (1.25, 1.50, 2.9047)$. At the first instance of DM calls (that is, at $t = \tau = 5$ generations), $\sigma = \exp(-0.5) = 0.606$, meaning a significantly different value function than what is required for the algorithm to converge to the most preferred point.

Table 8 shows the best, median and worst points obtained by the PI-NSGA-II procedure with $\eta = 5$, $\tau = 5$, $d_s = 0.01$ and other parameter values used in Section 5.2. Again, 21 different runs were performed from different initial random populations. As clearly shown in Table 9, the accuracy for the best and median runs is small, despite the large stochasticities involved in the early stages of the optimization process. Although the number of function evaluations and the number of DM calls are 20 to 40% more compared to that in the deterministic DM-emulated value function case (Table 4), the accuracy of the final

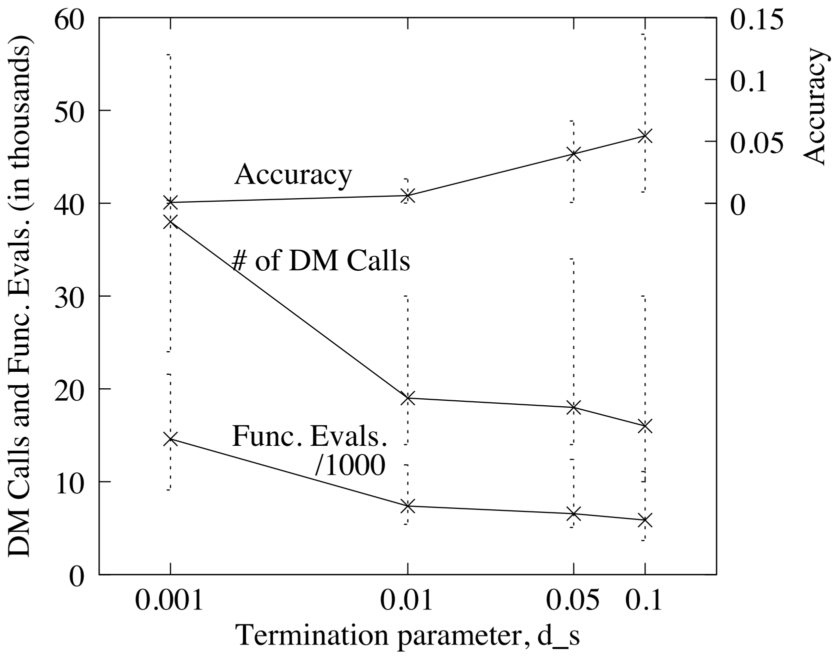

Figure 14: Three performance measures on modified ZDT1 problem for different d_s values.

Table 8: Final solutions obtained by PI-NSGA-II for the three-objective modified DTLZ2 problem with a stochastic DM-emulated value function.

	$\mathbf{z}^*$	Best	Median	Worst
f_1	1.2500	1.2555	1.2695	1.2902
f_2	1.5000	1.5105	1.5205	1.6437
f_3	2.9047	2.8969	2.8856	2.8078

point is good. This indicates that the final point is close to the most preferred solution for the deterministic case.

Next, we apply the PI-NSGA-II procedure to the five-objective modified DTLZ2 problem with the following stochastic value function to emulate the DM:

$$\begin{aligned} V(\mathbf{f}) &= \text{noise}(1.0,\sigma)f_1 + \text{noise}(1.1,\sigma)f_2 + \text{noise}(1.3,\sigma)f_3 \\ &\quad + \text{noise}(1.6,\sigma)f_4 + \text{noise}(2.4062,\sigma)f_5. \end{aligned} \tag{14}$$

The best, median, and worst points obtained by PI-NSGA-II are shown in Table 10. As shown by the performance measures in Table 11, despite somewhat larger function evaluations and number of DM calls, final points obtained by PI-NSGA-II are reasonably close to the most preferred point obtained for the deterministic version of the DM-emulated value function.

7.0.1 Effect of Extent of Stochasticity

In the above study, we used a noise factor on the coefficients of the DM-emulated value function, given as a function of generation counter t as follows: $\sigma = \exp(-t/10)$. As discussed above, at the first DM call with $\tau = 5$, this has an effect of having a standard deviation of 0.606 on each objective. We now investigate the effect of increasing the standard deviation by modifying the σ term as follows:

$$\sigma = s\exp(-t/10), \tag{15}$$

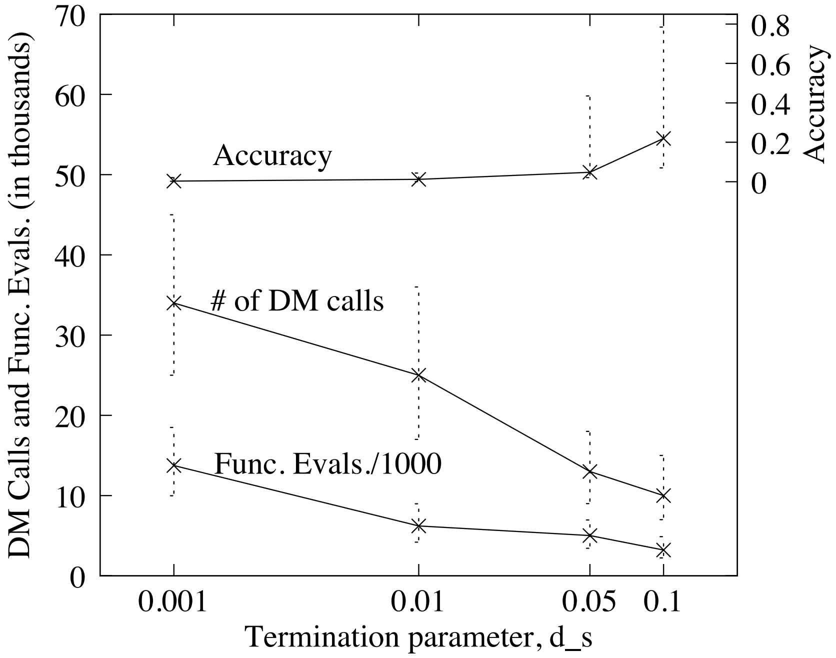

Figure 15: Three performance measures on three-objective modified DTLZ2 problem for different d_s values.

Table 9: Distance of obtained solution from the most preferred solution, function evaluations, and the number of DM calls by PI-NSGA-II for the three-objective modified DTLZ2 problem with a stochastic DM-emulated value function.

	Minimum	Median	Maximum
Accuracy	0.0142	0.0342	0.1779
Func. Evals.	5,841	7,608	9,663
# of DM Calls	24	31	39

where s is a stochasticity factor. For $s = 1$, we have an identical stochastic effect as in the previous subsection. By using a larger value of s, we can simulate a situation with more inconsistencies in the decision-making process. We use four different values of s: 1, 5, 10 and 100.

With a large value of s, it is expected that the DM-emulated value function provides a different ranking of η points than an ideal ranking (which would have been obtained without the stochasticity effect). We count the number of times the ranking of top three points is different from the ideal ranking of the same three points and tabulate it in Table 12 for a typical run. The corresponding function evaluations and accuracy in the final optimized point from $\mathbf{z}^*$ are also shown in the table. An increase in stochasticity in the decision-making process requires more DM calls and more function evaluations to achieve an acceptable accuracy and termination. Importantly, since all runs are terminated with $d_s = 0.01$ condition, despite large stochasticities involved in the beginning of PI-NSGA-II runs, the algorithm is able to find a point close to the most preferred Pareto-optimal point corresponding to the deterministic version of the DM-emulated value function.

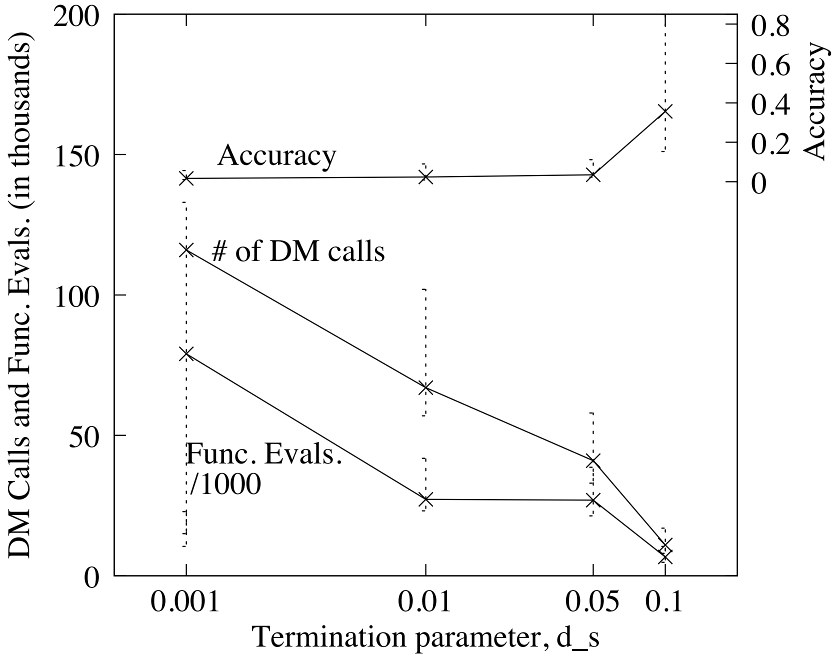

Figure 16: Three performance measures on modified five-objective modified DTLZ2 problem for different d_s values.

Table 10: Final solutions obtained by PI-NSGA-II for the five-objective modified DTLZ2 problem with a stochastic DM-emulated value function.

	$\mathbf{z}^*$	Best	Median	Worst
f_1	1.0000	1.0103	1.0875	1.0557
f_2	1.1000	1.1171	1.1495	1.2394
f_3	1.3000	1.3037	1.3525	1.4915
f_4	1.6000	1.6025	1.5942	1.4977
f_5	2.4062	2.4140	2.4125	2.4889

8 Extensions of Current Study

This study has suggested a simplistic yet elegant methodology by which the DM's preferences can be incorporated with an EMO algorithm so that the final target is not a complete Pareto-optimal set (as is usual in an EMO application), but a single preferred solution on the Pareto-optimal set. The ideas suggested can be extended in a number of different ways, which we discuss in the following paragraphs.

- *Incomparable class and constrained problems:* In this study, we have not considered the case in which the DM judges some of the η points to be incomparable. Although our optimization problem formulation (equation 4) considers this situation, a study is needed to implement the idea and particularly analyzing the effect of δ_V in the development of the value function. Since some cases may occur, in which a value function satisfying all DM's preferences is not possible, this study will also test the specific part (Step 3) of the proposed PI-EMO algorithm.

  Moreover, we have not tested the constrained optimization problems in this study. The modified constrained domination principle can now be used and tested on some challenging problems.

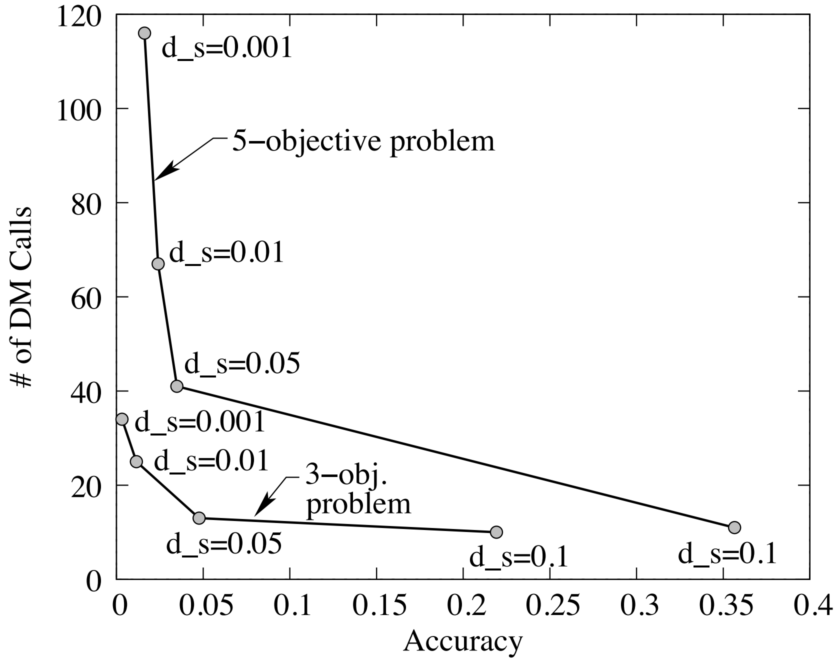

Figure 17: Trade-off between accuracy and the number of DM calls for the modified three and five-objective DTLZ problems.

Table 11: Distance of obtained solution from the most preferred solution, function evaluations, and the number of DM calls by PI-NSGA-II for the five-objective modified DTLZ2 problem with a stochastic DM-emulated value function.

	Minimum	Median	Maximum
Accuracy	0.0219	0.1137	0.2766
Func. Evals.	33,653	39,264	52,564
# of DM Calls	72	87	136

- *Other value functions:* In this study, we have restricted the value function to be of certain form (equation 4). Other more generic value function structures can also be considered. Our suggested value function construction procedure results in strictly increasing functions. However, a more generic non-concave value function may be obtained by using different conditions in the optimization problem formulation.

- *Robust value functions:* The optimization problem for deriving the value function can include a robustness consideration, in which the insensitivity of the value function coefficients in producing an identical ranking of η points can be ensured. This would be a different way of handling inconsistencies in decision-making.

- *Other termination conditions:* Our proposed PI-EMO algorithm terminates when there does not exist a far away point with a better value function value than the currently judged preferred point. Although this indicates somewhat the probability of creating better preferred points than the currently judged preferred point, other termination indicators are certainly possible. In this direction, instead of terminating based on Euclidean distance between the two points, the difference in value function values can be checked.

Table 12: Effect of stochastic factor on performance measures for three-objective modified DTLZ2 problem.

s	Incorrect ranking	Total DM Calls	Func. Evals.	Accuracy
1	12	20	6,786	0.0399
5	17	23	7,528	0.0437
10	19	28	8,572	0.0498
100	23	31	9,176	0.0512

- *Reduction in DM calls:* One outcome of the parametric study is that by fixing a relaxed termination criterion (relatively larger value of d_s), the number of DM calls can be reduced. However, there are other extensions to this study which may also reduce the number of DM calls. The basic operators in the suggested algorithm can be extended so that the modified procedure requires a reduced number of overall DM calls. The issue of having more points in each DM call, thereby reducing the overall number of DM calls to achieve a comparable accuracy will constitute an important study. Instead of keeping a fixed interval of τ generations for each DM call, DM call interval can be varied (or self-adapted) based on the extent of improvement achieved from the previous value function. Varying the number of points (η) in each DM call in a self-adaptive manner would be another important task. Since the points early on in the PI-EMO procedure are not expected to be close to the Pareto-optimal front, the number of DM calls and points per call can be made small. Thereafter, when the procedure approaches the Pareto-optimal front, more points can be included per DM call and the frequency of DM calls can be controlled by the observed rate of improvement of the performance of the procedure. Also, it would be an interesting study to ascertain the effect of cumulating the preference information from one decision call to the next and use it in approximating the value function.

- *Fixed budget of DM calls:* In this study, we have kept a termination criterion which is related to the extent of improvements in currently judged preferred solution. We then recorded the number of DM calls needed to achieve a limited extent of possible improvements (with the parameter d_s). However, a comparative study, in which different algorithms are compared for a fixed number of DM calls may be performed.

- *Value function based recombination and mutation operators:* In this study, we have modified the domination principles to emphasize points which have better value function value. However, EMO algorithm's recombination and mutation operators can also be modified based on developed value function. For example, restricting one of the top two currently judged preferred solutions as one parent in the SBX operator may help generate better preferred solutions.

- *PI-EMO with other EMO algorithms:* In this study, we have integrated the preference information in NSGA-II algorithm. A natural extension of this study would be to incorporate the preference handling approach with other popular EMO methodologies, such as SPEA2 [22], PESA [32], and others.

9 Conclusions

In this paper, we have suggested a simple-minded preference based evolutionary multi-objective optimization (PI-EMO) procedure, which iteratively finds new solutions by using an EMO algorithm and progressively sends a representative set of trade-off solutions to a DM for obtaining a complete or partial preference ranking. DM's preference information has been used in the following three ways in developing the new algorithm:

- Firstly, a strictly increasing value function is derived by solving an optimization problem, which maximizes the value function value between ranked points.
- Secondly, the resulting value function is then utilized to redefine the domination principle between the points. The modified domination principle is used to drive the EMO search.
- Thirdly, the resulting value function is used to constitute a termination criterion for the PI-EMO algorithm by executing a single-objective search along the gradient direction of the value function.

The above generic preference based EMO approach has been implemented with the NSGA-II procedure. The PI-NSGA-II procedure has then been applied to three different test-problems involving two, three and five objectives. By using a DM-emulated utility function, we have shown that the PI-NSGA-II is capable of finding the most preferred solution corresponding to the emulated utility function. A parametric study on the additional parameters has clearly indicated optimal parameter settings. Finally, to simulate the practical inconsistencies, which may arise in providing preference information was simulated by considering a stochastic value function with a noise effect reducing over time. Even in such cases, the PI-NSGA-II has been able to come closer to the most preferred point corresponding to the deterministic version of the DM-emulated value function.

Combining the ideas, from EMO algorithms and multiple criterion decision making (MCDM), seems an encouraging direction for future research in multi-objective optimization. In this paper, we have suggested one particular integration of DM's direct preference information into an EMO algorithm. The method is generic and the obtained results indicate that it is a promising approach. More emphasis must now be placed for developing pragmatic hybrid algorithms for multi-objective optimization and decision-making.

Acknowledgments

Kalyanmoy Deb wishes to thank the Academy of Finland and the Foundation of the Helsinki School of Economics for their support for this study under the FiDiPro programme. Pekka Korhonen and Jyrki Wallenius also wish to acknowledge the financial support from the Academy of Finland. (Grant No. 12198)

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] K. Deb and D. Saxena, "Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," in *Proceedings of the World Congress on Computational Intelligence (WCCI-2006)*, 2006, pp. 3352–3360.

[4] J. Knowles and D. Corne, "Quantifying the effects of objective space dimension in evolutionary multiobjective optimization," in *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO-2007)*, 2007, pp. 757–771, (LNCS 4403).

[5] J. Branke, T. Kaussler, and H. Schmeck, "Guidance in evolutionary multi-objective optimization," *Advances in Engineering Software*, vol. 32, pp. 499–507, 2001.

[6] J. Branke and K. Deb, "Integrating user preferences into evolutionary multi-objective optimization," in *Knowledge Incorporation in Evolutionary Computation*, Y. Jin, Ed. Heidelberg, Germany: Springer, 2004, pp. 461–477.

[7] K. Deb, J. Sundar, N. Uday, and S. Chaudhuri, “Reference point based multi-objective optimization using evolutionary algorithms,” *International Journal of Computational Intelligence Research (IJCIR)*, vol. 2, no. 6, pp. 273–286, 2006.

[8] L. Thiele, K. Miettinen, P. Korhonen, and J. Molina, “A preference-based interactive evolutionary algorithm for multiobjective optimization,” *Evolutionary Computation Journal*, in press.

[9] K. Deb and A. Kumar, “Interactive evolutionary multi-objective optimization and decision-making using reference direction method,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007)*. New York: The Association of Computing Machinery (ACM), 2007, pp. 781–788.

[10] ——, “Light beam search based multi-objective optimization using evolutionary algorithms,” in *Proceedings of the Congress on Evolutionary Computation (CEC-07)*, 2007, pp. 2125–2132.

[11] A. Jaszkiewicz and J. Branke, “Interactive multiobjective evolutionary algorithms,” in *Multiobjective optimization – Interactive and Evolutionary Approaches (LNCS volume 5252)*, J. Branke, K. Deb, K. Miettinen, and R. Slowinski, Eds. Heidelberg: Springer, 2008, pp. 179–193.

[12] P. Korhonen and J. Laakso, “A visual interactive method for solving the multiple criteria problem,” *European Journal of Operational Reseaech*, vol. 24, pp. 277–287, 1986.

[13] P. Korhonen and G. Y. Yu, “A reference direction approach to multiple objective quadratic-linear programming,” *European Journal of Operational Reseaech*, vol. 102, pp. 601–610, 1997.

[14] 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.

[15] 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.

[16] P. Korhonen, H. Moskowitz, P. Salminen, and J. Wallenius, “Further developments and tests of a progressive algorithm for multiple criteria decision making,” *Operations Research*, vol. 41, no. 6, pp. 1033–1045, 1993.

[17] 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.

[18] G. W. Greenwood, X. Hu, and J. G. D’Ambrosio, “Fitness functions for multiple objective optimization problems: Combining preferences with pareto rankings,” in *Foundations of Genetic Algorithms (FOGA)*. San Mateo: Morgan Kauffman, 1996, pp. 437–455.

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

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

[21] J. Figueira, S. Greco, and R. Slowinski, “Building a set of additive value functions representing a preference preorder and intensities of preference: GRIP method,” *European Journal of Operational Research*, vol. 195, no. 2, pp. 460–486, 2009.

[22] 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.

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

[24] R. L. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. New York: Wiley, 1976.

[25] 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.

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

[27] K. V. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin: Springer-Verlag, 2005.

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

[29] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation Journal*, vol. 8, no. 2, pp. 125–148, 2000.

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

[31] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychological Review*, vol. 63, no. 2, pp. 81–87, 1956.

[32] D. W. Corne, J. D. Knowles, and M. Oates, "The Pareto envelope-based selection algorithm for multiobjective optimization," in *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature VI (PPSN-VI)*, 2000, pp. 839–848.

[33] 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 *Submitted to European Journal of Operational Research*, 2009.

[34] S. Zionts and J. Wallenius "An interactive programming method for solving the multiple criteria problem," in *Management Science*, vol. 22, pp. 656–663, 1976.

[35] P. Korhonen "A Visual Reference Direction Approach to Solving Discrete Multiple Criteria Problems," in *European Journal of Operational Research*, vol. 34, no. 2, pp. 152–159 1988.

[36] A. Mas-Colell, M. D. Whinston and J. R. Green, *Microeconomic Theory*. New York: Oxford University Press, 1995.