Solving High Objective Problems in Fixed Interactions with the Decision Maker

Ankur Sinha, Anmol Pandey and Kalyanmoy Deb

KanGAL Report Number 2012004

***Abstract*— In recent advancements towards handling high objective optimization problems, it is proposed to progressively integrate the decision maker with the execution of an evolutionary multi-objective optimization algorithm. Preferences from the decision maker are accepted at the intermediate steps of the algorithm and a progress towards the most preferred point is made. In this paper, we extend the work on 'progressively interactive evolutionary multi-objective optimization using value function' (PI-EMO-VF) by allowing the optimization to be performed in a fixed number of interactions with the decision maker. In the PI-EMO-VF procedure, information is accepted from the decision maker, which is utilized by the evolutionary algorithm to perform a focused search in the region of interest. However, it is not possible to restrict the number of interactions required to handle an optimization problem. This paper contributes towards, solving the optimization problem in a pre-decided number of decision maker calls. Once the available budget of decision maker calls are known, it is optimally utilized to get close to the most preferred point on the Pareto-frontier. The paper evaluates the performance of the modified PI-EMO-VF algorithm on two, three and five objective test problems. A comparative study is performed against the previous proposal for the PI-EMO-VF procedure.**

***Index Terms*— Evolutionary multi-objective optimization algorithm, multiple criteria decision-making, interactive multi-objective optimization algorithm, value function, decision maker calls, preference based multi-objective optimization.**

I. Introduction

Arriving at the most preferred point in a high objective optimization problem typically comprises of a search and a decision making task. Based on the integration of search and decision making tasks, a variety of procedures can be developed. Progressively interactive procedure [4] is one such technique which involves seamless integration of the decision maker with the optimization run of an algorithm. Preference information from the decision maker is elicited at the intermediate steps of an evolutionary multi-objective algorithm (EMO), and progress towards the most preferred point is made. The efficacy of such a procedure in handling high objective problems with a relatively less number of function evaluations and a high accuracy has been shown recently [4], [10], [11]. In this paper, we consider the progressively interactive evolutionary multi-objective optimization using value function (PI-EMO-VF) procedure, which is a decision maker driven approach where the decision maker is in full control of the optimization algorithm. Although the procedure is able to successfully overcome the challenges faced by EMO algorithms in handling high objective problems, it suffers from a few drawbacks. The most important drawback being, an unknown number of decision maker calls, i.e. it is not known as to how many times a decision maker has to be called to provide preference information before the termination of the algorithm. This is a significant drawback as the time of a decision maker is usually important, and only a limited number of interactions can be performed. This paper aims at handling this drawback in the PI-EMO-VF procedure by performing the optimization in a fixed number of interactions with the decision maker. The maximum number of times a decision maker will be available for interaction with the algorithm is accepted as an additional input before the start of the optimization run.

It can be intuitively understood that an increase in the number of interactions with the decision maker will lead to a higher accuracy of the solution obtained. However, if the number of interactions to be performed with the decision maker is limited, it is not straight forward to design a methodology which utilizes the budget in an efficient manner. A designed procedure might be efficient for one class of problems, but might perform worse on others. The difficulty is due to the fact that before the start of the optimization run or even during the optimization run, it is not known as to how far the most preferred point is. This makes it difficult to decide the instances for the decision maker calls. To tackle this issue, we propose to acquire an initial idea about the distance of the most preferred point from the random population, and thereafter, utilize the decision maker calls in a judicious manner to approach the most preferred point. The idea is successfully incorporated in the PI-EMO-VF procedure leading to improvement in results.

In the next section of this paper we provide past studies performed in the direction of progressively interactive evolutionary multi-objective optimization algorithms. This is followed by a brief introduction to the value function fitting procedure to the preferences of the decision maker. Thereafter, we propose the modified PI-EMO-VF algorithm for handling high objective problems in a fixed number of interactions with the decision maker. Then, the results for

Ankur Sinha is a researcher at the Department of Information and Service Economy, Aalto University School of Economics, PO Box 21210, Aalto-00076, Helsinki, Finland (ankur.sinha@hse.fi)

Anmol Pandey is an undergraduate student at the Department of Mechanical Engineering, Indian Institute of Technology Kanpur, PIN 208016 , Kanpur, India (anmolp@iitk.ac.in)

Kalyanmoy Deb is a Professor at Indian Institute of Technology Kanpur, PIN 208016, India (deb@iitk.ac.in). He is also a Professor at Department of Information and Service Economy, Aalto University School of Economics, PO Box 21210, Aalto-00076, Helsinki, Finland (kalyanmoy.deb@hse.fi)

the performance of our procedure on three test problems involving two, three and five objectives are provided. These results have been contrasted with those obtained in the earlier study [4]. Finally, the paper ends with the conclusions.

II. Studies on progressively interactive methods

There exist only a few studies in the direction of progressive use of preference information in the intermediate steps of an evolutionary algorithms. Most of the studies from the recent past provide a sparse set of points to the decision maker, and elicit preference information on those set of points. The studies which could be found in the literature on progressively interactive methods are by Phelps and Köksalan [9], Fowler et al. [6], Jaszkiewicz [7], Branke et al. [1], Deb et al. [4] and Sinha et al. [11].

In this paper, we will focus on the approach proposed by Deb et al. [4] which relies on implicit construction of value functions based on the preferences provided by the decision maker. The preferences are mapped in a polynomial value function (other value function forms may also be used), which is used in the subsequent generations of the EMO for taking decisions. The value function gets modified based on each interaction with the decision maker. Most of the studies mentioned above, except the study by Sinha et al. [11], rely on value functions to make a progress towards the most preferred point. The study by Sinha et al. [11] utilizes polyhedral cones instead of value functions to search for the point most preferred for the decision maker.

III. Fitting value function to the preferences of the decision maker

Preferences of the decision maker are accepted by performing pair-wise comparisons among a given set of points. Whenever the decision maker compares a pair of points $\{P_i, P_j\}$, he would draw one of the following conclusions:

1) Prefers P_i over P_j
2) Prefers P_j over P_i
3) Is indifferent between the two points
4) Finds the points incomparable

The first two conclusions are easy to understand as they show a clear preference of the decision maker on one of the given points. However, it is important to understand and realize the difference between the other two conclusions, i.e. indifference and incomparability. From a behavioral perspective, the two conclusions are very different which we did not realize in our previous studies [4], [10]. A deeper understanding of the behavioral perspectives of a decision maker made us realize the difference between indifference and incomparability. When the decision maker is indifferent between two points, in terms of value function, it means that the values assigned to the points are so close that the utility of the two points are almost similar to the decision maker. However, when the decision maker finds the two points incomparable, it means that he does not have any answer to the question asked by us. This does not necessarily mean that the two points have similar values. Therefore, the third conclusion is a strong information which says that the values of two points are almost equal, whereas the fourth conclusion is no information at all.

Let us assume that a set of η points $\{P_1, P_2, \ldots, P_\eta\}$ are given to the decision maker for elicitation of preference information. The decision maker performs pair-wise comparisons which would lead to any of the following information:

1) Pair-wise comparisons between few points from the set
2) A small number of preferred points from the set
3) A small number rank ordered points from the set
4) Best point from the set

The value function construction procedure is able to incorporate any kind of information provided by the decision maker. However, in this study we expect the decision maker to provide us maximum information by rank-ordering the given set of η points. We do not consider indifference or incomparability in this study.

Now, we discuss the value function fitting procedure to the preferences of the decision maker. This requires solving an optimization problem to find the optimal parameters for the chosen form of the value function. A variety of value functions can be found in the literature like, constant elasticity of substitution (CES) [8], Cobb-Douglas [8] and polynomial [4], [10]. We consider here, the polynomial value function because of its flexibility to fit a larger class of quasi-concave value functions as compared to CES and Cobb-Douglas.

A. Value function optimization

The solution to the following optimization problem provides the optimal parameters to an assumed value function form. It is a generic approach which could be used to fit any value function form to the preferences provided by the decision maker.

$$\begin{array}{lll}
\text{Maximize} & \epsilon, & \\
\text{subject to} & \text{Value Function Constraints} & \text{(C1)} \\
& V(P_i) - V(P_j) \geq \epsilon, \quad \text{for all } (P_i, P_j) \text{ pairs} & \\
& \qquad\qquad \text{satisfying } P_i \succ P_j & \text{(C2)} \\
& |V(P_i) - V(P_j)| \leq \delta_V, \quad \text{for all } (P_i, P_j) \text{ pairs} & \\
& \qquad\qquad \text{satisfying } P_i \equiv P_j & \text{(C3)}
\end{array} \tag{1}$$

In the equations above, V represents the value function being used and P is a point which is a vector of objectives. Therefore, $V(P)$ is a vector to scalar mapping, such that every point P gets a value assigned to it. It represents the utility of the point to the decision maker. The higher the value, more preferred the point is to the decision maker. A closer look into the optimization problem would reveal that it attempts to find such parameters for the value function form for which the minimum difference in the value function values between pairs $P_i, P_j : P_i \succ P_j$ gets maximized.

The optimization problem contains three sets of constraints. The first set $(C1)$ ensures that the constraints related to the value function form are met, which includes variable bounds and other constraints related to the feasibility of the value function. The second set (C2) of constraints ensures that the preference statements for all the pairs where the decision maker has shown strict preference get satisfied. It makes the difference in the value function values for such pairs more than ϵ. The third set (C3) of constraints ensure that for any pair where the decision maker has shown indifference, the difference in the value function values is small. For such pairs of points, the absolute difference between their value function values is restricted within a small range (δ_V). The value function optimization task is considered successful only if a positive value of ϵ is obtained as a solution for the optimization problem. A non-positive value of ϵ means that all the preferences provided by the decision maker cannot be fitted by the assumed value function form.

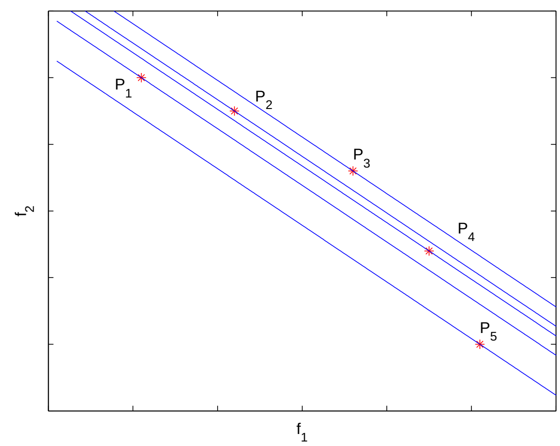

Fig. 1. Linear value function satisfying the decision maker preferences.

B. Polynomial value function

A generalized polynomial value function [10] has been stated in the following equation. The degree of the polynomial can be chosen based on the complexity of the preference information to be fitted. A single degree polynomial represents a linear value function.

$$
\begin{array}{lll}
\multicolumn{3}{l}{V(f_1, f_2, \ldots, f_M) = \prod_{j=1}^{p} \sum_{i=1}^{M} (\alpha_{ij} f_i + \beta_j)} \\
\multicolumn{3}{l}{\text{such that}} \\
\multicolumn{3}{l}{\quad \sum_{i=1}^{M} \alpha_{ij} = 1, \quad j = 1, \ldots, p} \\
\multicolumn{3}{l}{\quad S_j = \sum_{i=1}^{M} (\alpha_{ij} f_i + \beta_j) > 0, \quad j = 1, \ldots, p} \\
\multicolumn{3}{l}{\quad 0 \le \alpha_{ij} \le 1, \quad j = 1, \ldots, p} \\
\text{where} & f_i & \text{are the objective functions} \\
 & \alpha_{ij}, \beta_i, p & \text{are the value function parameters} \\
\text{and} & S_j & \text{are the linear product terms in the value function}
\end{array}
\tag{2}
$$

We provide a simple example to understand the fitting procedure on a problem with five points. Let us assume that the given points are non-dominated and the preference information received is in the form of pair-wise information between some of the points. Figure 1 shows a two criteria problem where five points $(P_1, \ldots, P_5)$ are given to the decision maker for preference information. We obtain preference information in the form of pair-wise comparisons between 3 pairs ($P_2 \succ P_1$, $P_3 \succ P_4$, $P_4 \succ P_5$). We begin by assuming the degree of the polynomial value function as 1 and try to fit the preferences. Solving the following optimization problem leads to a positive value for ϵ which suggests that a linear value function is able to fit the preference structure of the decision maker.

$$
\begin{array}{l}
\text{Maximize } \epsilon \\
\quad \text{subject to} \\
\quad \alpha_{11} + \alpha_{12} = 1 \\
\quad S_1 = \alpha_{11} f_1 + \alpha_{12} f_2 + \beta_1 > 0 \\
\quad \alpha_{11} > 0 \quad \alpha_{12} > 0 \\
\quad V(P_2) - V(P_1) \ge \epsilon \\
\quad V(P_3) - V(P_4) \ge \epsilon \\
\quad V(P_4) - V(P_5) \ge \epsilon \\
\quad \text{where} \\
\qquad P = \{f_1, f_2\} \\
\qquad V(P) = \alpha_{11} f_1 + \alpha_{12} f_2 + \beta_1
\end{array}
\tag{3}
$$

The second constraint and the variable β_1 are not necessary for a linear value function as it only ensures positive values for all the points. However, for higher degree polynomials it is important as it ensures quasi-concavity.

IV. PI-EMO-VF PROCEDURE IN FIXED INTERACTIONS WITH THE DECISION MAKER

A progressively interactive EMO [4] was proposed by Deb, Sinha, Korhonen and Wallenius where a value function is implicitly constructed after every few generations of the EMO based on the preference information received from the decision maker. A standard EMO algorithm (NSGA-II [3]) was combined with ideas from multiple criteria decision making (MCDM) field to implement the procedure. The decision maker's preference information is utilized periodically in the intermediate generations of the EMO which directs the search in a preferred region of search space. The decision maker is invited after every few generations of the EMO and preference elicitation is performed by asking the decision maker to rank-order a set of η sparse points in the search space. The decision maker is expected to provide a complete or partial information about the superiority or indifference of one solution over the other. Based on this information a value function is constructed which maps the preference structure of the decision maker. The constructed value function is used by the EMO to make decisions in the subsequent generations.

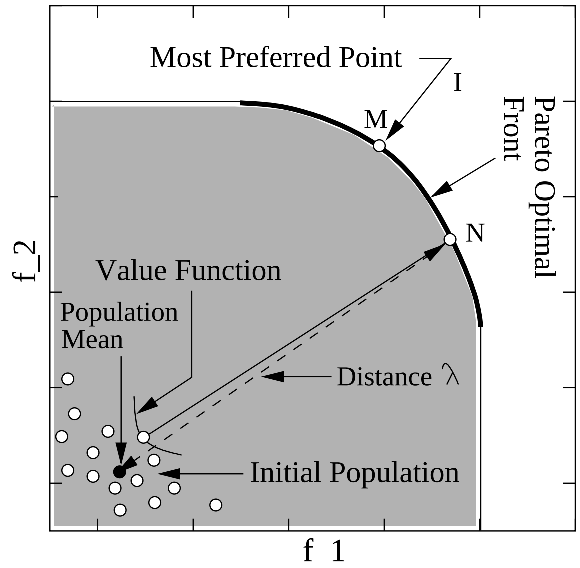

Fig. 2. Initial interaction with the decision maker to determine an approximate distance of the most preferred point from the mean of the initial population members.

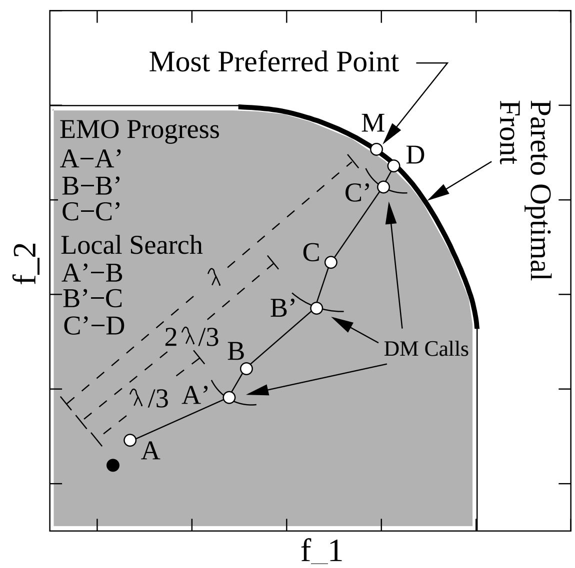

Fig. 3. Progress of the modified PI-EMO-VF procedure towards the most preferred point showing the instances of decision maker calls.

As already mentioned, there is no control in the algorithm as to how many times a decision maker will have to be invited for preference elicitation before the termination of the procedure. In the modified version of the procedure proposed in this paper, we eliminate some parameters, and terminate the optimization procedure in a fixed number of decision maker calls (say Ω).

To accomplish the optimization task in a fixed number of decision maker calls, we need to decide when to invite the decision maker for providing preferences. However, without any information about the problem, it is difficult to identify the instances when a decision maker should be invited. Frequent elicitations will consume the budget of decision maker calls with the algorithm still being far away from the frontier. Few elicitations will make the algorithm reach the frontier even before a large number of decision maker calls could be utilized. In order to handle this issue we attempt to obtain an approximate idea about the distance of the most preferred point from the initial population of the PI-EMO-VF procedure. We propose to elicit preference information on some points of the initial random population and generate a value function for the decision maker. A single objective search is performed along the gradient of the value function until termination on one of the points on the Pareto-optimal frontier. The obtained point is assumed to be a proxy for the most preferred point and its distance from the mean of the initial population is computed. We assume this distance to be an approximate distance to the true most preferred point. This has been shown in Figure 2, where a value function is constructed based on preference elicitation on few points in the initial population. Along the gradient of the value function, a single objective [1] search is performed which terminates at the point N. Though the actual most preferred point is M, distance (say λ) between the population mean and the point N gives us an approximate idea about the distance of the actual most preferred point from the population mean. Once this distance is obtained, we are able to make a decision maker call after every small improvement as shown in Figure 3. One decision maker call gets consumed in finding an approximate distance of the most preferred point, and we have $\Omega - 1$ decision maker calls available. We wish to use 75% of the decision maker calls in moving inside the dominated region and remaining 25% of the calls for search on the frontier. For this, we reserve 25% of the remaining decision maker calls and use $\Omega' = \lfloor (0.75(\Omega - 1)) \rfloor$ decision maker calls for improvement in the dominated region. In our implementation we call the decision maker after making an improvement of $\frac{\lambda}{\Omega'}, \frac{2\lambda}{\Omega'}, \frac{3\lambda}{\Omega'}, \ldots, \lambda$ in terms of distance from the mean of the initial population. In the example shown in Figures 2 and 3, the value of Ω is 6 and the first decision maker call is made in Figure 2. Now $\Omega' = 3$, therefore 3 calls are made for improvement in the dominated region as shown in Figure 3. The remaining 2 decision maker calls are reserved for search on the frontier. The distance λ is determined and then the decision maker is invited after improvements corresponding to $\frac{\lambda}{3}, \frac{2\lambda}{3}$ and λ from the mean of the initial population. Local search (discussed later) is performed until it makes an improvement of a requisite distance (discussed later) from the best point along the direction of the gradient.

[1]We convert the problem into a single objective optimization task by taking the best point as reference point and the gradient direction as reference direction. An achievement scalarization function [12] is formulated which is used to direct the search towards a single point on the frontier.

Local search helps the algorithm to quickly move towards the front. In the figure we have not shown the search performed on the frontier. There can be a few intricacies involved in implementing the procedure which have been clarified in the step-wise procedure for the algorithm.

In the following, we provide a step-by-step procedure for the modified PI-EMO-VF algorithm:

Step 1: Initialize a population and set an iteration counter $\omega = 0$. Set the maximum allowed decision maker calls as Ω.

Step 2: Cluster the current non-dominated front to choose η widely distributed points. If η is greater than number of non-dominated points in the current population then choose all the points, and cluster the second non-dominated front to choose the remaining point. If the total number of points in the two fronts are less than η then move to the third frontier to choose the remaining points. Repeat the process until η points are chosen.

Step 3: Obtain decision-maker's preferences on η points. Construct a value function $V(\mathbf{f})$ from this information by solving the value function optimization problem (VFOP).

Step 4: Obtain value function values for all the non-dominated points. Let the best solution be P^0_{best}. Perform local search using the achievement scalarization function with P^0_{best} as the reference point and gradient of the value function as the reference direction. Let the local search terminate with the KKT criteria getting satisfied. KKT criteria being met ensures convergence to a point P_f on the Pareto-front. If the mean of the initial population members is P^0_{mean}, then λ is obtained by computing the Euclidean distance between P^0_{mean} and P_f.

Step 5: Increment $\omega = \omega + 1$ and execute the EMO run with a value function based modified domination principle [4] and the genetic operators. Keep track of the best point (with respect to the most recent value function) and stop the execution if the Euclidean distance between P^{ω}_{best} and P^0_{mean} is greater than $\omega\frac{\lambda}{\Omega'}$, where $\Omega' = \lfloor(0.75(\Omega-1))\rfloor$. Let the number of generations required by the EMO to achieve the improvement be g^{ω}_{req}. If the EMO is not able to improve the best point by the required distance, then stop the execution with maximum number of generations $g^{\omega}_{max} = 2g^{\omega-i}_{req}$, where i is the largest integer for which EMO execution stopped with the distance criteria. To begin with, $2g^0_{req}$ may be set to a large value.

Step 6: If $P^{\omega-1}_{best}$ is not-dominated by the current non-dominated set then choose it, and choose $\eta-1$ additional points by clustering from the current non-dominated set. Otherwise, if $P^{\omega-1}_{best}$ is dominated then choose all η points from the current non-dominated set. If η is greater than number of non-dominated points then repeat the process in a similar manner as mentioned in Step 2 to choose η points.

Step 7: Invite the decision maker to provide his preferences and construct a value function by solving the value function optimization problem.

Step 8: Perform local search with the best point P^{ω}_{best} as reference point and gradient of the value function as reference direction. If $\omega < \Omega - 1$ then check for termination at each iteration of local search, and terminate if the point produced by local search is more than $\omega\frac{\lambda}{\Omega'} + d_s$ (In the modified procedure we fix the parameter d_s as $0.1\frac{\lambda}{\Omega'}$) distance away from P^0_{mean} or the KKT conditions get satisfied. If $\omega == \Omega - 1$ then terminate when the KKT conditions are satisfied. The point produced by local search replaces P^{ω}_{best} in the population.

Step 9: If $\omega = \Omega - 1$, then terminate the PI-EMO-VF procedure, otherwise go to Step 5.

The modified PI-EMO-VF algorithm expects the user to set a value for Ω and η other than the commonly used evolutionary algorithm parameters. The k-mean clustering algorithm [2], [13] is used to select η well-diversified points in the objective space. The value function optimization problem is solved using the *fmincon* code of MATLAB with a limit on the maximum number of function evaluations as 250M, where M is the number of objectives. A KKT tolerance of 10^{-5} is used.

1) Local search: A single objective search along the gradient of the value function will lead us to solutions with higher values for the chosen value function. We utilize this gradient information into the achievement scalarization function [12], where the gradient is used as the reference direction and the current best P^{ω}_{best} as the reference point $\mathbf{z}^b$. Following is the formulation of the single objective achievement scalarization function:

$$\begin{array}{ll} \text{Maximize} & \left(\min\limits_{i=1}^{M} \frac{f_i(\mathbf{x})-z_i^b}{\frac{\partial V}{\partial f_i}}\right) + \rho \sum_{j=1}^{M} \frac{f_j(\mathbf{x})-z_j^b}{\frac{\partial V}{\partial f_j}}. \\ \text{subject to} & \mathbf{x} \in \mathcal{S}. \end{array} \quad (4)$$

In the above formulation, $\mathcal{S}$ denotes the feasible region for the original problem. A small weight of $\rho = 10^{-8}$ is used for the second term, which prevents the solution from converging to a weak Pareto-optimal solution. Any single-objective optimization method can be used to solve the above problem. The modified PI-EMO-VF procedure has been implemented on MATLAB, therefore we use the *fmincon* routine to optimize the achievement scalarization function.

V. RESULTS

In this section, we present the results of the modified PI-EMO-VF procedure on two, three, and five objective test problems. Modified ZDT1 and DTLZ2 test problems described in [4] are used for evaluating the procedure. The section constitutes of three sub-sections, where we describe the parameter values used, present the performance of modified PI-EMO-VF on test problems, and compare the modified PI-EMO-Vf procedure with its older version [4]:

A. Parameter setting

We implement the modified PI-EMO-VF procedure on the NSGA-II framework. Apart from the parameters of the NSGA-II algorithm, there are two additional parameters Ω and η. There are other parameters like ρ and KKT error limit for terminating the single objective optimization based local search. However, these parameters are not significant to the performance of the algorithm. Two parameters have been eliminated from the previous version of the PI-EMO-VF procedure which are τ and d_s. Parameter d_s is still used in the algorithm, but is adjusted adaptively. Following is the parameter setting used for the modified PI-EMO-VF procedure.

1) Number of points for preference elicitation: $\eta = 5$.
2) Maximum number of DM interactions: $\Omega = 1, 5, 10, 15, 20$.
3) Crossover probability: $p_c = 0.9$
4) Distribution index for the SBX operator: $\eta_c = 15$.
5) Mutation probability: $p_m = 1/N$
6) Distribution index for the polynomial mutation operator: $\eta_c = 20$.
7) Population size: $N = 10M$, where M denotes the number of objectives.

B. Performance evaluation

The performance of the modified PI-EMO-VF procedure has been analyzed on three test problems in this sub section. An improvement in the accuracy achieved by the procedure with an increase in the number of decision maker calls Ω. 21 runs have been performed for each of the cases and the best, median and worst results are reported.

1) Two-objective test problem: Problem 1 is a two objective modified ZDT1 maximization problem with 30 variables [14].

$$\begin{array}{rl} \text{Maximize} & \mathbf{f}(\mathbf{x}) = \left\{ \begin{array}{c} x_1 \\ \frac{10-\sqrt{x_1 g(\mathbf{x})}}{g(\mathbf{x})} \end{array} \right\}, \\ \text{where} & g(\mathbf{x}) = 1 + \frac{9}{29}\sum_{i=2}^{30} x_i, \\ & 0 \leq x_i \leq 1, \quad \text{for } i = 1, 2, \ldots, 30, \end{array} \tag{5}$$

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

Following non-linear value function, which acts as a DM in providing a complete ranking of η solutions, is used.

$$V(f_1, f_2) = \frac{1}{(f_1 - 0.35)^2 + (f_2 - 9.6)^2}. \tag{6}$$

The above value function gives the most preferred solution as $\mathbf{z}^* = (0.25, 9.50)$.

Table I shows the best, median and worst function evaluations from 21 different PI-EMO-VF simulations for $\Omega = 10$, and Table II shows the accuracy for $\Omega = 10$ and 20. The performance (accuracy measure) is computed based on the Euclidean distance of the final solution from the most preferred point. Figure 4 shows the improvement in accuracy with increasing number of decision maker calls Ω. It can be seen that as Ω increases, the algorithm approaches towards the most preferred solution.

2) Three-objective test problem: The DTLZ2 [5] is a scalable test problem in terms of the number of objectives. For this test problem, the points in the objective space are bounded by two spherical surfaces in the first octant.

TABLE I

FUNCTION EVALUATIONS (FE) REQUIRED BY THE MODIFIED PI-EMO-VF APPROACH FOR TWO OBJECTIVE MODIFIED ZDT1 PROBLEM.

$\Omega = 10$	Best	Median	Worst
EMO Search FE	1190	2010	5650
Local Search FE	768	1056	3520
Total FE	1958	3066	9170

TABLE II

ACCURACY ACHIEVED BY THE MODIFIED PI-EMO-VF APPROACH FOR TWO OBJECTIVE MODIFIED ZDT1 PROBLEM.

	Best	Median	Worst
$\Omega = 10$, Accuracy	0.020174	0.188016	0.535933
$\Omega = 20$, Accuracy	0.000069	0.040225	0.207804

For the case where all the objectives are maximized, the outer spherical surface becomes the Pareto-optimal front. M-objective DTLZ2 problem for maximization is defined as follows:

$$\begin{array}{l} \text{Maximize} \quad \mathbf{f}(\mathbf{x}) = \\ \left\{ \begin{array}{l} (1.0 + g(\mathbf{x}))\cos(\frac{\pi}{2}x_1)\cos(\frac{\pi}{2}x_2)\cdots\cos(\frac{\pi}{2}x_{M-1}) \\ (1.0 + g(\mathbf{x}))\cos(\frac{\pi}{2}x_1)\cos(\frac{\pi}{2}x_2)\cdots\sin(\frac{\pi}{2}x_{M-1}) \\ \vdots \\ (1.0 + g(\mathbf{x}))\cos(\frac{\pi}{2}x_1)\sin(\frac{\pi}{2}x_2) \\ (1.0 + g(\mathbf{x}))\sin(\frac{\pi}{2}x_1) \end{array} \right\}, \\ \text{subject to} \quad 0 \leq x_i \leq 1, \quad \text{for } i = 1, \ldots, 12, \\ \quad \text{where} \quad g(\mathbf{x}) = \sum_{i=3}^{12}(x_i - 0.5)^2. \end{array} \tag{7}$$

The objectives vectors on the Pareto-optimal front follow the relation: $f_1^2 + f_2^2 + f_3^2 = 3.5^2$. The decision variables correspond to $x_1 \in [0, 1]$, $x_2 \in [0, 1]$ and $x_i = 0$ or 1 for $i = 3, 4, \ldots, 12$ on the frontier.

A real decision maker has been replaced by the following

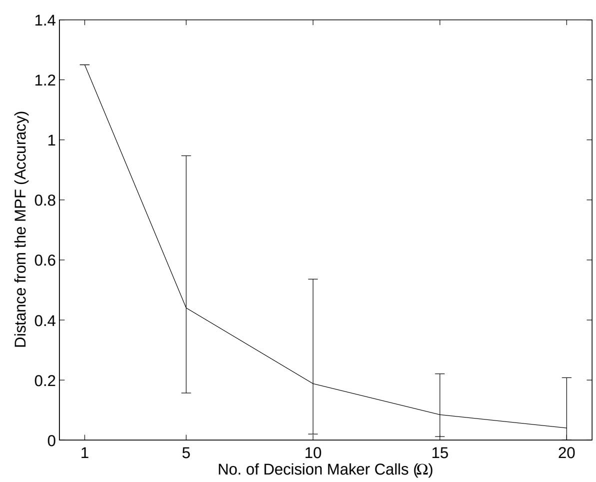

Fig. 4. Improvement in accuracy with increasing decision maker calls for two objective modified ZDT1.

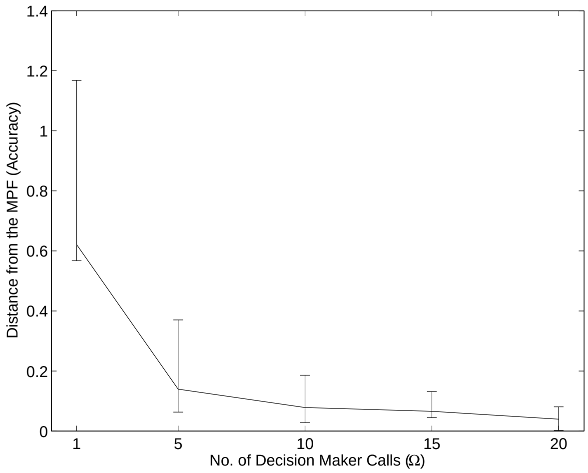

Fig. 5. Improvement in accuracy with increasing decision maker calls for three objective modified DTLZ2.

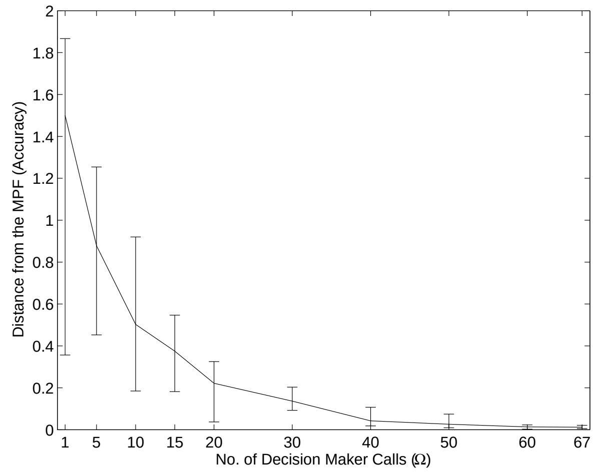

Fig. 6. Improvement in accuracy with increasing decision maker calls for five objective modified DTLZ2.

value function:

$$V(f_1, f_2, f_3) = 1.25f_1 + 1.50f_2 + 2.9047f_3. \quad (8)$$

The value function produces the most preferred solution on the Pareto-optimal front as $(1.25, 1.50, 2.9047)$.

Table III shows the best, median and worst function evaluations from 21 different PI-EMO-VF simulations for $\Omega = 10$, and Table IV shows the accuracy for $\Omega = 10$ and 20. Figure 5 shows the improvement in accuracy with increasing number of decision maker calls Ω. For this problem as well we observe that with an increase in Ω, the algorithm approaches towards the most preferred solution.

TABLE III

FUNCTION EVALUATIONS (FE) REQUIRED BY THE MODIFIED PI-EMO-VF APPROACH FOR THREE OBJECTIVE MODIFIED DTLZ2 PROBLEM.

$\Omega = 10$	Best	Median	Worst
EMO Search FE	1210	1990	3490
Local Search FE	420	772	1089
Total FE	1630	2762	4579

TABLE IV

ACCURACY ACHIEVED BY THE MODIFIED PI-EMO-VF APPROACH FOR THREE OBJECTIVE MODIFIED DTLZ2 PROBLEM.

	Best	Median	Worst
$\Omega = 10$, Accuracy	0.027705	0.180681	0.477523
$\Omega = 20$, Accuracy	0.002317	0.062025	0.266626

3) Five-objective test problem: Next, we consider the five-objective ($M = 5$) version of the DTLZ2 problem described previously. The Pareto-optimal front for the five objective version is given as $f_1^2 + f_2^2 + f_3^2 + f_4^2 + f_5^2 = 3.5^2$.

A non-linear DM-emulated value function has been chosen for this test problem, which is defined as follows:

$$V(\mathbf{f}) = 1/\sum_{i=1}^{5}(f_i - a_i)^2, \quad (9)$$

where $a = (1.1, 1.21, 1.43, 1.76, 2.6468)^T$. The most preferred point corresponding to this value function is $(1.0, 1.1, 1.3, 1.6, 2.4062)$.

Table V shows the best, median and worst function evaluations from 21 different PI-EMO-VF simulations for $\Omega = 10$, and Table VI shows the accuracy for $\Omega = 10$ and 20. Figure 4 shows the improvement in accuracy with increasing number of decision maker calls Ω. Once again we observe that as Ω increases, the algorithm approaches towards the most preferred solution.

TABLE V

FUNCTION EVALUATIONS (FE) REQUIRED BY THE MODIFIED PI-EMO-VF APPROACH FOR THREE OBJECTIVE MODIFIED DTLZ2 PROBLEM.

$\Omega = 10$	Best	Median	Worst
EMO Search FE	1610	2610	12310
Local Search FE	561	989	1579
Total FE	2171	3599	13889

TABLE VI

ACCURACY ACHIEVED BY THE MODIFIED PI-EMO-VF APPROACH FOR FIVE OBJECTIVE MODIFIED DTLZ2 PROBLEM.

	Best	Median	Worst
$\Omega = 10$, Accuracy	0.184899	0.502596	0.920110
$\Omega = 20$, Accuracy	0.037263	0.221858	0.325483

This study shows the trade-off between the decision maker calls Ω and the accuracy achieved. If a high accuracy is desired, the number of decision maker calls should be kept high. The algorithm is flexible to allow the decision maker to continue optimization even after termination, if he desires even better solution. It should be noted that for a large Ω the algorithm moves in smaller steps taking frequent information from the decision maker which leads to a higher accuracy. However, there can be other strategies where the algorithm moves with larger steps in the beginning and smaller steps towards the end. These strategies are not investigated yet and has been left for a future study.

C. Comparison study

In this sub-section, we provide a comparison of the modified PI-EMO-VF procedure against the previous version [4] of the procedure. The modified five-objective DTLZ-2 maximization problem has been chosen for comparison. The results for the previous version [4] of the PI-EMO-VF procedure is given in Table VII. The median number of decision maker calls required by the previous version of the PI-EMO-VF procedure to solve the problem is 67. For the modified version, we fix the number of decision maker calls as 67 and record the accuracy achieved and the number of function evaluations required. The results for the modified version of the algorithm are presented in Table VIII. It can be seen that the modified version offers a small improvement in accuracy along with a reduction in number of function evaluations. However, we would like to emphasize that the improvement of the modified algorithm should be seen more in terms of its ability to solve the problem to optimum in a fixed number of decision maker calls.

Figure 6 shows the performance of the modified PI-EMO-VF procedure on five objective DTLZ2 test problem for $\Omega = 1, 5, 10, 15, 20, 30, 40, 50, 60$ and 67 decision maker calls. At $\Omega = 60$ itself we obtain an accuracy which is better than the previous version the PI-EMO-VF procedure for 67 decision maker calls.

TABLE VII

DISTANCE OF OBTAINED SOLUTION FROM THE MOST PREFERRED POINT, FUNCTION EVALUATIONS (FE), AND THE NUMBER OF DM CALLS REQUIRED BY PI-NSGA-II-VF FOR THE FIVE-OBJECTIVE MODIFIED DTLZ2 PROBLEM USING THE PREVIOUS VERSION OF PI-EMO-VF [4].

	Best	Median	Worst
Accuracy	0.0084	0.0240	0.0902
Total FE	23,126	27,202	41,871
No. of DM Calls	57	67	102

TABLE VIII

DISTANCE OF OBTAINED SOLUTION FROM THE MOST PREFERRED POINT AND FUNCTION EVALUATIONS (FE) FOR FIXED NUMBER OF DM CALLS ($\Omega = 67$) FOR FIVE-OBJECTIVE MODIFIED DTLZ2 PROBLEM USING THE MODIFIED VERSION OF PI-EMO-VF.

	Best	Median	Worst
Accuracy	0.0054	0.0121	0.0211
Total FE	18034	21793	26238

VI. CONCLUSIONS

In this paper, we have described a technique to reach close to the most preferred solution in a fixed number of decision maker interactions. This is an improvement incorporated in a recently suggested PI-EMO-VF procedure. Using the modified procedure, problems with high objectives have been solved in 1,5, 10, 15 and 20 decision maker calls. A trade-off between number of decision maker calls and accuracy can be clearly observed from the graphs presented in the paper. The paper shows the efficacy of PI-EMO-VF procedure in handling high objective problems, and eliminates one of the significant drawbacks by allowing to fix the number of interactions with decision maker. A number of other possible improvements which are not a part of this study, but could be incorporated in the future studies are, trying other strategies to invite the decision maker during the optimization run, and cumulation of information from previous decision maker interactions.

REFERENCES

[1] 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), publisher=Berlin: Springer-Verlag, year=2009, pages=554–568*.

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

[3] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[4] K. Deb, A. Sinha, P. Korhonen, and J. Wallenius. An interactive evolutionary multi-objective optimization method based on progressively approximated value functions. *IEEE Transactions on Evolutionary Computation*, 14(5):723–739, 2010.

[5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization*, pages 105–145. London: Springer-Verlag, 2005.

[6] 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. *European Journal of Operational Research*, 206(2):417–425, 2009.

[7] A. Jaszkiewicz. Interactive multiobjective optimization with the pareto memetic algorithm. *Foundations of Computing and Decision Sciences*, 32(1):15–32, 2007.

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

[9] S. Phelps and M. Koksalan. An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. *Management Science*, 49(12):1726–1738, December 2003.

[10] A. Sinha, K. Deb, P. Korhonen, and J. Wallenius. Progressively interactive evolutionary multi-objective optimization method using generalized polynomial value functions. In *2010 IEEE Congress on Evolutionary Computation (CEC-2010)*, pages 1–8. IEEE Press, 2010.

[11] A. Sinha, P. Korhonen, J. Wallenius, and K. Deb. An interactive evolutionary multi-objective optimization method based on polyhedral cones. In *2010 Learning and Intelligent Optimization (LION-2010)*, pages 318–332. IEEE Press, 2010.

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

[13] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100,

Athens, Greece, 2001. International Center for Numerical Methods in Engineering (Cmine).

[14] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation Journal*, 8(2):125–148, 2000.