A MOPSO Algorithm Based Exclusively on Pareto Dominance Concepts

Julio E. Alvarez-Benitez*, Richard M. Everson and Jonathan E. Fieldsend**

Department of Computer Science, University of Exeter, UK

Revision: 1.13
22:25 13th October 2004

Abstract In extending the Particle Swarm Optimisation methodology to multi-objective problems it is unclear how global guides for particles should be selected. Previous work has relied on metric information in objective space, although this is at variance with the notion of dominance which is used to assess the quality of solutions. Here we propose methods based exclusively on dominance for selecting guides from a non-dominated archive. The methods are evaluated on standard test problems and we find that probabilistic selection favouring archival particles that dominate few particles provides good convergence towards and coverage of the Pareto front. We demonstrate that the scheme is robust to changes in objective scaling. We propose and evaluate methods for confining particles to the feasible region, and find that allowing particles to explore regions close to the constraint boundaries is important to ensure convergence to the Pareto front.

145 words

1 Introduction

Evolutionary algorithms (EA) have been used since the mid-eighties to solve complex single and multi-objective optimisation problems (see, for example, [1,2,3]). More recently the Particle Swarm Optimisation (PSO) heuristic, inspired by the flocking and swarm behaviour of birds, insects, and fish schools has been successfully used for single objective optimisation, such as neural network training and non-linear function optimisation [4]. Briefly, PSO maintains a balance between exploration and exploitation in a population (swarm) of solutions by moving each solution (particle) towards both the global best solution located by the swarm so far and towards the best solution that the particular particle has so far located. The global best and personal best solutions are often called *guides*.

Since PSO and EA algorithms have structural similarities (such as the presence of a population searching for optima and information sharing between population members) it seems a natural progression to extend PSO to multi-objective problems (MOPSO). Some attempts in this direction have been made with promising results such as [5,6,7,8,9]. In the most recent heuristics the guides are selected from the set of non-dominated solutions found so far. However, in

* Supported by and currently with Banco de la República, Colombia.
** Supported by EPSRC, grant GR/R24357/01.

a multi-objective problem each of non-dominated solutions is a potential global guide and there are many ways of selecting a guide from among them for each particle in the swarm. Heuristics to date have relied on proximity in objective space to determine this selection, however the relative weightings of the objectives are *a priori* unknown and the use of metric information in objective space is at variance with the notion of dominance that is central to the definition of Pareto optimality. In this paper we propose and examine MOPSO heuristics based entirely on Pareto dominance concepts. The manner in which particles are constrained to lie within the search space can have a marked effect on the optimisation efficiency: the other central purpose of this paper is to propose and compare constraint methods.

We start by briefly reviewing basic definitions of multi-objective problems and Pareto concepts (section 2), after which we describe the single objective PSO methodology in section 3. The multi-objective PSO algorithm is presented in section 4, and we present and evaluate methods for selecting guides here. Techniques for confining particles to the feasible region are described and evaluated in section 5. Finally, conclusions are drawn in section 6.

2 Dominance and Pareto optimality

In a multi-objective optimisation problem we seek to simultaneously extremise D objectives: $y_i = f_i(\mathbf{x})$, where $i = 1, \ldots, D$ and where each objective depends upon a vector $\mathbf{x}$ of K parameters or decision variables. The parameters may also be subject to the J constraints: $e_j(\mathbf{x}) \geq 0$ for $j = 1, \ldots, J$.

Without loss of generality it is assumed that these objectives are to be minimised, as such the problem can be stated as:

$$\text{minimise } \mathbf{y} = \mathbf{f}(\mathbf{x}) \equiv (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_D(\mathbf{x})) \tag{1}$$

$$\text{subject to } \mathbf{e}(\mathbf{x}) \equiv (e_1(\mathbf{x}), e_2(\mathbf{x}), \ldots, e_J(\mathbf{x})) \geq \mathbf{0}. \tag{2}$$

A decision vector $\mathbf{u}$ is said to *strictly dominate* another $\mathbf{v}$ (denoted $\mathbf{u} \prec \mathbf{v}$) if $f_i(\mathbf{u}) \leq f_i(\mathbf{v})$ $\forall i = 1, \ldots, D$ and $f_i(\mathbf{u}) < f_i(\mathbf{v})$ for some i; less stringently $\mathbf{u}$ *weakly dominates* $\mathbf{v}$ (denoted $\mathbf{u} \preceq \mathbf{v}$) if $f_i(\mathbf{u}) \leq f_i(\mathbf{v})$ for all i. A set of decision vectors is said to be a *non-dominated set* if no member of the set is dominated by any other member. The *true* Pareto front, $\mathcal{P}$, is the non-dominated set of solutions which are not dominated by any feasible solution.

3 Particle Swarm Optimisation – PSO

The particle swarm optimisation method evolved from a simple simulation model of the movement of social groups such as birds and fish [4], in which it was observed that local interactions underlie the group behaviour and individual members of the group can profit from the discoveries and experiences of other members. In PSO each solution (particle) $\mathbf{x}_n$ in the swarm of N particles is endowed with a velocity which determines its location at the next time step:

$$\mathbf{x}_n^{(t+1)} = \mathbf{x}_n^{(t)} + \chi \mathbf{v}_n^{(t)} + \boldsymbol{\epsilon}^{(t)} \tag{3}$$

where $\chi \in [0, 1]$ is a constriction factor which controls the velocity's magnitude; in the work reported here $\chi = 1$. The final term in (3) is a small stochastic perturbation, known as the *turbulence factor*, added to the position to help prevent the particle becoming stuck in local minima and to promote wide exploration of the decision space. Although originally introduced as a normal perturbation [8], here a perturbation to each dimension was added with probability 0.01 and ϵ_k itself was a perturbation from a Laplacian density $p(\epsilon_k) \propto e^{-|\epsilon_k|/\beta}$ with $\beta = 0.1$. The Laplacian distribution yields occasional large perturbations thus enabling wider exploration.

The velocities of each particle are modified to fly towards two different guides: their personal best, $\mathbf{P}_n$, for exploiting the best results found so far by each of the particles, and the global best, $\mathbf{G}$, the best solution found so far by the whole swarm for encouraging further exploration and information sharing between the particles. This is achieved by updating the K components of each particle's velocity as follows:

$$v_{nk}^{(t+1)} = wv_{nk}^{(t)} + c_1 r_1 (P_{nk} - x_{nk}^{(t)}) + c_2 r_2 (G_{nk} - x_{nk}^{(t)}) \tag{4}$$

r_1 and r_2 are two uniformly distributed random numbers in the range $[0, 1]$. The constants c_1 and c_2 control the effect of the personal and global guides, and the parameter w, known as the *inertia*, controls the trade-off between global and local experience; large w motivates global exploration by giving large weight to the current velocity. In the work reported here $c_1 = c_2 = 1$ and $w = 0.5$. The global guide carries a subscript n because for multi-objective PSO a (possibly different) global guide is associated with each particle; this is in contrast to uni-objective PSO in which there is a single global guide, namely the best solution located so far.

4 Multi-objective PSO

The main difficulty in extending PSO to multi-objective problems is to find the best way of selecting the guides for each particle in the swarm; the difficulty is manifest as there are no clear concepts of personal and global bests that can be clearly identified when dealing with D objectives rather than a single objective. Previous MOPSO implementations [5,6,7,8,9,10] have all used metrics in objective space (either explicitly or implicitly) in the selection of guides – thus making them susceptible to different scalings in objective space.

The algorithms we propose here are similar to recent MOPSO algorithms [7,8,9,10] in that they use an archive or repository, A, which contains the non-dominated solutions found by the algorithm so far. We emphasise that we do not

Algorithm 1 Multi-objective PSO.

1 :	$A := \emptyset$	*Initially empty archive*
2 :	$\{\mathbf{x}_n, \mathbf{v}_n, \mathbf{G}_n, \mathbf{P}_n\}_{n=1}^{N} :=$ `initialise`()	*Random locations and velocities*
3 :	for $t := 1 : G$	*G generations*
4 :	for $n := 1 : N$	
5 :	for $k := 1 : K$	*Update velocities and positions*
6 :	$v_{nk} := wv_{nk} + r_1(P_{nk} - x_{nk}) + r_2(G_{nk} - x_{nk})$	
7 :	$x_{nk} := x_{nk} + v_{nk} + \epsilon$	
8 :	end	
9 :	$\mathbf{x}_n :=$ `enforceConstraints`$(\mathbf{x}_n)$	
10 :	$\mathbf{y}_n := \mathbf{f}(\mathbf{x}_n)$	*Evaluate objectives*
11 :	if $\mathbf{x}_n \not\preceq \mathbf{u} \;\forall\, \mathbf{u} \in A$	*Add non-dominated $\mathbf{x}_n$ to A*
12 :	$A := \{\mathbf{u} \in A \,\vert\, \mathbf{u} \not\prec \mathbf{x}_n\}$	*Remove points dominated by $\mathbf{x}_n$*
13 :	$A := A \cup \mathbf{x}_n$	*Add $\mathbf{x}_n$ to A*
14 :	end	
15 :	end	
16 :	if $\mathbf{x}_n \preceq \mathbf{P}_n \vee (\mathbf{x}_n \not\prec \mathbf{P}_n \wedge \mathbf{P}_n \not\prec \mathbf{x}_n)$	*Update personal best*
17 :	$\mathbf{P}_n := \mathbf{x}_n$	
18 :	end	
19 :	$\mathbf{G}_n :=$ `selectGuide`$(\mathbf{x}_n, A)$	
20 :	end	

restrict the size of A by gridding, clustering or niching (as done in, for example, [10]) as that may lead to oscillation or shrinking of the Pareto front [11,12].

At the start of the optimisation, which is outlined in Algorithm 1, A is empty and the locations and velocities of the N particles are initialised randomly. The personal bests for each particle are initialised to be the starting location, $\mathbf{P}_n = \mathbf{x}_n$; likewise the global guide for each particle is initialised to be its initial location: $\mathbf{G}_n = \mathbf{x}_n$.

At each generation t the velocities $\mathbf{v}_n$ and locations $\mathbf{x}_n$ of each particle are updated according to (4) and (3) (lines 5–8 of Algorithm 1). Following updating, it is possible that the particle positions lie outside the region of feasible solutions. In this case it must be constrained to the feasible region; this is indicated in the Algorithm 1 by the function `enforceConstraints`, and we discuss methods for enforcing the constraints in section 5. With $\mathbf{x}_n$ in the feasible region the objectives may be evaluated (line 10), and any solutions which are not dominated by any member of the archive are added to A (line 13) and any elements of A which are dominated by $\mathbf{x}_n$ are deleted from A, thus ensuring that A is a non-dominating set.

The crucial parts of the MOPSO algorithm are selecting the personal and global guides. Selection of $\mathbf{P}_n$ is straightforward: if the current position of the n-th particle, $\mathbf{x}_n$, weakly dominates $\mathbf{P}_n$ or $\mathbf{x}_n$ and $\mathbf{P}_n$ are mutually non-dominating, then $\mathbf{P}_n$ is set to the current position (lines 16–18). Since members of A are mutually non-dominating and no member of the archive is dominated by any $\mathbf{x}_n$, so that in some senses the archive is globally 'better' than each member of the swarm, all the members of A are candidates for the global guide and we now

Algorithm 2 ROUNDS selection of global guides.

1 :	$X' := X$	*Swarm*
2 :	$A' := \emptyset$	*Candidate guides*
3 :	while $\lvert X'\rvert > 0$	
4 :	if $\lvert A'\rvert = 0$, then $A' = A$	*New round: all A are candidates*
5 :	for $\mathbf{a} \in A'$	
6 :	$X_{\mathbf{a}} := \{\mathbf{x} \in X' \mid \mathbf{a} \prec \mathbf{x}\}$	*Swarm members dominated by* $\mathbf{a}$
7 :	end	
8 :	$\mathbf{a}^\star := \arg\min_{\mathbf{a}\in A'}(\lvert X_{\mathbf{a}}\rvert)$	$\mathbf{a}^\star$ *dominates fewest particles*
9 :	$\mathbf{x}_n := \texttt{choose}(X_{\mathbf{a}^\star})$	*Random selection from* $X_{\mathbf{a}^\star}$
10 :	$\mathbf{G}_n := \mathbf{a}^\star$	
11 :	$X' := X' \setminus \mathbf{x}_n$	*Guide selected for* $\mathbf{x}_n$
12 :	$A' := A' \setminus \mathbf{a}^\star$	*Assigned, so delete from candidates*
11 :	end	

present alternative ways of selecting a global guide *for each* particle in the swarm from A.

4.1 Selecting global guides

Here we focus on methods of selecting global guides which are based solely on Pareto dominance and do not attempt to use metric information in objective space. Three alternatives are examined: ROUNDS, which is most complex and explicitly promotes diversity in the population; RANDOM, which is simple and promotes convergence; and PROB, which is a weighted probabilistic method and forms a compromise between RANDOM and ROUNDS. It may be supposed that the archive members which dominate particle $\mathbf{x}_n$ would be better global guides than those archive members which do not, and each of these schemes is based on the idea of selecting a guide for a particle from the members of the archive which dominate the particle.

ROUNDS The idea underlying this method is that in order to promote diversity in the population by attracting the swarm towards sparsely populated regions, members of the archive that dominate the fewest $\mathbf{x}_n$ should be preferentially assigned as global guides. As shown in Algorithm 2, this is achieved by first locating the member of the archive $\mathbf{a}^\star$ which dominates the fewest particles, which is then assigned to be the guide of one of the particles in $X_{\mathbf{a}^\star}$, the set of particles which it dominates. Having assigned $\mathbf{a}^\star$ as a guide, it is removed from consideration as a possible guide until all the other archive members have been assigned a particle to guide and a new round begins (line 4). Clearly, the algorithm can be coded more efficiently than the outlined in Algorithm 2, however, the procedure can be computationally expensive when the archive is large.

RANDOM While the ROUNDS methods associates a member of the archive with one of the particles in the swarm that it dominates, the RANDOM selection

Table 1. Test problems DTLZ1, DTLZ2 & DTLZ3 of [13] for 3 objectives.

DTLZ1	$f_1(\mathbf{x}) = \frac{1}{2}x_1x_2\left(1+g\left(\mathbf{x}\right)\right)$ $f_2(\mathbf{x}) = \frac{1}{2}x_1\left(1-x_2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $f_3(\mathbf{x}) = \frac{1}{2}\left(1-x_1\right)\left(1+g\left(\mathbf{x}\right)\right)$ $g\left(\mathbf{x}\right) = 100[\lvert\mathbf{x}\rvert - 2 + \sum_{k=3}^{K}\left(x_k-0.5\right)^2 - \cos\left(20\pi\left(x_k-0.5\right)\right)]$ $0 \le x_k \le 1$, for $k = 1, 2, \ldots, K$, $K = 7$
DTLZ2	$f_1(\mathbf{x}) = \frac{1}{2}x_1x_2\left(1+g\left(\mathbf{x}\right)\right)$ $f_2(\mathbf{x}) = \frac{1}{2}x_1\left(1-x_2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $f_3(\mathbf{x}) = \frac{1}{2}\left(1-x_1\right)\left(1+g\left(\mathbf{x}\right)\right)$ $g\left(\mathbf{x}\right) = \sum_{k=3}^{K}\left(x_k-0.5\right)^2$ $0 \le x_k \le 1$, for $k = 1, 2, \ldots, K$, $K = 12$
DTLZ3	$f_1(\mathbf{x}) = \cos\left(x_1\pi/2\right)\cos\left(x_2\pi/2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $f_2(\mathbf{x}) = \cos\left(x_1\pi/2\right)\sin\left(x_2\pi/2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $f_3(\mathbf{x}) = \sin\left(x_1\pi/2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $g\left(\mathbf{x}\right) = 100[\lvert\mathbf{x}\rvert - 2 + \sum_{k=3}^{K}\left(x_k-0.5\right)^2 - \cos\left(20\pi\left(x_k-0.5\right)\right)]$ $0 \le x_k \le 1$, for $k = 1, 2, \ldots, K$, $K = 7$

methods focuses on the particle $\mathbf{x}_n$ and selects a guide from among the archive members that dominate $\mathbf{x}_n$. If $A_{\mathbf{x}} = \{\mathbf{a} \in A \mid \mathbf{a} \prec \mathbf{x}\}$ is the set of archived points that dominate $\mathbf{x}$, then the RANDOM selection method simply chooses an element of $A_{\mathbf{x}_n}$ with equal probability to be the guide for $\mathbf{x}_n$. If $\mathbf{x}_n \in A$ then, clearly, $A'_{\mathbf{x}_n}$ is empty, so in this case a guide is selected from the entire archive. Thus

$$\mathbf{G}_n = \begin{cases} \mathbf{a} \in A \text{ with probability } \lvert A\rvert^{-1} & \text{if } \mathbf{x}_n \in A \\ \mathbf{a} \in A_{\mathbf{x}_n} \text{ with probability } \lvert A_{\mathbf{x}_n}\rvert^{-1} & \text{otherwise.} \end{cases} \tag{5}$$

PROB The RANDOM selection method gives equal probability of being chosen as the guide to all archive members dominating a particle. However, archive members in sparsely populated regions of the front and towards the 'edges' of the front are likely to dominate fewer particles than those in well populated regions or close to the centre of the front. To guide the search towards the sparse regions and edges, we adapt the RANDOM method to favour archive members that dominate the least points. Let $X_{\mathbf{a}} = \{\mathbf{x} \in X \mid \mathbf{a} \prec \mathbf{x}\}$ be the set of particles dominated by $\mathbf{a}$. Then guides are chosen as:

$$\mathbf{G}_n = \begin{cases} \mathbf{a} \in A \text{ with probability } \propto \lvert X_{\mathbf{a}}\rvert^{-1} & \text{if } \mathbf{x}_n \in A \\ \mathbf{a} \in A_{\mathbf{x}_n} \text{ with probability } \propto \lvert X_{\mathbf{a}}\rvert^{-1} & \text{otherwise.} \end{cases} \tag{6}$$

The PROB selection method thus combines the intention behind ROUNDS with the simplicity of RANDOM. With efficient data structures [12] or relatively small populations the computational expense in calculating $\lvert X_{\mathbf{a}}\rvert$ and $A_{\mathbf{x}}$ is not exorbitant and can be efficiently incorporated into the updating of A (lines 11–14 of Algorithm 1).

Table 2. GD(A) and $\mathcal{V}_{\mathcal{P}}(A)$ measures for the methods proposed to select guides. The best value across methods is highlighted in bold.

	GD(A)				$\mathcal{V}_{\mathcal{P}}(A)$			
DTLZ1	MT	ROUNDS	RANDOM	PROB	MT	ROUNDS	RANDOM	PROB
Best	0.0002	0.0048	3.77×10^{-5}	$\mathbf{3.15 \times 10^{-5}}$	0.9992	0.9823	**0.9997**	**0.9997**
Worst	0.7481	0.1761	**0.031**	0.0349	0.9270	0	0.9744	**0.9796**
Average	0.1016	0.0696	$\mathbf{3.10 \times 10^{-3}}$	5.55×10^{-3}	0.9824	0.5201	0.9965	**0.9974**
Median	0.0303	0.0656	3.23×10^{-4}	$\mathbf{1.41 \times 10^{-4}}$	0.9947	0.4952	0.9979	**0.9992**
S. dev.	0.2068	0.0572	7.30×10^{-3}	0.0114	0.0231	0.3413	0.0057	0.0047
DTLZ3								
Best	0.003	0.001	$\mathbf{4.81 \times 10^{-5}}$	5.77×10^{-5}	0.9921	0.9972	0.9979	**0.9981**
Worst	0.2195	1.0244	**0.1446**	0.2621	0	0	**0.7798**	0.704
Average	0.0413	0.2044	**0.0217**	0.0305	0.8743	0.1602	0.9352	**0.945**
Median	0.0134	0.1343	$\mathbf{1.44 \times 10^{-3}}$	1.52×10^{-3}	0.9635	0	0.9486	**0.9946**
S. dev.	0.0627	0.2451	0.0429	0.0687	0.2290	0.316	0.0674	0.086

4.2 Experiments

We compared the efficiency of the global guide selection methods on standard test problems DTLZ1–DTLZ3 [13], whose definitions for three objectives are provided in Table 1.

Unlike single objective problems, solutions to multi-objective optimisation problems can be assessed in several different ways. Here we use the Generational Distance (GD) introduced in [14] and used by others (e.g., [10]) as a measure of the mean distance between elements of the archive and the true Pareto front:

$$GD(A) = \left[\frac{1}{|A|} \sum_{\mathbf{a} \in A} d(\mathbf{a})^2 \right]^{\frac{1}{2}} \tag{7}$$

where $d(\mathbf{a})$ is the shortest Euclidean distance between $\mathbf{a}$ and the front $\mathcal{P}$. Clearly, this measure depends on the relative scaling of the objective functions, however, it yields a fair comparison here because the objectives for the DTLZ test functions have similar ranges.

An alternative measure which also measures the spread of the solutions found across the front is the volume measure, $\mathcal{V}_{\mathcal{P}}(A)$, which is defined as the fraction of the minimum axis-parallel hyper-rectangle containing $\mathcal{P}$ which is dominated by both $\mathcal{P}$ and A. It may be straightforwardly calculated by Monte Carlo sampling; see [12] for details.

We present results of two sets of experiments performed, firstly, in order to evaluate the selection methods and, secondly, illustrate the robustness of the selected method to rescaling of the objectives.

To evaluate the selection methods proposed, we assessed the fronts generated by the ROUNDS, RANDOM and PROB methods together with the fronts generated by an implementation of the Mostaghim & Teich's MOSPO (designated MT in

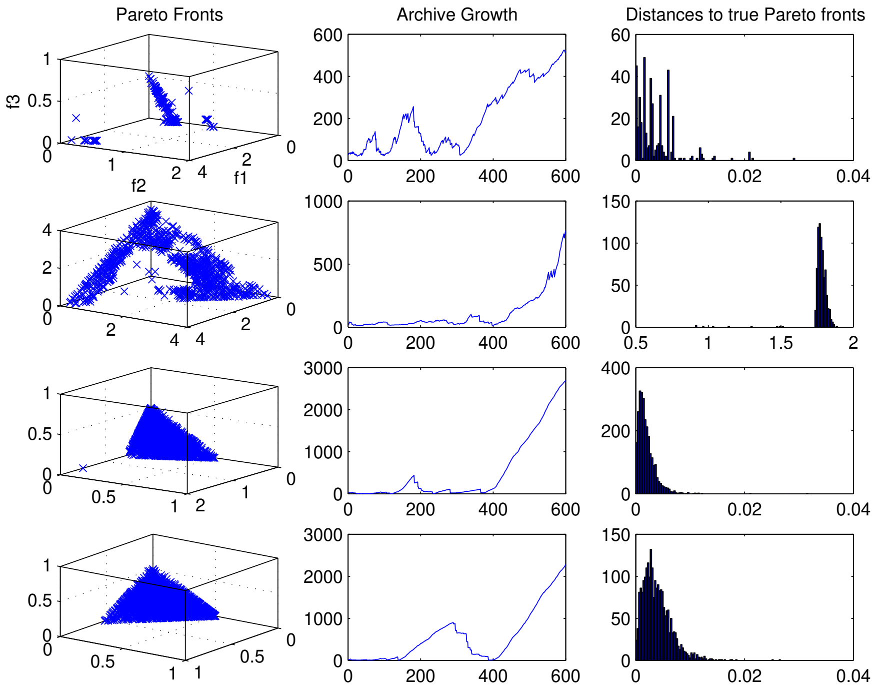

Figure 1. Archives, archive growth and histograms of distances to the DTLZ1 Pareto fronts corresponding to the median result of the GD metric for (top to bottom) MT, ROUNDS, RANDOM, PROB selection methods. (The more distant clusters of particles were cut from the MT histogram for visualisation purposes.)

this paper) [9]. In order to permit fair comparisons we did not limit the archive size in the MT algorithm. During early experimentation it was observed that more rapid convergence may be achieved (for all algorithms, including MT) by initially promoting more aggressive search (wider exploration); in all the work reported here this was done by ignoring the contribution from global guides ($c_2 = 0$ in (4)) when $|A| < 100$. For all methods $N = 100$ particles comprised the swarm and the algorithms were run for 600 generations.

Table 2 shows the mean, standard deviation, median, worst and best values of the $GD(A)$ and $\mathcal{V}_{\mathcal{P}}(A)$ measures over 20 different random initialisations of each method. On the basis of these results it is difficult to distinguish between the MT and ROUNDS methods, but it is clear that the RANDOM and PROB methods are generally superior to both of them. In terms of the GD measure the RANDOM selection scheme appears to be slightly superior to the PROB method, but the $\mathcal{V}_{\mathcal{P}}(A)$ measure favours the PROB method. This reflects the explicit promotion of search towards edges and sparsely populated regions by PROB, resulting in better coverage of the front, which is measured by $\mathcal{V}_{\mathcal{P}}(A)$, rather than merely distance

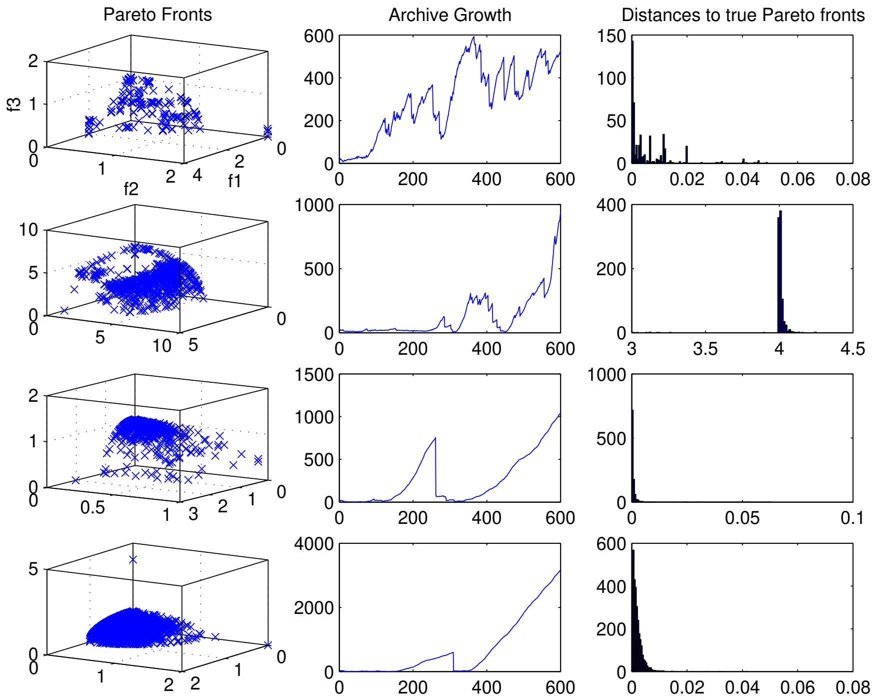

Figure 2. Archives, archive growth and histograms of distances to the DTLZ3 Pareto fronts corresponding to the median result of the GD measure for (top to bottom) MT, ROUNDS, RANDOM, PROB selection methods. (The more distant clusters of particles were cut from the MT histogram for visualisation purposes.)

from the front which is quantified by $GD(A)$. The fronts from the 20 different runs of RANDOM and PROB methods were compared pairwise by calculating the volume in objective space dominated by one front but not by the other [12]: in over 60% of the comparisons PROB outperformed RANDOM.

Figures 1 and 2 show for DTLZ1 and DTLZ3 respectively the archives, archive growth and histograms of the distances from $\mathcal{P}$ for the median run according to the GD measure. For both problems it is apparent that RANDOM selection achieves tightly grouped solutions close to the true front, but the PROB scheme yields a better coverage of particles. These figures also show that the PROB and RANDOM schemes both result in significantly larger archives than MT and ROUNDS. It is also interesting to note that although ROUNDS often fails to converge well it does provide good coverage.

These results, along with the preference for an algorithm promoting diversity, lead us to choose PROB selection as the best alternative and from now on we concentrate on this method.

Table 3. Comparison, using the GD measure, between the PROB and MT selection methods with and without scaling of objectives. Δ indicates the percentage change between scaled and unscaled quantities.

	PROB			MT		
$DTLZ2$	**unscaled**	**rescaled**	Δ	**unscaled**	**rescaled**	Δ
Best	5.79×10^{-4}	5.87×10^{-4}	+1.38%	5.2×10^{-3}	6.4×10^{-3}	+18.75%
Worst	1.10×10^{-3}	9.95×10^{-4}	-9.54%	0.0189	0.0174	-7.91%
Average	7.18×10^{-4}	7.06×10^{-4}	-1.67%	0.0113	0.0120	+5.83%
Median	6.64×10^{-4}	6.84×10^{-4}	+2.92%	0.0111	0.0119	+6.72%
S. dev.	1.44×10^{-4}	1.03×10^{-4}	-28.4%	3.8×10^{-3}	3.1×10^{-3}	-18.42%

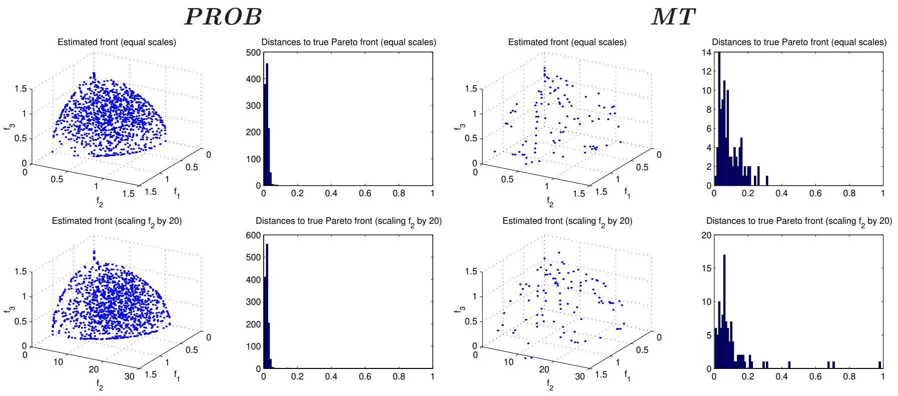

Figure 3. Pareto fronts and histograms of distances to the true Pareto fronts corresponding for unscaled (top) and with f_2 rescaled by 20 (bottom) using the PROB (left) and MT (right) selection rules.

As mentioned previously, the selection methods introduced here do not depend upon metric information in objective space and thus may be expected to be unaffected by the scales on which the objectives are measured. To illustrate the robustness of the method we compared 20 optimisations of the DTLZ2 test problem in which one of the objectives was rescaled with 20 optimisations without rescaling. (All optimisations started from different random initial particle locations.) On the i-th optimisation one of the objectives (chosen cyclically) was multiplied by $(i + 1)$. The fronts obtained after 45 generations were assessed using the $GD(A)$ measure, but to facilitate comparison the relevant objective was rescaled back to the usual scale.

Average results are shown in Table 3 and Figure 3 compares the estimated Pareto fronts for runs in which f_2 was multiplied by 20 with fronts from unscaled runs. As the table shows, the optimisations using PROB selection are unaffected by the rescaling, in contrast to the MT method which relies on objective space distances for its selection of guides.

5 Keeping particles within the search space

The velocity and position updates (4) and (3) are liable to cause particles to exceed the boundaries of the feasible regions and both single and multi-objective PSO algorithms must be modified to keep the particles within the constraints. The manner in which this is done may have a great impact on the performance of the algorithm as it affects the way in which particles move around the search space and it is particularly important when the optimum decision variables values lie on or near to the boundaries. In Algorithm 1 this is delegated to the `enforceConstraints` function and in this section we discuss methods for ensuring that particles remain in the feasible region.

A number of alternatives for this have been proposed: A straightforward method [10] is to truncate the location at the exceeded boundary at this generation and reflect the velocity in the boundary so that the particle moves away at the next generation. An alternative [18] is to resample the stochastic terms in the velocity update formula (4) until a feasible position is achieved. Other schemes rely on limiting the magnitude of the velocities, either explicitly [15] or by modifying the constriction factor χ and the other 'constants' w, c_2 and c_2 appearing in the update equations [16].

Other methods may involve using *a priori* knowledge about the particular problem being optimised. For example, in [9] the trespass rule for the first $D-1$ parameters was different from the remainder [17], exploiting the knowledge that in the DTLZ test functions the first $D-1$ parameters determine the coverage of the front while the remainder determine the distance from the front. This approach does improve the quality of solutions but is not used here because we are interested in examining generic methods and not those dependant on prior knowledge about the functions to be optimised.

Here we examine four methods of constraining the particles. In describing these we assume that the constraints are constraints on individual parameters (i.e., constraints of the form $L \leq x_k \leq U$ for some upper and lower limits, L and U), however, they are easily generalised to oblique or curved feasible regions.

TRC Particles exceeding a boundary are truncated at the boundary for this generation and the velocity is reflected in the boundary so that they tend to move away on the next update [10].

SHR In reflecting the particle at the boundary the TRC method endows the particle at the next generation with a velocity *away* from the boundary, which can be detrimental to finding optima if the optimal decision parameters lie on the boundary. To combat this the SHR method shrinks the magnitude of the velocity vector of the particle so that it arrives exactly at the boundary, but does not alter its direction, permitting the particle to stay in the vicinity of the boundary. Suppose that the k-th component of the particle's position exceeds a boundary at U, then the SHR scheme sets

$$\mathbf{x}_n^{(t+1)} = \mathbf{x}_n^{(t)} + \sigma(\chi \mathbf{v}_n^{(t)} + \boldsymbol{\epsilon}) \tag{8}$$

with

$$\sigma = \frac{x_{nk}^{(t)} - U}{\chi v_{nk}^{(t)} + \epsilon_k} \tag{9}$$

Note that, in contrast to the other methods discussed here, the SHR scheme affects all components of the particle's position, rather than just the component that has exceeded a constraint.

RES The resampling method merely resamples the stochastic variables r_1 and r_2 in (4) and ϵ_k in (3) for each velocity component until the particle location is in the feasible region [18].

EXP The final method we examine updates the position component with a random draw when that particular component, say $x_k^{(t)}$, would have been updated to a position beyond a boundary at, say, b. For convenience, suppose $x_k^{(t)} < U$. In this case we sample from a truncated exponential distribution oriented so that there is a high probability of samples close to the boundary and a lower probability of samples at the current position $x_k^{(t)}$. More precisely a new location $x_k^{(t+1)}$ is drawn with probability:

$$p(x_k^{(t+1)}) \propto \begin{cases} \exp\left\{-\frac{|U-x_k^{(t+1)}|}{|U-x_k^{(t)}|}\right\} & \text{if } x_k^{(t)} \leq x_k^{(t+1)} \leq U \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

with obvious modifications if $U < x_k^{(t)}$. In a similar manner to the SHR method this scheme tends to allow particles that would have exceeded the boundaries to remain close to the boundaries.

5.1 Experiments

To determine the impact of each of the four methods, we compared the fronts located for the DTLZ1 and DTLZ3 problems using each of them in conjunction with the PROB guide selection scheme. The fronts were all assessed against the true Pareto front using the $GD(A)$ and $\mathcal{V}_{\mathcal{P}}(A)$ measures. Each version was run 20 times (using the same parameters as described above) and the results are presented in Table 4.

It is clear that the SHR method, which shrinks the velocity vector so that the particle arrives exactly at the boundary, yields superior results on both test problems according to both the generational distance and volume measures. The EXP method, which resamples giving preference to locations close to the boundary is the next best, while the two methods that tend to move a particle away from the boundary, TRC and RES, give the poorest results. Indeed RES and TRC occasionally prevent convergence.

Further insight into the way in which the RES and SHR methods behave may be gained by examining the trajectory of a single particle, as shown in Figure

Table 4. Generational distance and $\mathcal{V}_{\mathcal{P}}(A)$ measures for constraint handling methods compared on DTLZ1 & DTLZ3

	$GD(A)$				$\mathcal{V}_{\mathcal{P}}(A)$			
DTLZ1	SHR	RES	EXP	TRC	SHR	RES	EXP	TRC
Best	$\mathbf{3.77}\times10^{-5}$	1.3588	6.36×10^{-5}	7×10^{-3}	**0.9997**	0.1289	0.9996	0.9957
Worst	**0.0349**	11.8362	0.1996	0.4706	**0.9796**	0	0.6958	0
Average	$\mathbf{5.55}\times10^{-3}$	8.2132	0.0336	0.1747	**0.9974**	0.0084	0.9645	0.6983
Median	$\mathbf{1.41}\times10^{-4}$	8.6158	0.0178	0.2064	**0.9992**	0	0.992	0.8221
S. dev.	0.011426	2.3872	0.05	0.147	0.0047	0.0297	0.0699	0.3035
DTLZ3								
Best	$\mathbf{5.77}\times10^{-5}$	21.75	2.03×10^{-4}	0.0295	**0.9981**	0	0.9965	0.9964
Worst	**0.2621**	41.08	1.5826	4.09	**0.704**	0	0	0
Average	**0.0305**	31.74	0.1539	1.61	**0.945**	0	0.8145	0.2189
Median	$\mathbf{1.52}\times10^{-3}$	32.29	0.0381	1.54	**0.9946**	0	0.9367	0.0955
S. dev.	0.068707	4.25	0.3501	1.21	0.086	0	0.2892	0.2945

4. The figure shows 7 coordinates of a single particle during an optimisation of the DTLZ1 problem. As remarked previously, in this problem the optimum value for variables x_3 to x_7 is 0.5, while $0 \le x_1, x_2 \le 1$ provide coverage of the front when x_3 to x_7 are at their optimum value. It is clear from Figure 4 that the resampling method RES promotes greater movements across the space which may be beneficial for exploration. However during the resampling the particle is pushed away from the optimal locations. In contrast the SHR scheme permits the particle to remain close to the boundaries during the search process.

The DTLZ test problems which we analyse here are special in that the roles of the decision variables may be clearly distinguished. However, it is likely that in real problems optima may lie close to or on the constraint boundaries or these regions will be visited *en route* to the optima and it will be important to permit particles to properly explore these regions.

6 Conclusions

We have examined several methods of choosing global guides in multi-objective extensions of particle swarm optimisers. Unlike previous work, guides are selected without reference to distance information in the objective space, which renders them robust to the relative scalings of the objectives. Indeed, if the relative importance or scales of the objectives were known in advance it might be more straightforward to optimise a single, appropriately weighted, sum of the objectives. Notions of dominance and Pareto optimality are well suited to handling competing objectives whose relative importance is *a priori* unknown and it is therefore natural to eschew metric information in favour of dominance concepts when choosing guides. We find that selecting guides probabilistically from the archive of non-dominated solutions, giving more weight to solutions that dominate few particles, provides both good convergence and widespread coverage.

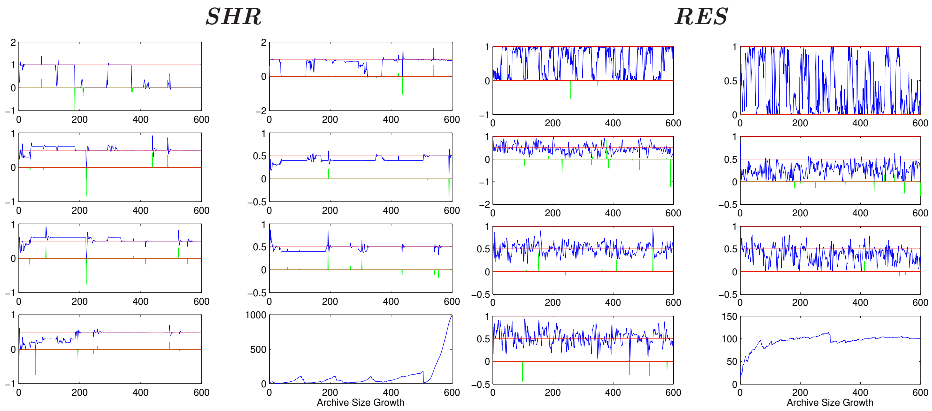

Figure 4. Movements of a particle and growth of the archive during the searching process when using SHR and RES methods on DTLZ1. (x_1 to x_7 are shown top-left, top-right, . . . , to bottom-right.)

This method yields superior performance to an existing MOPSO technique and is robust to changes of scale in the objective functions.

The computation involved in the selection is a little more extensive than other recently proposed schemes (e.g., [9,10]) but is more than compensated for by improved convergence and coverage.

The PROB selection method selects guides with probability inversely proportional to the number of particles the potential guide dominates (c.f., (6)). It would be interesting to examine the performance of an algorithm which selects guides with probability proportional to $|X_{\mathbf{a}}|^{-q}$; as $q \to 0$ the method becomes the RANDOM method, but as q increases additional weight is given to sparse regions. Although this might enable finer control of the convergence and coverage it would introduce an additional parameter to be 'tweaked'.

It was found that the manner in which particles are constrained to the feasible region can vastly affect the performance of a MOPSO. Four methods of constraining particles were examined and it was found that a method which permits particles to remain close to the boundaries enables more rapid location of the Pareto front. We anticipate that careful handling of solutions to allow exploration close to the boundaries will be important not only in MOPSO, but also in other approaches to multi-objective optimisation.

References

1. Srinivas, N., Deb, K.: Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. IEEE Transactions on Evolutionary Computation **2(3)** (1995) 221–248

2. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation **3(4)** (1999) 257–271
3. Laumanns, M., Zitzler, E., Thiele, L.: A Unified Model for Multi-Objective Evolutionary Algorithms with Elitism. In: Proceedings of the 2000 Congress on Evolutionary Computation. (2000) 46–53
4. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of the Fourth IEEE International Conference on Neural Networks. (1995) 1942–1948
5. Hu, X., Eberhart, R.: Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization. In: Proceedings of the 2002 Congess on Evolutionary Computation, IEEE Press (2002)
6. Parsopoulos, K., Vrahatis, M.: Particle Swarm Optimization Method in Multiobjective Problems. In: Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002). (2002) 603–607
7. Coello, C., Lechunga, M.: MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization. In: Proceedings of the 2002 Congress on Evolutionary Computation, IEEE Press (2002) 1051–1056
8. Fieldsend, J., Singh, S.: A Multi-Objective Algorithm based upon Particle Swarm Optimisation, an Efficient Data Structure and Turbulence. In: Proceedings of UK Workshop on Computational Intelligence (UKCI 02). (2002) 37–44
9. Mostaghim, S., Teich, J.: Strategies for Finding Good Local Guides in Multi-Objective Particle Swarm Optimization (MOPSO). In: IEEE 2003 Swarm Intelligence Symposium. (2003) 26–33
10. Coello, C., Pulido, G., Lechunga, M.: Handling Multiple Objectives with Particle Swarm Optimization. IEEE Transactions on Evolutionary Computation **8(3)** (2004) 256–279
11. Hanne, T.: On the convergence of multiobjective evolutionary algorithms. European Journal of Operational Research **117** (1999) 553–564
12. Fieldsend, J., Everson, R., Singh, S.: Using Unconstrained Elite Archives for Multi–Objective Optimisation. IEEE Transactions on Evolutionary Computation **7** (2003) 305–323
13. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Multi–Objective Optimization Test Problems. In: Congress on Evolutionary Computation (CEC'2002). Volume 1. (2002) 825–830
14. Veldhuizen, D.V., Lamont, G.: Multiobjective Evolutionary Algorithms Research: A History and Analysis. Technical Report TR-98-03, Dept. Elect. Comput. Eng., Graduate School of Eng., Air Force Institute Technol., Wright-Patterson AFB, OH (1998)
15. van den Bergh, F.: An Analysis of Particle Swarm Optimizers. PhD thesis, Faculty of Natural and Agricultural Science, University of Pretoria (2001)
16. Clerc, M.: The Swarm and the Queen: Towards a Deterministic and Adaptative Particle Swarm Optimization. In: Proceedings of the Congress on Evolutionary Computation, IEEE Press (1999) 1951–1957
17. Mostaghim, S., Teich, J.: Private communication. (2004)
18. Fieldsend, J.: Multi–Objective Particle Swarm Optimization Methods. Technical Report No. 419, Department of Computer Science, University of Exeter (2004)