Using a Distance Metric to Guide PSO Algorithms for Many-Objective Optimization

Upali K Wickramasinghe, Xiaodong Li
School of Computer Science and Information Technology
RMIT University
Melbourne, VIC 3001, Australia
{uwickram, xiaodong}@cs.rmit.edu.au

ABSTRACT

In this paper we propose to use a distance metric based on user-preferences to efficiently find solutions for many-objective problems. We use a particle swarm optimization (PSO) algorithm as a baseline to demonstrate the usefulness of this distance metric, though the metric can be used in conjunction with any evolutionary multi-objective (EMO) algorithm. Existing user-preference based EMO algorithms rely on the use of dominance comparisons to explore the search-space. Unfortunately, this is ineffective and computationally expensive for many-objective problems. In the proposed distance metric based PSO, particles update their positions and velocities according to their closeness to preferred regions in the objective-space, as specified by the decision maker. The proposed distance metric allows an EMO algorithm's search to be more effective especially for many-objective problems, and to be more focused on the preferred regions, saving substantial computational cost. We demonstrate how to use a distance metric with two user-preference based PSO algorithms, which implement the reference point and light beam search methods. These algorithms are compared to a user-preference based PSO algorithm relying on the conventional dominance comparisons. Experimental results suggest that the distance metric based algorithms are more effective and efficient especially for difficult many-objective problems.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms

Keywords

Particle swarm optimization, Many-objective optimization, Multi-objective optimization, User-preference methods, Reference point method, Light beam search

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

1. INTRODUCTION

Integrating user-preferences in Evolutionary Multi-objective Optimization (EMO) algorithms has been gathering interest during the recent years [3, 4, 5, 22]. These preference mechanisms were seen originally in Multi-Criteria Decision Making (MCDM) literature [8]. These classical approaches have been integrated into Evolutionary Algorithms (EA) as search strategies to efficiently find preferred solutions in multi-objective problem instances. In recent studies user-preference methods have been applied to both Genetic Algorithms (GA) [3, 4, 5] and Particle Swarm Optimization (PSO) [22] algorithms. In all of these studies the concept of dominance plays a major role in the functionality of the algorithms. In many-objective optimization problems (where the number of objectives are greater than three), comparing individuals using Pareto dominance becomes less effective [12, 13, 14, 15]. Theoretical results in [13] shows that in many-objective search-spaces the number of non-dominated individuals increases to a point where the entire population becomes non-dominated to each other. This severely limits an algorithm's ability to compare and search for solutions in many-objective problems. To combat this problem, in this paper we propose to use a distance metric (rather than dominance comparisons) to guide an EMO algorithm to move towards the preferred region of the objective space.

In user-preference based EMO algorithms, a Decision Maker (DM) is required to first indicate preferred regions of the objective-space for an algorithm to find solutions in. This information is extremely valuable and can be used to guide the EMO algorithm to further explore the search-space. In the multi-objective PSO algorithms described in this paper, we use a distance metric to measure the closeness of each particle to the preferred regions. Particles will move in the search-space towards these preferred regions, updating their velocities and positions according to this distance metric. Once a particle has changed its position it will be evaluated to see how close it is to the preferred regions. In PSO, each particle has a *memory* of the best position it has visited so far [11]. Updating each particle's memory using the distance metric and following particles that are close to the preferred regions gives the necessary selection pressure to allow the population to converge towards the Pareto-front near the preferred regions.

To demonstrate the effectiveness of the proposed distance metric, in this paper we use two user-preference methods, one being the reference point method [18] and the other being the light beam search [10]. The distance metric is incorporated into an existing multi-objective PSO, the Multi-objective Differential Evolution and PSO (MDEPSO) algorithm [21]. We used MDEPSO because it has been shown to be effective in finding solutions in difficult multi-modal (having numerous local Pareto fronts and one true global Pareto front) multi-objective problem instances. The dominance comparisons used in MDEPSO will be replaced by the guiding mechanism provided by the distance metric. The proposed distance metric is less computationally expensive, compared with the dominance based methods. Furthermore, the two EMO algorithms using this distance metric are shown to perform well on several difficult multi-modal many-objective problems. In contrast, the dominance-comparison based approaches perform poorly on these same problems.

This paper is organized as follows. Section 2 briefly describes the MDEPSO algorithm, which is followed by descriptions of the reference point method and light beam search. Section 3 presents an overview on the related work carried out in the field of user-preference based EMO algorithms and many-objective optimization methods. Section 4 describes the distance metric and its incorporation into MDEPSO. Experiments used to evaluate the algorithm are provided in section 5. Finally section 6 presents our conclusions and avenues for future research

2. BACKGROUND

The MDEPSO algorithm and the main building blocks of user-preference mechanisms are described in the following sections.

2.1 MDEPSO algorithm

The MDEPSO algorithm [21] is a hybrid Differential Evolution (DE) and PSO multi-objective algorithm. In PSO algorithms, individuals (usually known as particles) update their velocities and positions with respect to a known *global best's* or *leader's* position ($\vec{p}_g$) and *personal best* position ($\vec{p}_i$). The *Constriction Type 1"* PSO version used in MDEPSO updates a particle's velocity ($\vec{v}_i$) and position ($\vec{x}_i$) at time t to $t+1$ according to the following two equations:

$$\vec{v}_i(t+1) = \chi(\vec{v}_i(t) + \phi_1(\vec{p}_i(t) - \vec{x}_i(t)) + \phi_2(\vec{p}_g(t) - \vec{x}_i(t))) \quad (1)$$

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t) \quad (2)$$

where ϕ_1 and ϕ_2 are random numbers generated uniformly between $[0, \frac{\varphi}{2}]$. φ is a constant equal to 4.1 [1]. χ is the so called *constriction factor*, which is used to prevent a particle from exploring too far into the search-space. χ is normally set to 0.7298, which is calculated according to $\frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|}$ [1].

In order to maintain a better diversity (thereby reducing the chance of getting stuck on local fronts), leaders are generated by using a DE scheme, more specifically the **DE/rand/1/bin** scheme. For a particle $\vec{x}_i$, a leader $\vec{u}_i$ is generated using three other individuals $\vec{x}_{r1}, \vec{x}_{r2}, \vec{x}_{r3}$ from the population such that $i \neq r1 \neq r2 \neq r3$. The j^{th} decision variable of the leader $\vec{u}_i$ is generated using (3).

$$\vec{u}_i = u_{j,i} = \begin{cases} x_{j,r1} + F(x_{j,r2} - x_{j,r3}) \\ \quad if \ \ (rand_j < CR \ \ or \ \ j = j_{rand}) \\ x_{j,i} \ \ otherwise \end{cases} \quad (3)$$

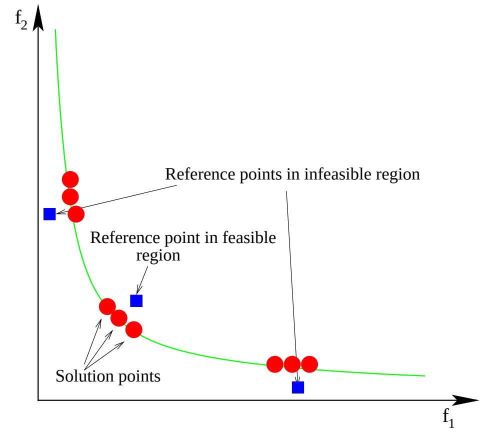

Figure 1: Reference point method

where $j_{rand} \in [1, D]$, and D is the number of dimensions in the search-space. $F \in [0, 1]$ is a scaling factor. CR is the crossover ratio and $rand_j$ is a random number generated uniformly between $[0, 1]$. The values used in MDEPSO were $CR = 0.2$ and $F = 0.4$ [21].

The original MDEPSO algorithm [21] relies on *dominance comparisons* to choose individuals to move to the next iteration. In this paper, we replace this dominance comparison scheme with the proposed distance metric (see (9) and (10)).

2.2 The reference point method

The classical reference point method was first described by Wierzbicki [5, 18]. It has been included successfully in several EMO algorithms [5, 22]. A reference point $\vec{\bar{z}}$ for a multi-objective problem consists of *aspiration values* for each objective. In the classical MCDM literature this reference point is used to construct a single objective function (given by (4)), which is to be minimized over the entire search-space. If $\vec{x}$ is a solution in the search-space,

$$minimize \quad \max_{i=0,\ldots,M-1} \{w_i(f_i(\vec{x}) - \bar{z}_i)\} \quad (4)$$

where $\vec{\bar{z}} = [\bar{z}_0, \ldots, \bar{z}_{M-1}]$ is the reference point and $\vec{w} = [w_0, \ldots, w_{M-1}]$ is a set of weights. f_i is the i^{th} objective function, while M denotes the number of objectives. The DM can assign values for weights, which represent any bias towards that objective.

Figure 1 illustrates the classical reference point method in a two-objective space. The DM indicates (to the algorithm) his/her preferred regions in the objective-space with the use of *reference points*. Then the algorithm is expected to concentrate on the regions around the reference points and obtain solutions on the Pareto front near these reference points.

2.3 The light beam search

The light beam search was first introduced by Jaszkiewicz and Slowinski [10]. The DM first needs to indicate two points in the objective-space, the Aspiration Point (AP), denoted by $\vec{z}^r$ and the Reservation Point (RP), denoted by $\vec{z}^v$. In situations where the AP and RP are not given, some other points like the *nadir point* and *ideal point* can be used instead. The search direction is given from AP to

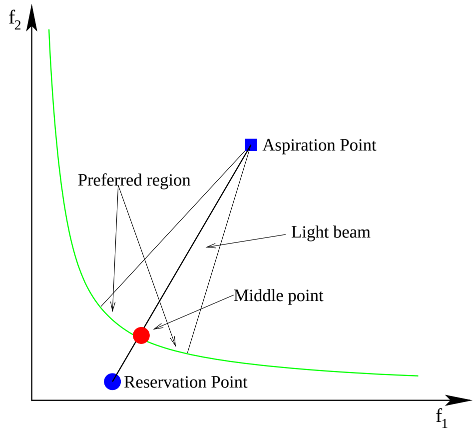

Figure 2: Light beam search

RP. Metaphorically, this illustrates a light beam originating from AP in the direction of RP. Figure 2 illustrates the classical light beam search setup in a two-objective space.

In the classical MCDM literature the light beam search method uses an *achievement scalarizing function* (given by (5)), which is to be minimized. If $\vec{x}$ is a solution in the search-space,

$$minimize \quad \max_{i=0,\ldots,M-1} \{\lambda_i \left(f_i(\vec{x}) - z_i^r\right)\} + \rho \sum_{i=0}^{M-1} \left(f_i(\vec{x}) - z_i^r\right) \tag{5}$$

where, $\vec{z}^r = [z_0^r, \ldots, z_{M-1}^r]$ and $\vec{z}^v = [z_0^v, \ldots, z_{M-1}^v]$. ρ is a sufficiently small positive number called the *augmentation coefficient* usually set to 10^{-6}. $\vec{\lambda} = [\lambda_0, \ldots, \lambda_{M-1}]$, where $\lambda_i > 0$ is a weighted vector. This weighted vector is derived from (6), where in a minimization problem $z_i^r > z_i^v$.

$$\lambda_i = \frac{1}{z_i^r - z_i^v} \tag{6}$$

The projection of the AP in the direction of the RP will result in a middle point on the non-dominated solution front. In the usual notation, a middle point is given by $\vec{z}^c = [z_0^c, \ldots, z_{M-1}^c]$. The DM can then decide on a region surrounding this middle point, which gives the preferred region. This region is obtained by the notion of *outranking* (S) [10]. $\vec{a}$ outranks $\vec{b}$ (denoted by $\vec{a}S\vec{b}$) if $\vec{a}$ is considered to be at least as good as $\vec{b}$. This outranking is defined by either one of three possible threshold values. They are the *indifference threshold* (m_q), *preference threshold* (m_p) or *veto threshold* (m_v). For example, if the veto threshold values are $\vec{v} = [v_0, \ldots, v_{M-1}]$ then $\vec{x}S\vec{z}^c$ if $m_v(\vec{z}^c, \vec{x}) = 0$ where,

$$m_v(\vec{z}^c, \vec{x}) = card\{i : f_i(\vec{x}) - z_i^c \geq v_i, i = 0, \ldots, M-1\} \tag{7}$$

Solutions are obtained in this preferred region *illuminated* by the light beam.

3. RELATED WORK

Integrating user-preferences into EMO algorithms has been increasing in popularity during the past few years. Deb *et al.* [5] presented an EMO algorithm incorporating the reference point method into NSGA-II [2], which was one of the very first attempts in integrating a preference method to an EMO algorithm. The reference direction method, an extension of the reference point method, was also incorporated into NSGA-II [3], and also subsequently the light beam search method [4]. In [22] the work in [5] was extended by introducing reference point based PSO algorithms. These user-preference based EMO algorithms all use *dominance comparisons* to select their candidate solutions. Unfortunately, these algorithms suffer from the problem of not being able to distinguish solutions effectively for problems with a large number of objectives, where most solutions are non-dominated to each other. Consequently these algorithms are less effective in search, and inclined to converge prematurely to local Pareto-fronts. To address this issue, this paper introduces a distance metric utilizing the user-preference information which is provided by the DM. This method removes the need to use dominance comparisons.

There are several examples in the EMO literature where the dominance concept has been altered to better suit algorithms in many-objective problems. A scheme named *ranking dominance* was introduced in [15], where solution points are ranked according to each objective. Then an aggregation function is used to obtain a fitness value from all the rank values. For a minimization problem, if an individual's fitness obtained from the aggregate function is less than another individual, then that individual is said to be *better* or dominant than the other.

In [20] a relation named $\epsilon - Preferred$ was introduced. This was an extension of the original *Favour* relation introduced in [7]. The Favour relation is a relaxed version of dominance. Here, $\vec{a}$ is said to be *preferred* (or *favoured*) to $\vec{b}$ if $\vec{a}$ is better than $\vec{b}$ in a larger number of objectives. It is also interesting to note that this relation is not transitive like dominance.

A method to avoid using the standard dominance relation in the leader selection stage of a multi-objective PSO algorithm was introduced in [14], where leaders are obtained from the population using a *gradual dominance* relation. The particles are assigned with a *ranking value*, which is the maximum of the *degree of being dominated*. This value is calculated using a *fuzzy scheme* described extensively in [12]. Leaders are chosen from a set of particles with the lowest ranking values. These lowest ranked particles will also be the least crowded. A distance measurement in [19] was used to rank particles to obtain suitable leaders for the population to follow in many-objective problems. This distance metric differs from our proposed approach because it is used only to rank particles, but not to guide the population towards preferred regions of the objective-space.

Comparing with these approaches modifying the dominance concept, the advantage of the proposed distance metric is its simplicity and efficiency, as shown in the following sections.

4. MDEPSO ALGORITHM WITH THE DISTANCE METRIC

The distance metric obtained by user-preference methods is integrated into the original MDEPSO [21] as:

- **Step 1: Initialize the particles**
  A population of size N is first initialized. Here, a particle's decision variables are obtained from (8).

$$rand(0.0, 1.0) * (UB - LB) + LB \tag{8}$$

where, $rand(0.0, 1.0)$ represents a random number generated uniformly between $[0.0, 1.0]$. LB and UB are the lower-bounds and upper-bounds respectively of the decision variables of a multi-objective problem instance. The velocity is initialized to a random value in the interval $[0, UB - LB]$. The personal best of an individual is set to its current position. Half of the population's direction is reversed by setting the velocity to negative according to a coin toss.

After initialization, each particle's distance to the preferred regions are calculated. In the reference point method, for any particle $\vec{x}$, its distance to a reference point $\vec{\bar{z}}$ can be derived from (4) as:

$$dist(\vec{x}) = \max_{i=0,\ldots,M-1} \{w_i(f_i(\vec{x}) - \bar{z}_i)\} \tag{9}$$

Similarly in the light beam search for any particle $\vec{x}$, its distance to a middle point $\vec{z^c}$ can be derived from (5) as:

$$dist(\vec{x}) = \max_{i=0,\ldots,M-1} \{\lambda_i \left(f_i(\vec{x}) - z_i^r\right)\} + \rho \sum_{i=0}^{M-1} \left(f_i(\vec{x}) - z_i^r\right) \tag{10}$$

The particles are then evaluated with the objective functions and fitness is assigned.

- **Step 2: Obtain leaders to guide the population**
  Leaders are generated using the DE rule given by (3). These leaders are sorted according to the distance metric given by (9) or (10), depending on the preference method.

  The following strategies are used to control the spread of solutions in the preferred regions. When sorting the leaders for the reference point method we use a $\delta(> 0)$ value to maintain a control over the spread of solutions. For two leaders $\vec{a}$ and $\vec{b}$ we consider $dist(\vec{a}) \equiv dist(\vec{b})$ if and only if $\left|dist(\vec{a}) - dist(\vec{b})\right| < \delta$. This provides a diverse set of potential leaders. A small value for δ represents a smaller spread, while a large value will give a larger spread. A subset of the sorted leaders (for example 10% of the population) closest to the preferred regions is chosen to guide the population. For the light beam search we used the set of leaders who outrank the middle point, which are obtained by (7). The threshold values in the outranking procedure will indicate an amount of spread. Empirical results show that these mechanisms provide *some control* over the spread of solutions.

- **Step 3: Move the particles**
  Each particle chooses its leader randomly from the sorted set of potential leaders. Using this leader as the global best the particle updates its velocity and position according to the PSO update rules (1) and (2).

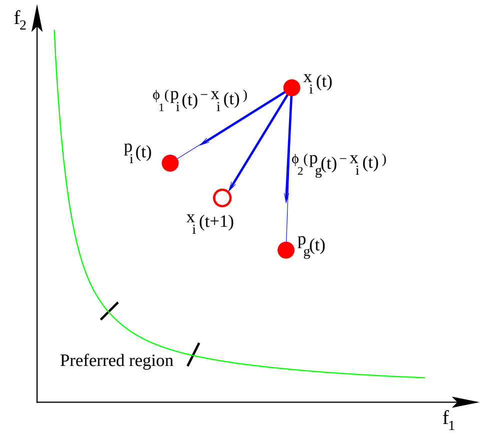

Figure 3: Move a particle and update its personal best depending on the distance to the preferred region

- **Step 4: Update the particles' personal bests**
  Each particle updates its personal best according to the distance metric given in (9) or (10), depending on the preference method. For example, as shown in figure 3, since $\vec{x}_i(t+1)$ is closer to the reference point or the middle point it will become its new personal best $\vec{p}_i(t+1)$. If $\vec{x}_i(t+1)$ is further away from the reference point or the middle point then the personal best is unchanged ($\vec{p}_i(t+1) = \vec{p}_i(t)$).

- **Step 5: Obtain the particles to move to the next iteration**
  The population of N particles at the beginning of the iteration is combined with the N updated particles to create a population of size $2N$. This $2N$ population is sorted according to (9) or (10) to obtain a population of size N closest to the preferred region. These N particles will survive to the next iteration.

The steps 2 to 5 are repeated until the maximum number of iterations is reached.

The crux of this algorithm is seen in steps 4 and 5. In previous multi-objective PSO algorithms [16, 17, 21, 22] the updating of the personal bests and the selection of the next iteration were done using dominance comparisons. In this proposed algorithm these steps are done using the distance metric utilizing the user-preference information. It is also useful to realize that the many-objective problem is not converted to a single-objective problem with the use of scalarizing functions as seen in traditional MCDM literature. Although (9) and (10) provide a single dimension value, the target is not to optimize that value but to use the value as a metric to guide the population. This approach is especially effective in moving particles towards the Pareto front of the preferred regions for many-objective problems. We believe that this approach can be adopted in a similar way for any EMO algorithms incorporating user-preferences.

A dominance comparison based EMO algorithm normally has a computational complexity of $O(MN^2)$ because of the use of the non-dominated sorting procedure [2, 16]. However, the proposed distance metric approach only depends on the sorting procedure. As a result, the computational complexity of using the distance metric (for the entire population) is $O(NlogN)$.

5. EXPERIMENTS

We implemented two versions of MDEPSO, one with the reference point method and one with the light beam search. To evaluate the performance of the MDEPSO algorithms we used following test problem suits; ZDT [23] and WFG [9] for two-objective problems and DTLZ [6] for three and up to ten objective problems. These test problem suites contain many varieties of multi-objective problems including some multi-modal problems.

To compare the performance of our approach to a traditional dominance based EMO, we used the user-preference based NSPSO [16, 22] algorithm, implemented in two versions, one for the reference point method and the other for light beam search. Results were obtained by averaging over 50 runs on each algorithm on each problem instance. All the standard configurations of MDEPSO and NSPSO were used without tweaking any parameters. A population of 200 individuals were used for a maximum of 750 iterations.

For the user-preference based EMO algorithms described in [3, 4, 5, 22], the use of dominance comparisons may cause them less effective in handling difficult multi-modal problems. In some difficult multi-modal problems like ZDT4 and DTLZ3 the initial population can be generated in positions very far from the Pareto optimal front. In such situations the initial population can have objective values as large as 1000. The individuals have to navigate across many local fronts towards the Pareto optimal front, where the objective values are in the interval of $[0.0, 1.0]$. Such problems in higher number of objective-spaces become very difficult for dominance based user-preference algorithms because the individuals who move to the next iteration have to picked from a large number of non-dominated individuals. These *chosen* individuals may not necessarily be the best to guide the population to the preferred regions. With the use of a distance metric this scenario can be avoided.

In our initial experiments we observed that by restricting the search-space using a preference mechanism has a downside reducing the diversity of the population. To avoid this phenomenon particles will follow the leaders generated by the DE step in the original MDEPSO algorithm from time to time, rather than following a leaders closest to preferred regions.

The user-preference based MDEPSO algorithms using the distance metric is much more effective in handling problem instance of many-objectives. Due to the limitation of space we will only illustrate the results obtained on some of the difficult multi-modal problem instances.

5.1 MDEPSO with the reference point method

Figure 4 shows the result obtained for a ten-objective multi-modal DTLZ1 instance. Here, the reference point was at 0.5 for all of the objectives in the objective-space. $\delta = 0.01$. The default is not to have any bias towards any objective, which is given by setting the weights to equal 1.0 in (9).

The sum of the objective values of each particle was found to be in the range $[0.5012, 0.5109]$. This suggests that the particles are very close to the true Pareto front of DTLZ1, since it holds the condition $\sum_{i=0}^{M-1} f_i(\vec{x}) = 0.5$ for every $\vec{x}$ on the true Pareto optimal front.

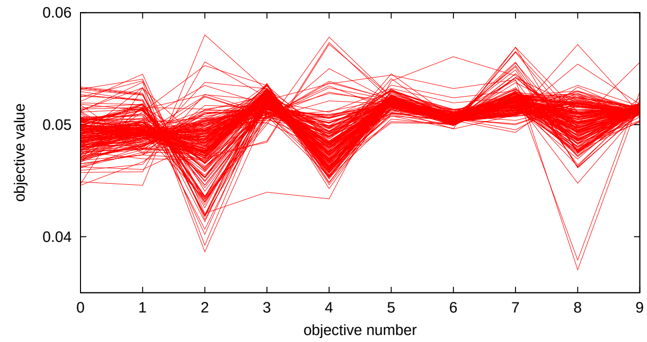

Figure 4: Ten-objective DTLZ1 with 1 reference point on MDEPSO (each line represents a solution point, where the intersection at the objectives axis represents the value for that objective)

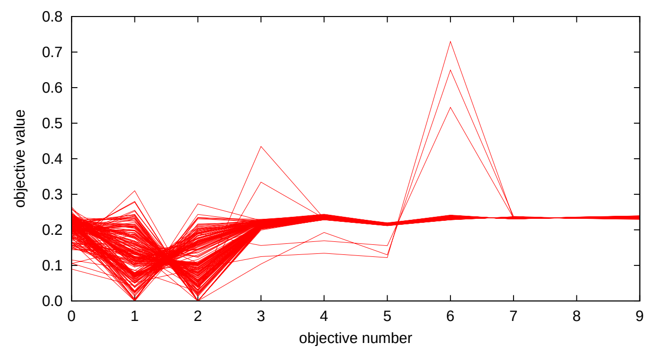

Figure 5: Ten-objective DTLZ1 with 1 reference point on NSPSO

For comparison figure 5 shows the result for NSPSO (using dominance comparisons) on DTLZ1 with the same parameter setting. The sum of the objective values of each particle was in $[2.0293, 2.0807]$. This shows that NSPSO had converged to a local optimal front.

Figure 6 shows the solutions fronts obtained for the two-objective multi-modal ZDT4 with two reference points on MDEPSO. Here, the two reference points have spread values of $\delta = 0.01$ and $\delta = 0.05$. Figures 7 and 8 illustrates the solution fronts obtained for two-objective multi-modal WFG4 and three-objective DTLZ1 problem instances respectively on MDEPSO. Here $\delta = 0.01$ and no bias in any objective was used.

In our experiments we observed that for simpler test functions such as ZDT1–ZDT3 and DTLZ2 the population converged to the preferred regions in about 200 iterations on average. The results obtained for ZDT6 with two reference points is given in figure 9. Here, spread values of $\delta = 0.01$

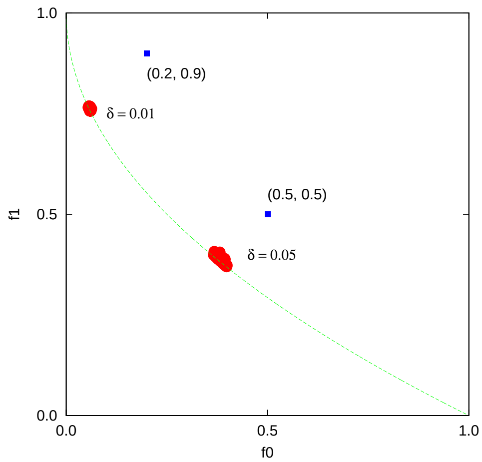

Figure 6: Two-objective ZDT4 with 2 reference points

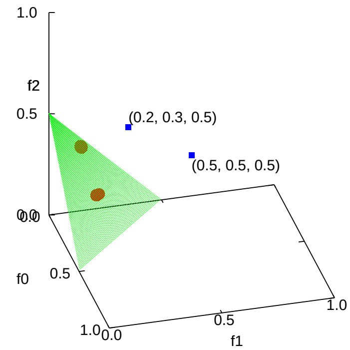

Figure 8: Three-objective DTLZ1 with 2 reference points

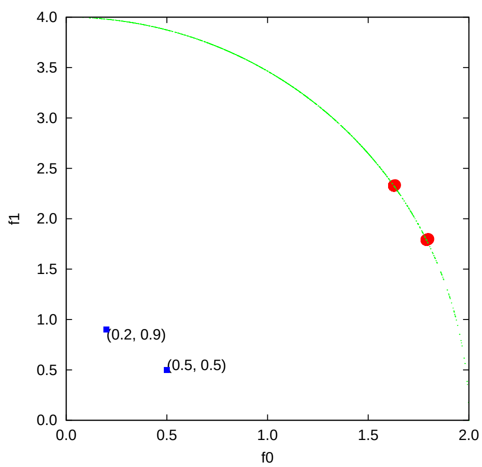

Figure 7: Two-objective WFG4 with 2 reference points

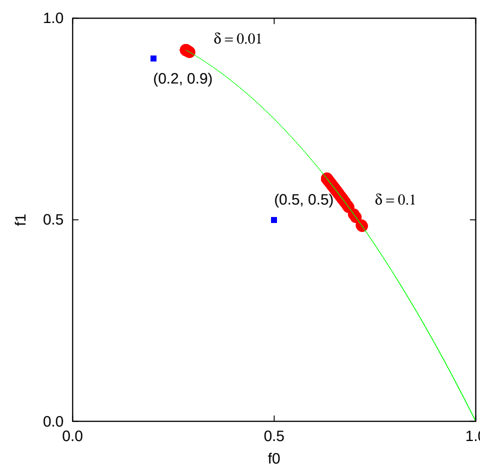

Figure 9: Two-objective ZDT6 with 2 reference points

and $\delta = 0.1$ were used for the two reference points separately. It is also interesting to note that the reference point at $(0.2, 0.9)$ is outside the region bounded by the Pareto optimal front. However, MDEPSO successfully obtained the solutions near the reference point that is at the extreme end of the Pareto front.

5.2 MDEPSO with the light beam search

In our experiments for the light beam search we used a veto threshold value of 0.05 in every objective to obtain the preferred region. In the ten-objective DTLZ3 instances the AP was set to be the *nadir* point having the value of 1.0 for all objectives and the RP to be the *ideal point* having 0.0 for all objectives. DTLZ3 is one of the more difficult multi-modal problems having close to 3^{10} number of local Pareto fronts and one global Pareto front.

The solutions given in figure 10 show that for each particle $\vec{x}$, the sum of its squared objective values $(\sum_{i=0}^{M-1} (f_i(\vec{x}))^2)$ gives values in $[1.005, 1.0083]$. This shows that the points are very close to the true Pareto front. This is deduced from the property of DTLZ3 where each $\vec{x}$ on the true Pareto front gives $\sum_{i=0}^{M-1} (f_i(\vec{x}))^2 = 1$.

For comparison figure 11 shows the result for NSPSO (using dominance comparisons) with the same parameter settings. Here values obtained by $\sum_{i=0}^{M-1} (f_i(\vec{x}))^2$ for each of the final solution points were in the range $[5.8417, 5.9102]$. This indicates that NSPSO was unable to locate the global front of the ten-objective multi-modal DTLZ3 problem.

Figure 12 shows the final solutions obtained for DTLZ3 with three-objectives. Here, two light beams having AP $(1.0, 1.0, 1.0)$ and RPs at $(0.5, 0.0, 0.0)$ and $(0.0, 0.0, 0.0)$ respectively were used. Two veto thresholds were used one with 0.1 and the other having 0.05, in all objectives.

A very interesting result can be seen for the two-objective WFG4 in Figure 13. Here, the light beams are located in the infeasible region of the objective-space. However, MDEPSO using the light beam search still managed to guide the pop-

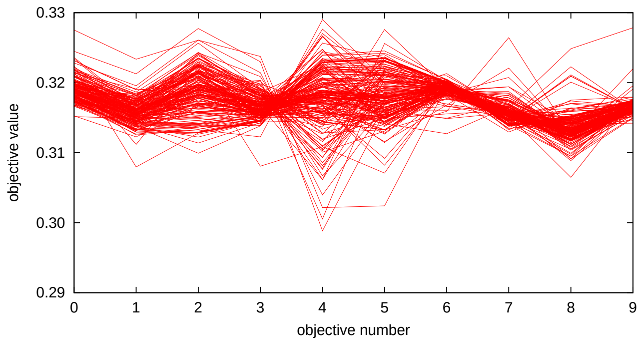

Figure 10: Ten-objective DTLZ3 with 1 light beam on MDEPSO

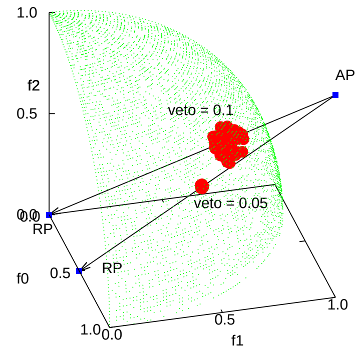

Figure 12: Three-objective DTLZ3 with 2 light beams

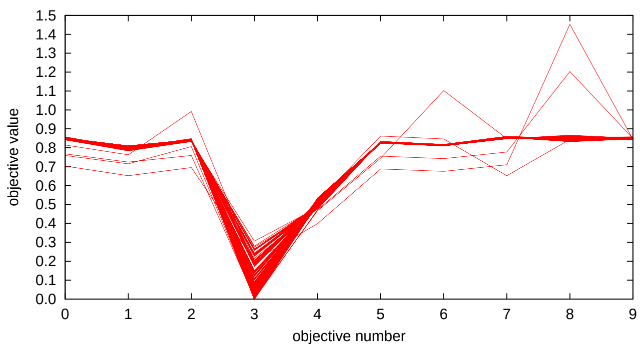

Figure 11: Ten-objective DTLZ3 with 1 light beam on NSPSO

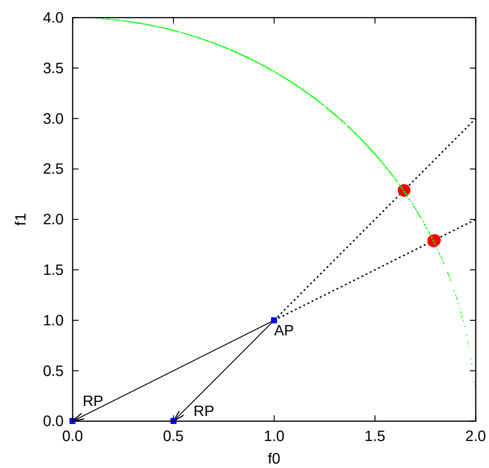

Figure 13: Two-objective WFG4 with 2 light beams

ulation in the direction of the light beams until solutions are located on the global Pareto front.

Figure 14 shows a two-objective WFG2 problem instance with two light beams. It is interesting to note that though the light beam (with AP $(5.0, 5.0)$ and RP $(0.0, 2.0)$) goes through the disjoint Pareto front, it was still able to locate solutions on the region of the Pareto front which is closest to this light beam. The distance metric guides particles in the direction given by the vector from AP to RP. This is possible because the algorithm concentrates its search in the direction of this vector. With a population of particles the algorithm has the ability to move in parallel along the direction of this vector until a middle point is found on the Pareto front.

6. CONCLUSION AND FUTURE WORK

In this paper we have proposed a distance metric for many-objective PSO algorithms which does not rely on dominance comparisons to find solutions. The proposed distance metric obtained by utilizing user-preferences, either by the reference point or light beam search method, has been integrated into a previously developed MDEPSO algorithm. Compared with a user-preference based EMO algorithm (NSPSO) that uses only dominance comparisons for selection, the resulting user-preference based MDEPSO algorithm is shown to provide better performances especially for problems characterized by a high number of objectives and multiple local Pareto-fronts,

Interesting results can be also observed in the behaviour of the proposed EMO algorithms when the preferred regions specified by the DM are in the infeasible regions. In such cases the EMO algorithms are still able to converge to the Pareto front near those specified preferred regions. This property provides an advantage to the DM, since the DM does not have to have the knowledge of where the actual true Pareto optimal front is.

In future we will carry out more comprehensive studies on the distance metric and variations of it. We are also interested in applying EMO algorithms based on this distance metric to solving real world problems.

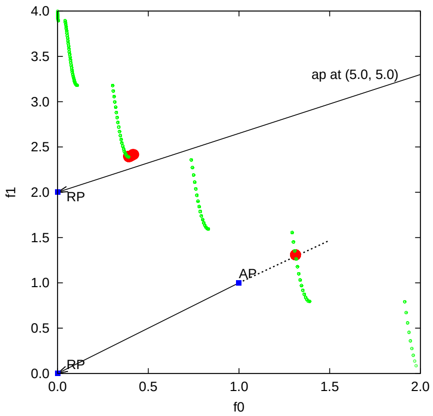

Figure 14: Two-objective WFG2 with 2 light beams

7. REFERENCES

[1] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6:58–73, 2002.

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

[3] K. Deb and A. Kumar. Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 781–788. ACM Press, 2007.

[4] K. Deb and A. Kumar. Light beam search based multi-objective optimization using evolutionary algorithms. In *IEEE Congress on Evolutionary Computation (CEC)*, 2007.

[5] K. Deb and J. Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 635–642. ACM Press, 2006.

[6] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. In *Evolutionary Multiobjective Optimization (EMO): Theoretical Advances and Applications*, pages 105–145. Springer, 2005.

[7] N. Drechsler, R. Drechsler, and B. Becker. Multi-objective optimisation based on relation favour. In *Evolutionary Multi-Criterion Optimization (EMO)*, volume 1993 of *Lecture Notes in Computer Science*, pages 154–166. Springer, 2001.

[8] M. Ehrgott and X. Gandibleux. *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Survey*, volume 52 of *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, 2002.

[9] S. Huband, P. Hingston, L. Barone, and R. L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10:477–506, 2006.

[10] A. Jaszkiewicz and R. Slowinski. The light beam search approach -an overview of methodology and applications. *European Journal of Operational Research*, 113:300–314, 1999.

[11] J. Kennedy and R. C. Eberhart. *Swarm intelligence*. Morgan Kaufmann Publishers Inc., 2001.

[12] M. Köppen and R. Vicente-Garcia. A fuzzy scheme for the ranking of multivariate data and its application. In *North American Fuzzy Information Processing Society (NAFIPS)*, pages 140–145, 2004.

[13] M. Köppen, R. Vicente-Garcia, and B. Nickolay. Fuzzy-pareto-dominance and its application in evolutionary multi-objective optimization. In *Evolutionary Multiobjective Optimization (EMO)*, volume 3410 of *Lecture Notes in Computer Science*, pages 399–412. Springer, 2005.

[14] M. Köppen and K. Yoshida. Many-objective particle swarm optimization by gradual leader selection. In *International Conference on Adaptive and Natural Computing Algorithms (ICANNGA)*, volume 4431 of *Lecture Notes in Computer Science*, pages 323–331. Springer, 2007.

[15] S. Kukkonen and J. Lampinen. Ranking-dominance and many-objective optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 3983–3990, 2007.

[16] X. Li. A non-dominated sorting particle swarm optimizer for multiobjective optimization. In *Genetic and Evolutionary Computation Conference (GECCO)*, volume 2723 of *Lecture Notes in Computer Science*, pages 37–48. Springer, 2003.

[17] X. Li. Better spread and convergence: Particle swarm multiobjective optimization using the maximin fitness function. In *Genetic and Evolutionary Computation Conference (GECCO)*, volume 3102 of *Lecture Notes in Computer Science*, pages 117–128. Springer, 2004.

[18] K. Miettinen. Some methods for nonlinear multi-objective optimization. In *Evolutionary Multi-Criterion Optimization (EMO)*, volume 1993 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2001.

[19] S. Mostaghim and H. Schmeck. Distance based ranking in many-objective particle swarm optimization. In *Parallel Problem Solving from Nature (PPSN)*, Lecture Notes in Computer Science, pages 753–762. Springer, 2008.

[20] A. Sülflow, N. Drechsler, and R. Drechsler. Robust multi-objective optimization in high dimensional spaces. In *Evolutionary Multiobjective Optimization (EMO)*, volume 4403 of *Lecture Notes in Computer Science*, pages 715–726. Springer, 2006.

[21] W. R. M. U. K. Wickramasinghe and X. Li. Choosing leaders for multi-objective pso algorithms using differential evolution. In *Simulated Evolution And Learning (SEAL)*, volume 5361 of *Lecture Notes in Computer Science*, pages 249–258. Springer, 2008.

[22] W. R. M. U. K. Wickramasinghe and X. Li. Integrating user preferences with particle swarms for multi-objective optimization. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 745–752. ACM, 2008.

[23] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8:173–195, 2000.