Integrating User Preferences with Particle Swarms for Multi-objective Optimization

W. R. M. U. K. Wickramasinghe, X. Li
School of Computer Science and Information Technology
RMIT University
Melbourne, VIC 3001, Australia
{uwickram, xiaodong}@cs.rmit.edu.au

ABSTRACT

This paper proposes a method to use reference points as preferences to guide a particle swarm algorithm to search towards preferred regions of the Pareto front. A decision maker can provide several reference points, specify the extent of the spread of solutions on the Pareto front as desired, or include any bias between the objectives as preferences within a single execution. We incorporate the reference point method into two multi-objective particle swarm algorithms, the non-dominated sorting PSO, and the maximinPSO. This paper first demonstrates the usefulness of the proposed reference point based particle swarm algorithms, then compare the two algorithms using a hyper-volume metric. Both particle swarm algorithms are able to converge to the preferred regions of the Pareto front using several feasible or infeasible reference points.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

Particle swarm optimization, Multi-objective problems, User-preference methods, Reference point method, Non-dominated sorting, Maximin strategy

1. INTRODUCTION

The use of evolutionary algorithms in the field of multi-objective optimization has been a very popular area of research in the last decade [5]. Recently there has been an increasing interest to incorporate user-preference mechanisms seen in the Multi-Criteria Decision Making (MCDM) [11] literature into these Evolutionary Multi-objective Optimization (EMO) algorithms [2, 7, 8, 9, 22]. In this paper we

describe an attempt to integrate a preference method using reference points with Particle Swarm Optimization (PSO) algorithms. These PSO algorithms can be used effectively to direct the search to preferred regions of the Pareto front. We will demonstrate how to integrate the reference point method into two multi-objective PSO algorithms, the non-dominated sorting PSO (NSPSO) [16] and maximinPSO [17], and provide a comparison of their performance.

User-preference methods as described in the MCDM literature come in two forms. 1) *A priori* methods, where a Decision Maker (DM) gives preferences first then the algorithm finds solutions considering those preferences; 2) *A posteriori* methods, where after an algorithm provides all possible solutions the DM selects the interesting ones [19]. The reference point method is an *a priori* approach, where first the DM provides reference points in the objective-space and the search algorithm will concentrate around those points to find solutions. The advantage is that most computing effort can be spent on the preferred areas, instead of the entire search-space. This is especially important as the number of objectives increases.

PSOs have been very successful as EMO algorithms [21]. This paper shows that using reference points as preferences, a PSO can be guided towards desired regions of the search-space. We also propose a comparison mechanism for user preference based EMO algorithms using the hyper-volume metric. This mechanism is illustrated in the comparison of the reference point based NSPSO and maximinPSO algorithms.

The paper is organized as follows. Section 2 briefly describes the background material for the study including the formal definitions of multi-objective optimization, particle-swarms and the reference point method. Section 3 presents related work carried out in the field. Section 4 presents our algorithms. The experiments used to evaluate the algorithms will be provided in section 5. Finally, in section 6, we present our conclusions and avenues for future research.

2. BACKGROUND

We will first present the background material used in this research, then give some examples of similar work in the field.

2.1 Multi-objective optimization

We can formally define such an optimization as; there is a set of objective functions which need to be maximized or minimized. Without loss of generality we assume that the objectives are to be minimized. This can be generally defined as a set of functions described in equation 1.

$$minimize \ \ f_m(\vec{x}) \ \ where \ \ m = 1, 2, \dots, M \ \ , \qquad (1)$$

here $f_m(\vec{x})$ is an objective function. The solution $\vec{x}$ is a vector of N decision variables, which is given by $\vec{x} = (x_1, x_2, \dots, x_N)^T$. These decision variables should be between an upper and lower bound values as defined by the problem. The objective function provides the mapping between the decision-space and the objective-space.

In multi-objective optimization, solutions are compared with each other using the notion of *dominance*. For two decision vectors $\vec{x}^{(1)}$ and $\vec{x}^{(2)}$, dominance (denoted by $\prec$) is defined as follows:

$$\vec{x}^{(1)} \prec \vec{x}^{(2)} \Leftrightarrow \forall i \ f_i(\vec{x}^{(1)}) \leq f_i(\vec{x}^{(2)}) \wedge \exists j \ f_j(\vec{x}^{(1)}) < f_j(\vec{x}^{(2)})$$
$$(2)$$

The decision vector $\vec{x}^{(1)}$ is said to dominate $\vec{x}^{(2)}$ if and only if, $\vec{x}^{(1)}$ is as good as $\vec{x}^{(2)}$ in all objectives and $\vec{x}^{(1)}$ is strictly better than $\vec{x}^{(2)}$ in at least one objective.

A decision vector is said to be *Pareto optimal* if and only if there is no other vector in the search-space that dominates it. This set of Pareto optimal solutions is called the Pareto optimal set.

2.2 Particle swarms

PSO is a nature-inspired algorithm which mimics the behaviour of swarming bees, flocking birds or schooling fish [14]. These behaviours are modelled as rules governing the movement of particles in the search-space. Each particle moves in the search-space by adjusting its position and velocity, which are influenced by its interaction with other particles in its neighbourhood. The ith particle's velocity and position at time t are updated to time $t + 1$ according to the following two equations respectively:

$$\vec{v_i}(t + 1) = \chi(\vec{v_i}(t) + \phi_1(\vec{p_i} - \vec{x_i}(t)) + \phi_2(\vec{p_g} - \vec{x_i}(t))) \quad (3)$$

$$\vec{x_i}(t + 1) = \vec{x_i}(t) + \vec{v_i}(t) \qquad (4)$$

This version of PSO is known as the *Constriction Type 1"* as defined by Clerc and Kennedy [3]. Here $\vec{v_i}$ is the velocity of the ith particle and $\vec{x_i}$ its position. The variables ϕ_1 and ϕ_2 are random numbers generated uniformly between $[0, \frac{\varphi}{2}]$. Here φ is a constant equal to 4.1 [3]. $\vec{p_i}$ is the best position found by the particle (also known as *personal best*); $\vec{p_g}$ is the best position (also known as *global best*) found in the particle's neighbourhood. χ is called the constriction factor, and is used to prevent a particle from exploring too far in the search-space. We used $\chi = 0.7298$, which was calculated from $\frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}$ [3]. Essentially each particle moves towards somewhere between its personal best and the global best.

2.3 Reference point method

The classical reference point method was first described by Wierzbicki [9, 19]. A reference point $\overline{z}$ for a multi-objective problem is a point consisting of *aspiration values* for each objective. This reference point is used to construct a single objective function (equation 5), which is to be minimized over the entire search-space S, where $\vec{x} \in S$.

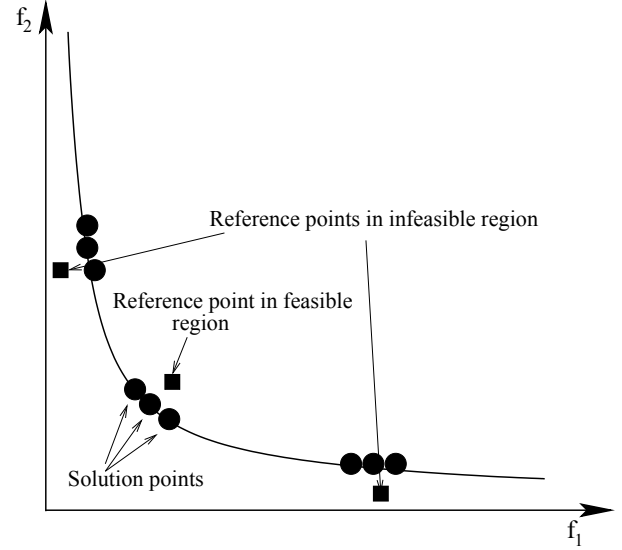$$minimize \ \ \max_{i=1\dots M}\{w_i(f_i(\vec{x}) - \overline{z}_i)\} \ \ , \qquad (5)$$

Figure 1: Reference point method

where $\overline{z}_i$ is the ith component of the reference point. w_i is a weight associated with the ith objective. The DM can assign a value for this weight, which represents any bias towards that objective.

The DM is presented with the objective-space where preferred regions can be indicated to the algorithm with the use of reference points. Figure 1 illustrates the classical reference point method in a two-objective space. In our research we define a reference point as an array of aspiration values. The number of elements in the array corresponds to the number of objectives for a given problem. In the general sense such a reference point would indicate a potential solution point consisting of values for each objective.

A reference point could be in any region either feasible or infeasible, because the DM might not know beforehand where the true Pareto front is for a given problem. The algorithm will then attempt to find a set of solution points on the Pareto front which is closest to the given reference point. An advantage of using an evolutionary algorithm is, unlike a classical optimization approach where a single solution is found, a set of solutions can be found in a single run near the reference point.

2.4 Multi-objective PSO algorithms

In our research we used two multi-objective PSOs to find the non-dominated particles in the search-space. We will first present these multi-objective PSOs and then discuss their performance.

2.4.1 Non-dominated Sorting PSO

The notion of dominance is widely used in EMO literature [6, 16]. The most representative example is Non-dominated Sorting Genetic Algorithm II (NSGA-II), which has been shown to be very effective in multi-objective optimization [6].

NSPSO [16] employs the non-dominated sorting procedure used in NSGA-II. In NSPSO, a population of N parents generate N offspring to create a population of size $2N$. First, from this population, the non-dominated solutions are extracted. This would be called the first non-dominated

front. This front is removed temporarily from the population. Then from the dominated set, the non-dominated solutions are extracted again. This front is the second non-dominated front. This process is repeated until there are a total of N individuals in the non-dominated fronts. These N individuals will be carried over to the next generation.

All particles in the first non-dominated front are sorted according to the crowding distance values. Each particle chooses its leader (global best) from a set of least-crowded particles (top 10% of the particles in the first non-dominated front sorted in decreasing order of crowding distance values). Consequently, particles of the future iterations are well distributed along the Pareto front.

2.4.2 Maximin PSO

The maximin strategy originated in game theory [20]. Balling [1] first proposed a method of deriving the fitness using the maximin approach for multi-objective problems. The fitness of any decision vector can be computed using Balling's approach.

In maximinPSO [17], which also adopts the maximin strategy, the minimum of the difference of each objective i between any two decision vectors $\vec{x}^{(k)}$ and $\vec{x}^{(j)}$ in the population is given by $\min_{i=1...M}\{f_i(\vec{x}^{(k)}) - f_i(\vec{x}^{(j)})\}$. The fitness of any $\vec{x}^{(k)}$ is given by the maximum of the minimum values between $\vec{x}^{(k)}$ and all other vectors in the population (other than $\vec{x}^{(k)}$):

$$fitness(\vec{x}^{(k)}) = \max_{j=1...pop\_size:j\neq k}\{\min_{i=1...M}\{f_i(\vec{x}^{(k)}) - f_i(\vec{x}^{(j)})\}\}$$

(6)

The particles with fitness values less than 0.0 are considered as the non-dominated solutions. These non-dominated solutions are in the first non-dominated front. The total number of individuals that move onto the next generation will consist of all individuals in the first non-dominated set. If the non-dominated set is less than desired then dominated individuals are chosen at random to fill the vacant positions. A particle will choose a leader (global best), randomly from the top 10% of least crowded particles in the non-dominated front.

In [17], it was shown that the maximinPSO performed better than NSGA-II on ZDT test functions [23]. However, the maximinPSO seems to be sensitive to population sizes. The maximinPSO can only identify the first non-dominated front, which could also be a drawback. The upside is that maximin fitness provides information on both non-dominance and diversity. There is also a subtle difference between NSPSO and maximinPSO in ways of choosing a leader for each particle. In NSPSO a particle chooses its leader from the top 10% of the least-crowded particles randomly. However, in maximinPSO this was done by choosing values from all dimensions from particles with smallest maximin fitness values. One of our aims in this study is to investigate the effects of these differences when employing a reference point search strategy.

2.5 Hyper-Volume metric

To compare NSPSO and maximinPSO we used the Hyper-Volume (HV) metric [15]. HV gives the total volume bounded by the solutions points on the Pareto front and a selected point in the search-space. This selected point is usually called the *nadir point*. At the nadir point, all objectives are
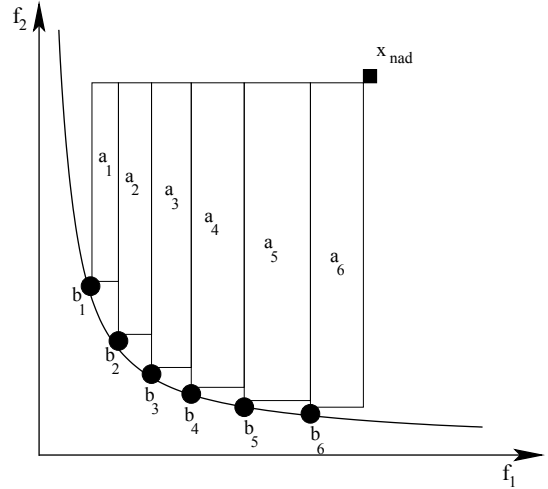

Figure 2: HV for a two-objective problem

at their worst values simultaneously [18]. The nadir point x_{nad} is given as $x_{nad} = (f_1^{nad}(\vec{x}), f_2^{nad}(\vec{x}), \ldots, f_M^{nad}(\vec{x}))^T$, where $f_i^{nad}(\vec{x}) = \max_{j=1...pop\_size}\{f_i(\vec{x}^{(j)})\}$.

In experiments where multiple runs are required to obtain an average HV value, the population will be the sum of all the final non-dominated particles from each run combined together. The nadir point is computed from this combined population, and then will be used to obtain the HV value for each run. This method ensures a consistent nadir point for all runs of the experiment.

The formal definition of HV according to [12] is the Lebesgue measure (Λ) of the union of all hypercubes a_i defined by a non-dominated point $b_i \in B$ (B is the set of all the non-dominated points), and the nadir point x_{nad} (equation 7). Figure 2 illustrates hypercubes for a two-objective problem. The HV value is the sum of all the rectangular areas.

$$hv(B) = \Lambda(\{\bigcup_i a_i | b_i \in B\}) = \Lambda(\bigcup_{b\in B}\{x | b \prec x \prec x_{nad}\})$$

(7)

When comparing two EMO algorithms, the one which gives a larger HV value is considered to be better. The volume calculated by this metric gives a measure on both the spread and the closeness of the solutions to the Pareto front. We use the HV to compare the reference point based NSPSO and maximinPSO. In our proposed algorithm the HV is calculated on the solution points of the preferred regions, rather than the entire Pareto front.

3. RELATED WORK

User-preference methods are used extensively in the classical optimization methods in MCDM literature [11]. A survey by Miettinen [19] shows many different *a priori* and *a posteriori* methods for nonlinear multi-objective optimization problems. However, most existing EMO algorithms are *a posteriori* methods, since they perform a search first before presenting a set of final solutions to the DM. This section presents some of the early EMO algorithms incorporating *a priori* methods.

Deb [4] proposed a *goal programming* approach with Genetic Algorithms (GA) as a mechanism to find solutions to

multi-objective problems. A modified NSGA algorithm was used to find a set of points closest to the supplied goal, which is a set of aspiration values for each objective. This approach did not care if the solutions were not found on the Pareto front. This is because; if the goal was in the feasible region then the algorithm is directed to that solution point. If the goal was in the infeasible region then the solution points closest to the goal was obtained.

Branke et al. [2] proposed the Guided Multi-Objective Evolutionary Algorithm (G-MOEA), where the definition of *dominance* was modified using the preferences given by the DM. Here the DM can provide accepted tradeoff values for each objective in a two-objective optimization problem. The tradeoff values for each objective gives a notion of preference between the two objectives. This provides guidance to the algorithm.

Thiele et al. [22] proposed an interactive EMO based on preferences. First, a rough idea of the Pareto front is presented to the DM, who then specifies some reference points as desired. The algorithm will then concentrate the search around these reference points. This method is a two-step approach since the algorithm must first find an approximate Pareto front and then incorporate the reference points.

Most recently Deb et al. [9] presented an EMO incorporating the reference point method into NSGA-II. This work was then extended to incorporate the reference direction method [7]. In a later study [8] the light beam search method [13] was integrated into NSGA-II. These studies showed that user-preference methods can be successfully integrated into GA. However, a performance metric for these user-preference based EMO algorithms was yet to be proposed. This paper extends the research in [9] by introducing reference point based PSOs. We use the HV metric as a comparison mechanism for user-preference based EMO algorithms.

4. REFERENCE POINT BASED PSO

The proposed reference point based PSO algorithm involves the following steps:

- **Step 1: Obtain the DM's preferences for reference points, spread and bias**
  The DM will provide one or more reference points, in the objective-space. The DM does not have to worry about the feasibility of the reference points, because the algorithm attempts to find the closest solutions on the Pareto front to the reference points. The DM can also specify the spread as a preference. The spread defines the extent of the solutions on the Pareto front near the reference point. The spread is given by a value δ, which is defined as the maximum variance of the distance values of the population. Figure 3 illustrates a small spread and a large spread for a reference point. Here $\delta = 0.01$ would mean that the variance of the population's distance values (derived from equation 8) should be less than 0.01. If the DM requires a smaller spread then the value of δ can be set to a lower value, as 0.001 in Figure 3. The value of δ defines the stop criteria. When the population's variance of the distance values are less than δ the algorithm stops otherwise the algorithm will continue until the maximum number of iterations is reached. The DM can also provide an array of weights (w_i) if any bias is required
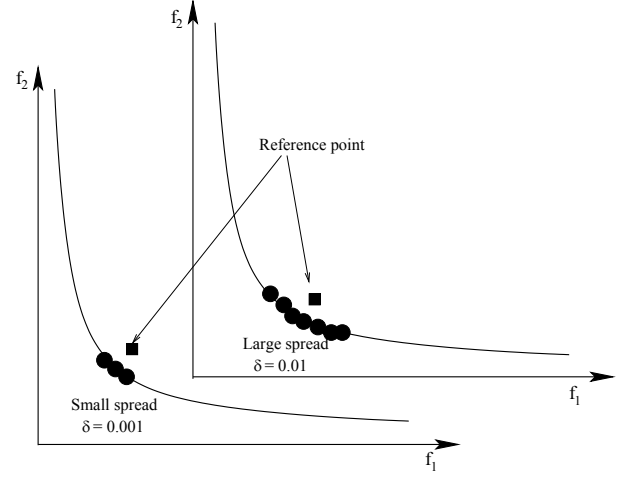

Figure 3: Spread of solutions along the Pareto front

between the objectives. Here $\sum_{i=1}^{M} w_i = 1.0$. If no bias is required then the values for all weights will be set to 1.0. The default execution of the algorithm is with no bias.

- **Step 2: Initialize the particles**
  The population is first initialized. Normally half the population's direction is reversed according to a coin toss. The particles are evaluated according to the objective functions and fitness is assigned. Each particle is assigned to the closest reference point. For any vector $\vec{x}$ the distance to a reference point $\overline{z}$ is defined by equation 8.

$$dist(\vec{x}) = \max_{i=1...M}\{w_i(f_i(\vec{x}) - \overline{z}_i)\} \qquad (8)$$

A particle's assigned reference point will remain unchanged throughout the run. The particle will choose a leader, which also has the same reference point.

- **Step 3: Obtain non-dominated solutions in the preferred regions**
  The non-dominated particles are extracted from the population. The non-dominated particles are selected by either NSPSO or maximinPSO.

- **Step 4: Rank non-dominated solutions according to the closeness to the reference points**
  On the first non-dominated front, each non-dominated particle's distance value to its assigned reference point is derived from equation 8. Then, the non-dominated particles assigned for each reference point are ranked according to the ascending order of distance values. Particles with low distance values are considered as candidates to be leaders.

- **Step 5: Choose leaders from the assigned ranked non-dominated set and move the particles**
  Each particle in the population will choose a leader (global best) from the assigned set of non-dominated solutions on the lowest ranked front. Here we consider a set (top 10% closest to the assigned reference point) from the ranked non-dominated particles as the pool
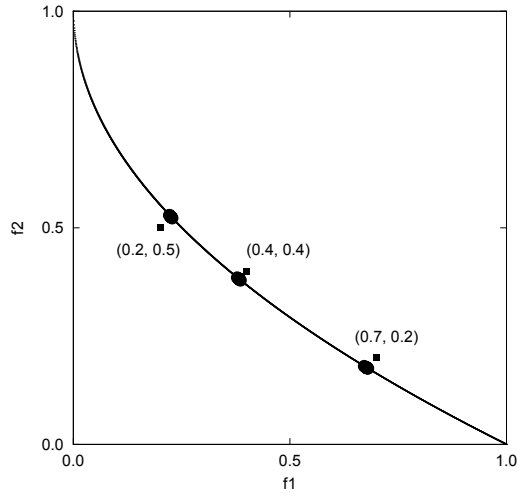
Figure 4: maximinPSO on ZDT1

Figure 5: maximinPSO on ZDT1 with different δ values

Figure 6: NSPSO on ZDT2

to choose leaders. NSPSO will choose one leader randomly, while maximinPSO will choose random values from all dimensions of a set of potential leaders and produce a global best. Then each particle will adjust their velocities and positions according to equations 3 and 4.

- **Step 6: Evaluate particles using the objective functions**
  The entire population is evaluated and fitness values are assigned. This fitness value will be next used to determine the dominance.

  The steps 3 to 6 are repeated until the stop criteria is met. The algorithm will stop once it has reached the desired spread of particles on the Pareto front near the reference points or the maximum number of iterations allowed.

5. EXPERIMENTS

We used two-objective problems from the ZDT test problem suite [23] and three-objective problems from the DTLZ test problem suite [10]. We will first demonstrate that the PSO variants can be guided using reference points and achieve various spread and bias of objectives on the Pareto front. Then we will compare the maximinPSO and NSPSO using the HV metric with varying population sizes.

The following two-objective problems were used: ZDT1 (convex Pareto optimal front), ZDT2 (concave Pareto optimal front), ZDT3 (disjoint convex Pareto optimal front) all having 30 decision variables each. DTLZ6 (concave Pareto optimal region) and DTLZ7 (disjointed Pareto optimal regions) were used as the three-objective problems. There were 12 and 22 decision variables for DTLZ6 and DTLZ7 respectively. For every decision variable $x_i \in [0.0, 1.0]$ in all the test problems. We set the maximum number of iterations to 500 for the two-objective problems and 750 for the three-objective problems.

5.1 Two-objective test problems

On ZDT1, the reference points (0.2, 0.5), (0.4, 0.4) and (0.7, 0.2) were used in a single run. (0.2, 0.5) is in the in-
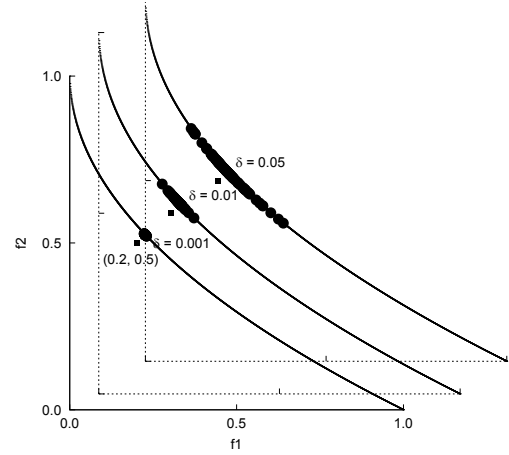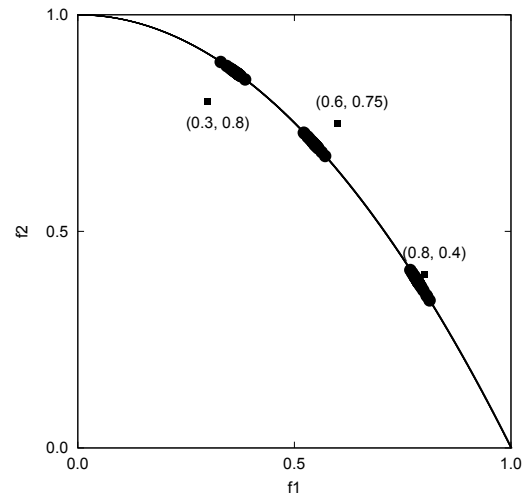
feasible region, while the other two reference points are in the feasible region. Figure 4 illustrates the result for maximinPSO at the end of 110 iterations.

Figure 5 shows different values of δ depicting the spread of solutions near the reference point (0.2, 0.5), for the maximinPSO. For results in Figure 4 the spread was set to $\delta = 0.001$. NSPSO depicted a similar behaviour for the solution points and spread on the Pareto front as the maximinPSO for the same reference points.

The results of NSPSO on ZDT2 with $\delta = 0.01$ is given in Figure 6 after 100 iterations. Here three reference points (0.3, 0.8), (0.6, 0.75) and (0.8, 0.4) were used in a single run. The first reference point is in the infeasible region, while other two are in the feasible region. Both the maximinPSO and NSPSO performed equally well for the same reference points.

Figure 7 illustrates the spread of solutions for the reference point (0.8, 0.4), $\delta = 0.001$ and when the DM has a bias preference between the objectives. Here we illustrate the solutions on the Pareto front where $\{w_1 = 0.8, w_2 = 0.2\}$ and $\{w_1 = 0.3, w_2 = 0.7\}$, while $\{w_1 = w_2 = 1.0\}$ shows no bias.
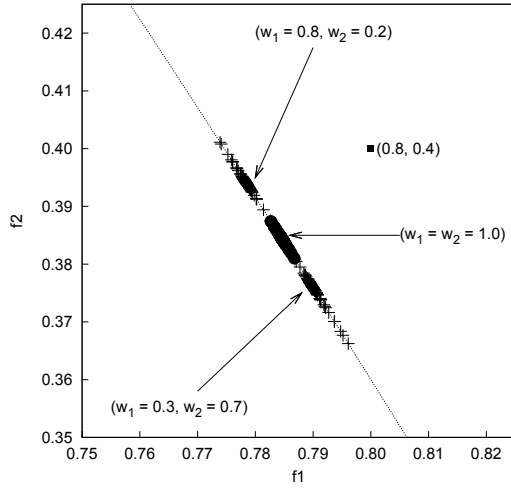
Figure 7: NSPSO on ZDT2 using bias for objectives

Figure 9: NSPSO on DTLZ6

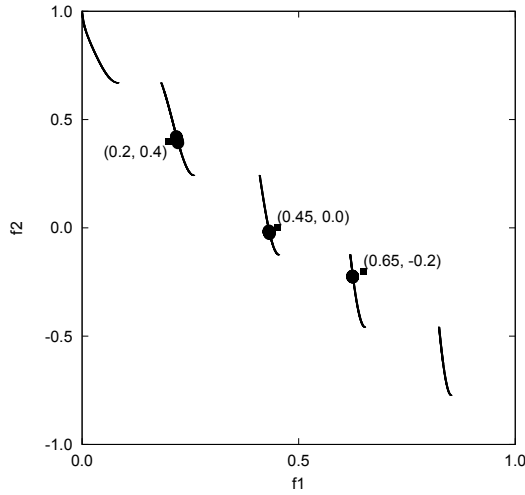Figure 10: maximinPSO on DTLZ7

Figure 8: maximinPSO on ZDT3

For ZDT1 and ZDT2 we observed that both maximinPSO and NSPSO were able to find solutions closest to feasible or infeasible reference points in both convex and concave problems. The DM was able to specify the required spread (by δ) and bias (by w_i) for any reference point.

We used the reference points (0.2, 0.4), (0.45, 0.0) and (0.65, -0.2) in a single run on the ZDT3 test problem using $\delta = 0.001$. Figure 8 illustrates the distribution of the final solution points at the end of 110 iterations with no bias using maximinPSO. The NSPSO was also able to find solutions similarly for the same reference points.

Our proposed reference point based PSO was able to find solutions near the specified reference points using both maximinPSO and NSPSO in a single run. The algorithms can also handle problems with disjoint Pareto fronts (ZDT3).

5.2 Three-objective test problems

Figure 9 shows the final solutions after 450 iterations with the reference points (0.2, 0.4, 0.9) and (0.7, 0.7. 0.5) in a single run on DTLZ6 using the NSPSO, $\delta = 0.001$ and no bias. The result was similar for maximinPSO.
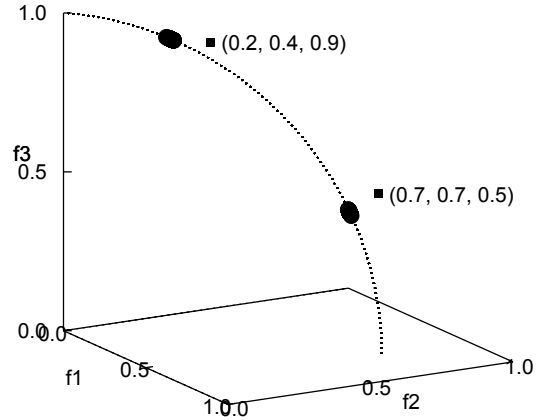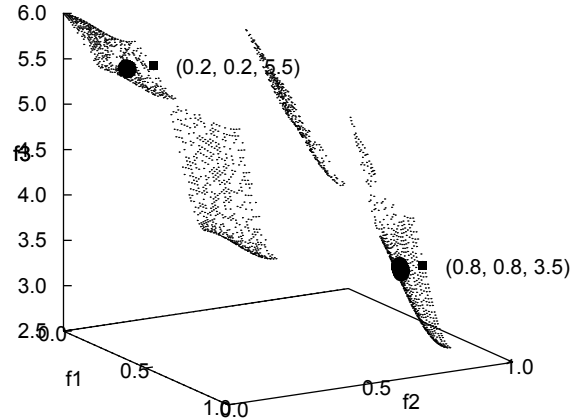
Figure 10 shows the results for reference points (0.2, 0.2, 5.5) and (0.8, 0.8, 3.5) in a single run on DTLZ7 for maximinPSO at the end of 550 iterations. Similar results were observed for NSPSO.

Figure 11 shows a series of snapshots of the swarm population on DTLZ7 using maximinPSO with reference points (0.2, 0.2, 5.5), (0.8, 0.8, 3.5) and $\delta = 0.01$ with no bias.

In the three-objective DTLZ7 problem the area covered by the solutions points does not depict a circular region. The goal of the experiments was not to achieve such results. The main aim was to guide the PSO in finding some preferred solutions points on the Pareto front, with some control over the spread as the DM desires.

5.3 Effect of varying population sizes

For each test problem and the PSO variants we conducted experiments with varying population sizes. The results were averaged over 50 runs. We used the worst values of the solution points from all 50 runs to determine the nadir point for the HV calculation. The average and standard deviation of the HV values over 50 runs are presented in Table 1 and 2.

Table 1 and 2 show the variable population sizes for the two-objective and the three-objective problems respectively. A fixed reference point for each test problem was used. The reference points are: (0.2, 0.5) for ZDT1, (0.3, 0.8) for ZDT2, (0.65, -0.2) for ZDT3, (0.2, 0.4, 0.9) for DTLZ6 and (0.2, 0.2,

Table 1: HV values for two-objective ZDT test problems

Pop Size / Problem	50		100		250		500	
	maximinPSO	NSPSO	maximinPSO	NSPSO	maximinPSO	NSPSO	maximinPSO	NSPSO
ZDT1	**0.4335** ±**0.0173**	0.4299 ±0.0176	**0.4344** ±**0.0149**	0.4173 ±0.0162	0.4290 ±0.0145	**0.4405** ±**0.0124**	**0.4424** ±**0.0127**	0.4279 ±0.0149
ZDT2	0.0215 ±0.0069	**0.0337** ±**0.0095**	**0.0383** ±**0.0087**	0.0317 ±0.0080	**0.4054** ±**0.0166**	0.0468 ±0.0114	**0.4178** ±**0.0125**	0.4041 ±0.0184
ZDT3	0.3040 ±0.0674	**0.3156** ±**0.0565**	0.2799 ±0.0523	**0.2895** ±**0.0533**	**0.3367** ±**0.0380**	0.3177 ±0.0471	0.3372 ±0.0403	**0.3865** ±**0.0373**

Table 2: HV values for three-objective DTLZ test problems

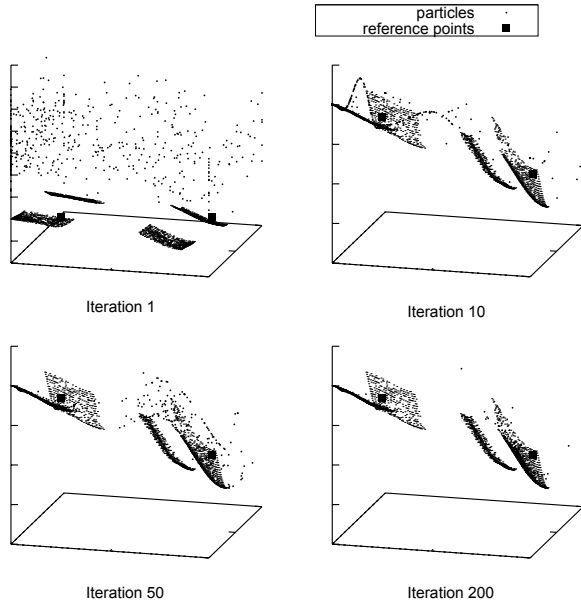| Pop Size / Problem	250		500		750	
	maximinPSO	NSPSO	maximinPSO	NSPSO	maximinPSO	NSPSO
DTLZ6	**0.0803** ±**0.0247**	0.0441 ±0.0093	**0.0544** ±**0.0111**	0.0542 ±0.0107	**0.0665** ±**0.0097**	0.0628 ±0.0125
DTLZ7	0.2167 ±0.0548	**0.2344** ±**0.0465**	**0.2994** ±**0.0073**	0.2802 ±0.0033	**0.3846** ±**0.0154**	0.3214 ±0.0157

Figure 11: Snapshots of maximinPSO on DTLZ7 at 1, 10, 50 and 200 iterations

5.5) for DTLZ7. They all had a spread $\delta = 0.001$ with no bias.

In the two-objective problems a population size less than 50 failed to reliably find solutions on the Pareto front near the regions denoted by the reference points. Therefore we set the population size to 50 and above for the two-objective problems. Similar observation was noted for the three-objective problem. Here the minimum population size was 250.

It can be seen in Table 1 that on average for a two-objective problem maximinPSO was better than NSPSO. For the three-objective problems the maximinPSO on average gave a better HV values than NSPSO (Table 2). We also observed that a larger population is needed to find solutions for a three-objective problem than for a two-objective problem. This was the case if a better spread of solutions is also required.

The results of this comparison experiments showed that given the same set of reference points and spread requirements, the HV metric can be used to provide a meaningful comparison between two user-preference based EMO algorithms.

6. CONCLUSION AND FUTURE WORK

In this study we have shown that a reference point based guidance can be successfully integrated into two PSO algorithms. The DM can specify varying spread as desired near multiple reference points. Bias on certain objectives can be also achieved. Our experiments showed that the proposed multi-objective PSO algorithms incorporating preferences are able to guide the search to several preferred areas of a Pareto front for both two and three-objective problems in a single run. The HV values demonstrated to be a useful metric for comparing user-preference based EMO algorithms.

The reference point is just one preference method that can be integrated into multi-objective PSO algorithms. We plan to extend this research by integrating other approaches like the light beam search. We also would like to investigate other suitable metrics, which can be used for comparing user-preference based EMO algorithms.

7. REFERENCES

[1] R. Balling. The maximin fitness function; multi-objective city and regional planning. In *Evolutionary Multiobjective Optimization (EMO)*, volume 2632 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2003.

[2] J. Branke, T. Kaußler, and H. Schmeck. Guidance in Evolutionary Multi-Objective Optimization. *Advances in Engineering Software*, 32:499–507, 2001.

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

[4] K. Deb. Solving goal programming problems using multi-objective genetic algorithms. In *Congress on Evolutionary Computation (CEC)*, 1999.

[5] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

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

[7] 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, New York, NY, USA, 2007. ACM Press.

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

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

[10] 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.

[11] 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, Boston, 2002. 496 pages. ISBN 1-4020-7128-0.

[12] M. Emmerich, N. Beume, and B. Naujoks. An emo algorithm using the hypervolume measure as selection criterion. In *Evolutionary Multi-Criterion Optimization (EMO)*, Lecture Notes in Computer Science (LNCS), vol. 3410, pages 62–76, Berlin, 2005. Springer.

[13] A. Jaszkiewicz and R. Slowinski. The light beam search approach -an overview of methodology and applications. *European Journal of Operational Research*, 113(2):300–314, 1999.

[14] J. Kennedy and R. C. Eberhart. *Swarm intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

[15] J. Knowles and D. Corne. On metrics for comparing non-dominated sets. In *Congress on Evolutionary Computation (CEC)*, 2002.

[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] A. Messac and C. A. Mattson. Normal constraint method with guarantee of even representation of complete pareto frontier. *American Institute of Aeronautics and Astronautic (AIAA) Journal*, 2004.

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

[20] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, July 1994.

[21] M. Reyes-Sierra and C. A. Coello Coello. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*, 2(3):287–308, 2006.

[22] L. Thiele, K. Miettinen, P. Korhonen, and J. Molina. A preference-based interactive evolutionary algorithm for multiobjective optimization. In *Helsinki School of Economics Technical Report Number W-412*, 2007.

[23] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.