On the Effect of Selection and Archiving Operators in Many-Objective Particle Swarm Optimisation

Matthaus M. Woolard
Computer Science
University of Exeter
Exeter, UK
mmw203@exeter.ac.uk

Jonathan E. Fieldsend*
Computer Science
University of Exeter
Exeter, UK
J.E.Fieldsend@exeter.ac.uk

ABSTRACT

The particle swarm optimisation (PSO) heuristic has been used for a number of years now to perform multi-objective optimisation, however its performance on *many-objective* optimisation (problems with four or more competing objectives) has been less well examined. Many-objective optimisation is well-known to cause problems for Pareto-based evolutionary optimisers, so it is of interest to see how well PSO copes in this domain, and how non-Pareto quality measures perform when integrated into PSO. Here we compare and contrast the performance of canonical PSO, using a wide range of many-objective quality measures, on a number of different parametrised test functions for up to 20 competing objectives. We examine the use of eight quality measures as *selection* operators for guides when truncated non-dominated archives of guides are maintained, and as *maintenance* operators, for choosing which solutions should be maintained as guides from one generation to the next. We find that the Controlling Dominance Area of Solutions approach performs exceptionally well as a quality measure to determine archive membership for global and local guides. As a selection operator, the Average Rank and Sum of Ratios measures are found to generally provide the best performance.

Categories and Subject Descriptors

I.2.8 [**Problem solving, control methods and search**]: Heuristic methods

General Terms

Algorithms, Performance, Experimentation

Keywords

Particle swarm optimisation, multi-objective optimisation, selection

*Corresponding author.

1. INTRODUCTION

Since its inception in 1995 [14] the particle swarm optimisation (PSO) heuristic has gained rapid popularity as a technique to facilitate single objective optimisation. Like the standard evolutionary algorithm (EA) methods of genetic algorithms (GAs) and evolution strategies (ESs), PSO was inspired by nature, but instead of evolution it was the flocking and swarming behaviour of birds and insects that motivated its development.

A population (swarm) of individual solutions is maintained in PSO, whose representation is typically a vector of floating point decision parameters, which are used in a solution's (particle's) evaluation. During the optimisation process of PSO (following initialisation), members of this population are flown (have their parameters adjusted) according to their previous 'flying experience'. This flying experience is both in terms of the particle as an individual, and as a member of a wider group (the entire swarm, or a subset of it). The general PSO model implements this by adjusting an individual's decision parameters to make them 'closer' to the decision parameters of two other solutions; a neighbourhood guide (which may be global or local), and the best evaluated position found previously by that individual. A particle's position also includes some temporal adjustment via a *velocity* vector, which tracks the movement the particle made in the previous iteration of the optimiser, and uses this to adjust the particle's position in the current iteration.

A decade ago (circa 2002), researchers began publishing multi-objective (MO) variants of PSO [2, 11, 13, 18] (although an unpublished paper on the area exists from 1999 [15]), typically referred to as MOPSO algorithms. Since these works there has been a large growth in the number and range of MOPSO algorithms published in the literature, which has largely tracked the growth of, and range of, general multi-objective evolutionary algorithms (MOEAs), with comparison/selection/variation operators popularised in the MOEA field rapidly being converted into aspects of MOPSOs when direct analogies could be drawn (e.g. the use of *dominance*, *hypervolume indicator*, *clustering*, *archive maintenance*, *mutation/turbulence operators*, etc.). As the number of distinct MOPSOs has grown, a number of papers have provided overviews of the range of approaches that can be taken, along with some empirical comparisons (e.g. [10, 16, 17, 19]). However there has been relatively little work thus far examining *many*-objective PSO performance (i.e., on problems with four or more objectives) [21, 5].

In this paper we are concerned with the performance of the PSO heuristic on *many*-objective problems, and the effectiveness of different quality measures in this domain [1, 6, 8, 9, 20] when used within PSO. As such, we limit the variation of the baseline optimiser used to a simple PSO, and vary purely the use of these measures for archive maintenance and for selection. The paper proceeds as follows, we first describe canonical PSO, and multi-objective PSO in Sec. 2, we then describe the problems inherent in many-objective optimisation and describe a range of quality measures used in the wider MOEA community to mitigate them in Sec. 3. In Sec. 4 we present the experiments, which compare a range of many-objective quality measures embedded within the PSO heuristic across a range of test problems with up to 20 competing objectives. We analyse the results of these experiments in Sec. 5 and the paper concludes with a discussion in Sec. 6.

2. CANONICAL PSO AND MOPSO

The PSO heuristic was first proposed by Kennedy and Eberhart [14] for the optimisation of continuous non-linear functions. A fixed population of solutions is used, where each solution (or particle) is represented by a point in n-dimensional space. The ith particle is commonly represented as $\mathbf{x}_k = (x_{k,1}, x_{k,2}, \dots x_{k,n})$, and its performance evaluated on a given problem and stored. Each particle maintains knowledge of its best previous evaluated position, represented as $\mathbf{p}_k$, and also has knowledge of the single best solution found so far in some defined neighbourhood, $\mathbf{g}_k$, often this is a global neighbourhood (all particles are considered), however other neighbourhood definitions are also popular. The rate of position change of a particle then depends upon its previous personal best position, its neighbourhood best, and its previous velocity. For particle i this velocity is $\mathbf{v}_k = (v_{k,1}, \dots, v_{k,n})$. The general algorithm for the adjustment of these velocities is:

$$v_{k,j} := wv_{k,j} + c_1 r_1 (p_{k,j} - x_{k,j}) + c_2 r_2 (g_{k,j} - x_{k,j}), \quad (1)$$

and the position is updated as:

$$x_{k,j} := x_{k,j} + \chi v_{k,j}, \quad j = 1, \dots, n, \quad (2)$$

where w, c_1, c_2, $\chi \geq 0$. w is the inertia of a particle, c_1 and c_2 are constraints on the velocity toward local best and neighbourhood best - referred to as the *cognitive* and *social* learning factors respectively, χ is a constraint on the overall shift in position, and $r_1, r_2 \sim \mathcal{U}(0, 1)$. In [14], the final model presented $w, \chi = 1.0$ and $c_1, c_2 = 2.0$.

As discussed in [10], in this classical form of PSO each particle $\mathbf{x}_k$ is flown toward $\mathbf{p}_k$, $\mathbf{g}_k$ and $\mathbf{v}_k$. This, in effect, means that a hypercuboid is generated in solution/particle space, the bounds of which are the sum of the distances from $\mathbf{x}_k$ to the other three guides (weighted by the appropriate multiplier constants from (1) and (2)). Formally, the length of the jth dimension of the containing hypercuboid of $\mathbf{x}_k$ is:

$$l_j = \chi(wv_{k,j} + c_1(p_{k,j} - x_{k,j}) + c_2(g_{k,j} - x_{k,j})). \quad (3)$$

A particle $\mathbf{x}_k$ can therefore effectively move to any point within this hypercuboid (determined by the draws of $\mathbf{r}_1$ and $\mathbf{r}_2$), but not *outside* of it. Note that depending on the values of χ, c_1 and c_2, it is possible for one or more of $\mathbf{v}_k$, $\mathbf{p}_k$ and $\mathbf{g}_k$ to lie outside this bounded region. This restriction on a particle's movement means that local optima within this bound may be found, but any global optima outside will not be found on this iteration by $\mathbf{x}_k$, and may never be attainable. When there is a single global best for the entire swarm, then $\mathbf{g}_k$ is the same for all k.

Prior to 2002 published PSO work had only been applied to single objective problems, however, in a large number of design applications there are multiple competing quantitative measures that define the quality of a solution. For instance, in designing the ubiquitous widget, a company may wish to minimise its production cost, but also maximise/minimise one or more widget performance properties. These objectives cannot be typically met by a single solution, so, by adjusting the various design parameters, the firm may seek to discover what possible combinations of these objectives are available, given a set of constraints (for instance legal requirements and size limits of the product). The curve (for two objectives) or surface (more than two objectives) that describes the optimal trade-off possibilities between objectives is known as the Pareto front, $\mathcal{F}$. A feasible solution lying on the Pareto front cannot improve any objective without degrading at least one of the others, and, given the constraints of the model, no solutions exist beyond the Pareto front. The goal, therefore, of multi-objective algorithms (MOAs) is to locate the Pareto front of these non-dominated solutions. More formally, a multi-objective problem can be defined, without loss of generality, as:

$$\min_{\mathbf{x} \in X \subset \Re^n} f_i(\mathbf{x}) \quad \forall i = 1, \dots, m \quad (4)$$

subject to any non-negative and equality constraints:

$$\mathbf{e}(\mathbf{x}) \equiv (e_1(\mathbf{x}), \dots e_a(\mathbf{x}) \geq 0), \quad (5)$$

and

$$\mathbf{b}(\mathbf{x}) \equiv (b_1(\mathbf{x}), \dots b_d(\mathbf{x}) = 0). \quad (6)$$

If there are m different objectives, then the image of the feasible search space, X, through $\mathbf{f}()$ can be denoted by $Y \subset \Re^m$. Elements of Y are commonly referred to as objective vectors (or criteria vectors). As often the objectives being optimised are in competition, there is typically no single global optimum to multi-objective problems, rather a set of globally optimal solutions exist (potentially infinite in cardinality), referred to as the Pareto set, containing Pareto optimal solutions. A decision vector $\mathbf{x}$ is said to be Pareto optimal ($\mathbf{x} \in \mathcal{P}$) iff $\nexists \mathbf{u} \in X, \mathbf{u} \prec \mathbf{x}$, where the $\prec$ (dominance) relationship is defined as:

$$\mathbf{u} \prec \mathbf{x} \texttt{ if } (f_i(\mathbf{u}) \leq f_i(\mathbf{x}), \forall i) \wedge (\exists i \,|\, f_i(\mathbf{u}) \leq f_i(\mathbf{x})). \quad (7)$$

Via dominance a partial order can be placed on pairs of decision vectors; either vector $\mathbf{x}$ dominates $\mathbf{u}$ (in which case we can say that $\mathbf{x}$ is *better* than $\mathbf{u}$), $\mathbf{u}$ dominates $\mathbf{x}$ ($\mathbf{u}$ is better), or neither dominate each other (they are mutually non-dominating), in which case, without additional preference information about the objectives, we are *indifferent* between the solutions. This potential for indifference is one of the principal contributors to the differences between the various MOPSO approaches in the literature, as there are many different reasonable ways one may chose a personal best and a neighbourhood guide from a set. The PSO heuristic puts a number of constraints on MOPSO that MOEAs in general are not subject to. In PSO the swarm population is fixed in size, and its members cannot be replaced, only adjusted by their $\mathbf{v}$, $\mathbf{p}$ and $\mathbf{g}$, which are themselves easy to define. However, in order to facilitate an MO approach to PSO a set of non-dominated solutions (the best individuals found so far using the search process) must replace the single global best individual in the standard uni-objective PSO case. In addition, there may be no single previous best individual for each member of the swarm. Interestingly the conceptual barrier of global and local neighbourhood guides tends to get blurred in the MO application of PSO. A local individual may be selected for each swarm member, however these guides may all also be globally non-dominated (representing local areas of the estimated Pareto front maintained by the swarm), making them all also global bests. Choosing which guides to direct a swarm member's flight therefore is not trivial in MOPSO.

3. MANY-OBJECTIVE PSO

As the number of objectives increases, so does the relative proportion of objective space which is *mutually non-dominating* with a solution $(1 - \frac{1}{2^{m-1}})$. Because of this, Pareto quality measures on solutions rapidly lose their discriminating capabilities as m increases, as the probability that any other point in space is *incomparable* with another point fast approaches 1. Additionally, as the overwhelming likelihood is that *any* solution evaluated when m is large is mutually non-dominating with the set of best solutions found so far, any archive of 'best' solutions stored rapidly reaches capacity (if limited by size), or grows at such a rate as to impede algorithm convergence (by spreading out solutions in a region which is not close to $\mathcal{F}$).

In light of this a number of other quality measures have recently been devised in the literature, which aim to provide a degree of differentiation between solutions which would be viewed as otherwise equivalent when using a Pareto comparison. We now briefly outline those we will examine empirically within many-objective PSO.

3.1 Favour

The Favour Relation (FR) [9] uses partial dominance to build a directed 'relation' graph used to rank solutions.

$$fav(\mathbf{u},\mathbf{v}) =$$

$$\begin{cases} \mathbf{u} \prec_{fav} \mathbf{v} & \text{if } |\{i \mid f_i(\mathbf{u}) < f_i(\mathbf{v})\}| > |\{i \mid f_i(\mathbf{v}) < f_i(\mathbf{u})\}| \\ \mathbf{v} \prec_{fav} \mathbf{u} & \text{if } |\{i \mid f_i(\mathbf{u}) < f_i(\mathbf{v})\}| < |\{i \mid f_i(\mathbf{v}) < f_i(\mathbf{u})\}| \\ \mathbf{u} \equiv_{fav} \mathbf{v} & \text{otherwise} \end{cases} \tag{8}$$

As can be seen from (8), if $\mathbf{u} \prec \mathbf{v}$ then $\mathbf{u} \prec_{fav} \mathbf{v}$, however if $\mathbf{u}$ and $\mathbf{v}$ are mutually non-dominating (incomparable under Pareto comparison) $\mathbf{u}$ will still dominate under the favour relation if is it better on more objectives. Cycles in the graph constructed using this relation are collapsed, and an order is placed on the resultant acyclic directed graph, which is used to rank solutions.

3.2 k-optimality

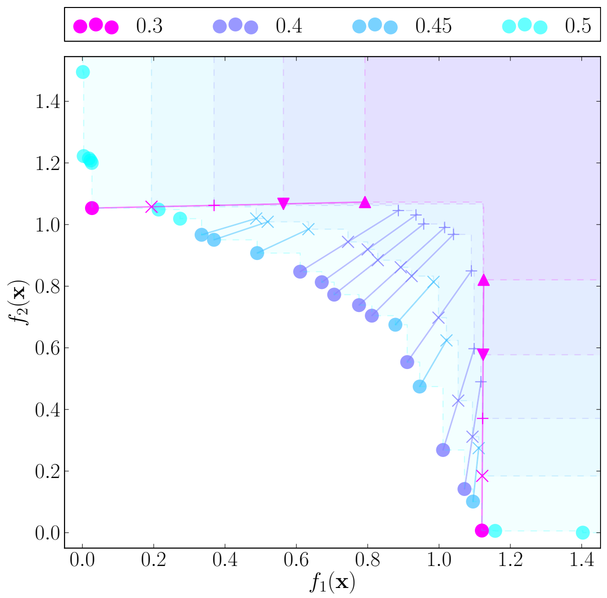
k-optimality (KO) proposed in [8] ranks solutions by dominance on subsets of objects.

$$rank_{\mathbf{u}}^{KO} = \max\left(\left\{k | \mathbf{u} \in F^{\mathbb{O}}, \forall \mathbb{O}, |\mathbb{O}| = m - k\right\}\right) \tag{9}$$

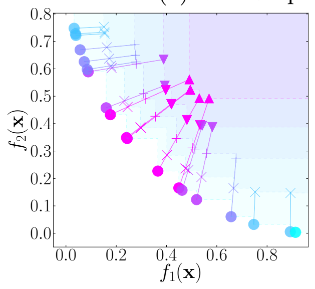
where $\mathbb{O} \subset \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_i(\mathbf{x}), \ldots, f_m(\mathbf{x})\}$ is a subset of objective functions and $F^{\mathbb{O}}$ is the non-dominated set defined given the set of objective functions defined by $\mathbb{O}$.

3.3 Controlling Dominance Area of Solutions

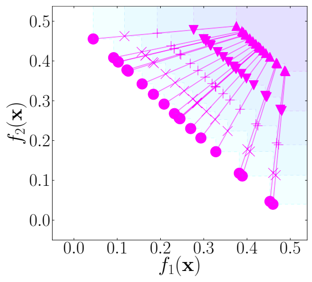
Control of Dominance Area of Solutions (CDAS) is a parametrised measure, $f'_i(\mathbf{s},\mathbf{x})$, which remaps the objective space



(a) CDAS upon a convex front.



(b) CDAS with concave surface.



(c) CDAS with a simplex.

Figure 1: CDAS-R ranked front, and the CDAS mapped fronts. The mapped sets of points '$\circ$', '$\times$', '$+$', '$\bigtriangledown$', '$\bigtriangleup$', illustrate the set of non-dominated projected points from the original set for $\mathbf{s} = 0.5$ to 0.3. Note for the non-linear front how the set membership shrinks with the reduction in s. The original non-dominated set (circles) are coloured according to their imputed CDAS rank.

to either increase or reduce selection strength [20], by setting the value of s_i used in (10) either smaller or larger than 0.5 respectively (when minimising objectives).

$$f'_i(\mathbf{s},\mathbf{x}) = \frac{||\mathbf{f}(\mathbf{x})|| \cdot \sin\left(\arccos\left(\frac{f_i(\mathbf{x})}{||\mathbf{f}(\mathbf{x})||}\right) - s_i\pi\right)}{\sin(s_i\pi)} \tag{10}$$

It has recently been applied to many objective PSO [5] with promising results (with 0.3 indicated as a good value for all s_i).

The nature of the transform when $s_i < 0.5$ increases selection pressure upon the edges and centre of a convex front (see Figure 1) to varying degrees, depending upon the shape of the non-dominated front being transformed.

3.3.1 *CDAS as a ranking method, CDAS-R*

By iteratively reducing the values in $\mathbf{s}$ it is possible to apply a ranking on a mutually non-dominated set F based upon the minimum values in $\mathbf{s}$ for which $\mathbf{u} \in F$ is within the new non-dominated front achieved by this mapping ($F'_{\mathbf{s}}$).[1] To our knowledge this is the first work to investigate modifying CDAS in this fashion, so we shall now provide further details of this approach. The rank of a solution, $\mathbf{u}$ is determined as:

$$rank_{\mathbf{u}}^{CDAS-R} = \min(\{S|\mathbf{u} \in F'_{\mathbf{s}}\}). \tag{11}$$

Since $\arccos\left(\frac{f_i(\mathbf{x})}{||\mathbf{f}(\mathbf{x})||}\right)$ and $||\mathbf{f}(\mathbf{x})||$ are independent of s_i they can be precomputed once on the first accepted insertion into an archive (in the case of PSO, this could be either personal or global). If a fixed set of d transform vectors, $S = \{\mathbf{s}_j\}_{j=1}^{d}$, is considered, then $f'_i(\mathbf{s}, \mathbf{x}), \forall j$ can also be computed only when first required, and stored for future use (across guide archives and generations).

Since if $\mathbf{u} \notin F'_{\mathbf{s}_j}$ then $\mathbf{u} \notin F'_{\mathbf{s}_{j+1}}$ where $\mathbf{s}_{j+1} \leq \mathbf{s}_j$. Iterative construction of sets for a particular rank can be determined sequentially as:

$$F'_{\mathbf{s}_{j+1}} = \left\{\mathbf{u} \in F'_{\mathbf{s}_j} \mid \nexists \quad \mathbf{v} \in F'_{\mathbf{s}_j}, \mathbf{v} \prec_{CDAS}^{\mathbf{s}_{j+1}} \mathbf{u}\right\}. \tag{12}$$

With $\prec_{CDAS}^{\mathbf{s}_{j+1}}$ defined as the dominance using the mapped CDAS values for $\mathbf{s}_{j+1}$ which only need to be computed once $\forall \mathbf{v} \in \mathbb{F}'_{\mathbf{s}_j}$. Note that $F = F'_{\mathbf{0.5}}$ (as when $\mathbf{s}_j = \mathbf{0.5}$ the mapping is the same as standard dominance).

Since $\forall \mathbf{u} \in F'_{\mathbf{s}_i}$ the rank of $\mathbf{u}$ is independent of all solutions $\mathbf{v} \in F_{\mathbf{s}_j}, \forall j \leq i$ it is possible to remove the worst solutions without affecting the rank of any other solution.

As with CDAS a greater selection pressure is applied to the edges and centre of a convex front. But unlike CDAS it is still possible to maintain a good diversity of solutions. In Figure 1 the solutions are ranked by $\mathbf{s}$ value. The shells are the result of a CDAS mapping on the front with decreasing values of $\mathbf{s}$. Here we chose $S = \{\mathbf{0.45}, \mathbf{0.40}, \ldots, \mathbf{0.25}\}$.

3.4 Crowding

We implement the crowding distance (CD) as used in NSGA-II, which computes the size of the hypercube around each $\mathbf{u} \in F$ [6].

With increasing number of dimensions finding a uniform spread of points on the true Pareto becomes very difficult. CD attempts to maximise the even spread of solutions on the front (and has recently been used in many-objective archive maintenance, e.g. [5]).

3.5 Average Ranking and Sum of Ratios

Average Ranking (AR) and Sum of Ratios (SR) map the fitness values $f_i(\mathbf{x})$ to a single value which is used directly to rank solutions. AR and SR are equivalent to Average Weighted Ranking and Sum of Weighted Ratios [1] with a weight of 1 for all objectives.

$$rank_{\mathbf{u}}^{AR} = \sum^{\forall i} \mathbb{S}_{\mathbf{u}}^{i} \tag{13}$$

$$rank_{\mathbf{u}}^{SR} = \sum^{\forall i} \frac{f_i(\mathbf{u}) - \min(\mathbb{S}^i)}{\max(\mathbb{S}^i) - \min(\mathbb{S}^i)} \tag{14}$$

Where $\mathbb{S}^i$ is the sorted set of objective values for the i^{th} objective and $\mathbb{S}_{\mathbf{u}}^{i}$ is the location of the objective vector corresponding to $\mathbf{u}$ in the sorted set.

4. EXPERIMENTAL DESIGN

We undertake two main sets of experiments in this study, to examine the impact of using different many-objective quality measures *during* the optimisation process of a standard MOPSO algorithm. We compare across a wider range of measures than considered in other recent work in the field [5], and across a wider range of test problems.

In the first set of experiments we look at the effect of using one of the quality measures as a *selector*, and keep the design of the optimisers consistent apart from this variation, allowing us to isolate its effect.

The MOPSO algorithm is as described in (1) and (2). A set of non-dominated solutions is maintained for the swarm as the source of global bests, and also each particle has a set of non-dominated solutions which they maintain and provide their personal bests. These sets are bounded at 100 elements, and if this limit is breached, random removal is performed until 100 elements is reached. This simple MOPSO is then run 30 times with seven different selection protocols for determining the particular global best and personal best to be chosen for each particle in each generation, across the DTLZ1-4 test functions with recommended parametrisation [7], for objectives = $\{5, 10, 15, 20\}$. The protocols were: FR, KO, CDAS-R, CD, AR, SR, and the baseline of random selection from the non-dominated sets (i.e., Pareto-based selection), which we denote by RR. As the first six protocols rank the non-dominated sets maintained, these rankings were used to select guides based on tournament selection of size 5 (as used in [4], where a subset of the many-objective operators considered here were compared in terms of their ability to *discriminate* between non-dominated solutions, and empirically within a GA). Element-wise truncation is used to manage boundary conditions, initial velocities are set at $\mathbf{0}$ and initial particle locations are distributed uniformly at random within the feasible search space. The PSO parameters are as described in Sec. 2, with $w = 0.5$. The number of swarm members is set at 100, and the optimisers ware run for 500 generations.

In the second set of experiments we look at the effect of using each of the quality measures for *archive maintenance*, and keep the design of the optimisers consistent apart from this variation, allowing us to isolate its effect.

In this set of experiments, the MOPSO is as described as above, however the selection is performed at random from the non-dominated guide sets maintained. *Entry* into the guide sets however is now determined by one of eight protocols: FR, KO, CDAS-R, CD, AR, SR, CDAS$^{\mathbf{0.3}}$, and the baseline of random truncation of a non-dominated set. For the first six measures when the non-dominated sets breach the capacity limits, their contents are ranked by the quality measure, and the top 100 ranked solutions are kept (the others are iteratively discarded[2]). For CDAS$^{\mathbf{0.3}}$ the archive only contains those solutions which are non-dominated under the CDAS transformation when $\mathbf{s} = \mathbf{0.3}$, if the set exceeds 100 elements, then it is truncated by random removal.

[1]It is simplest to have all elements of $\mathbf{s}$ set to the same value – in situations where the objectives are known to live on different ranges the solutions can be normalised by the observed range in the stored solutions before projecting to the new locations using (10). This approach makes considering the order on potential $\mathbf{s}$ easier to consider.

[2]Solutions must often be discarded one-by-one as for most of the quality measures described their value is affected by the set membership, and thus must be recomputed each time.

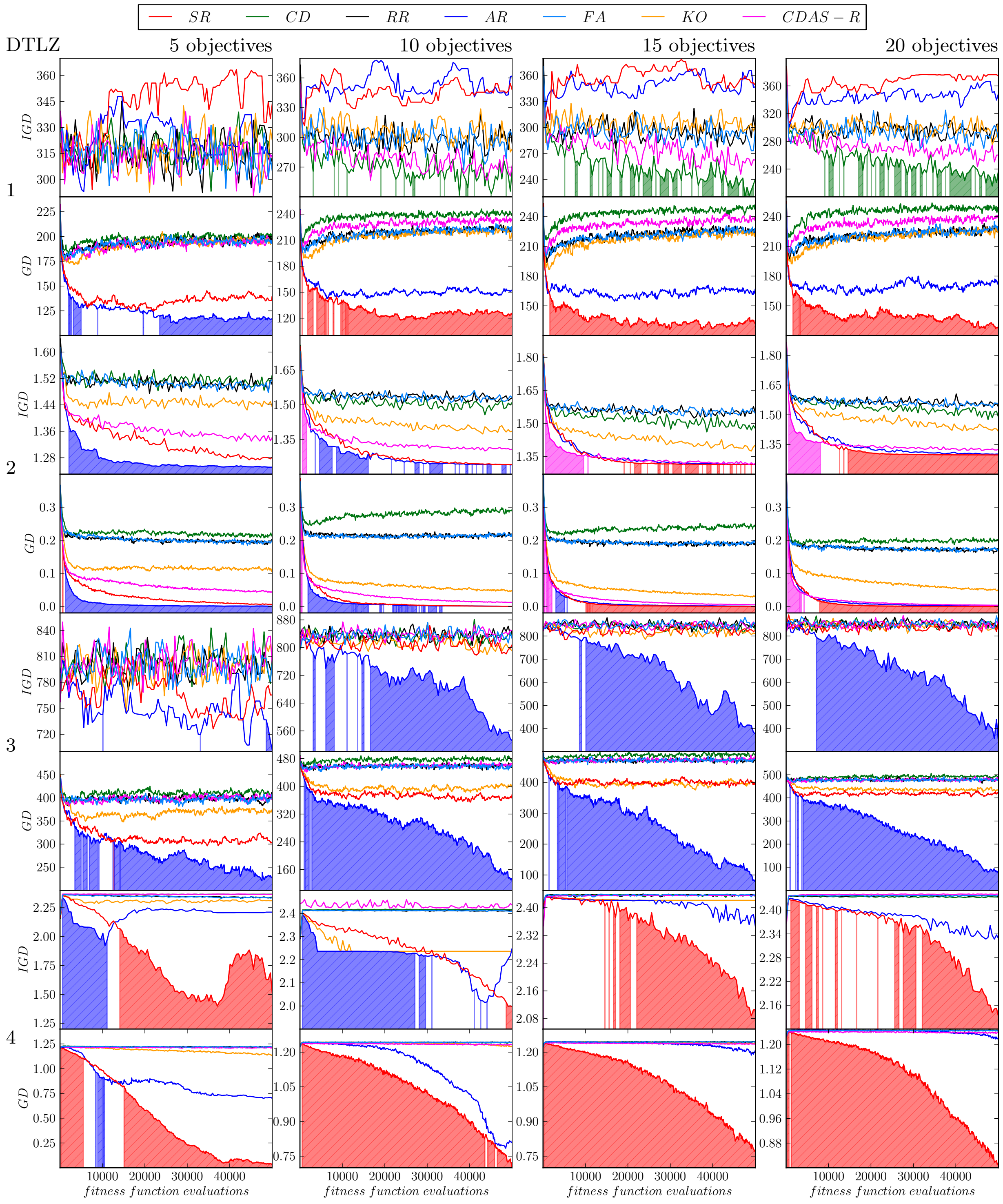


Figure 2: Selection results. Median IGD and GD values over 30 runs plotted for each method. The shaded area underneath a method indicates that it is significantly better over the range of the shaded region compared to *all* other methods (using pairwise comparisons using the non-parametric Mann-Whitney U test, at the 5% level).

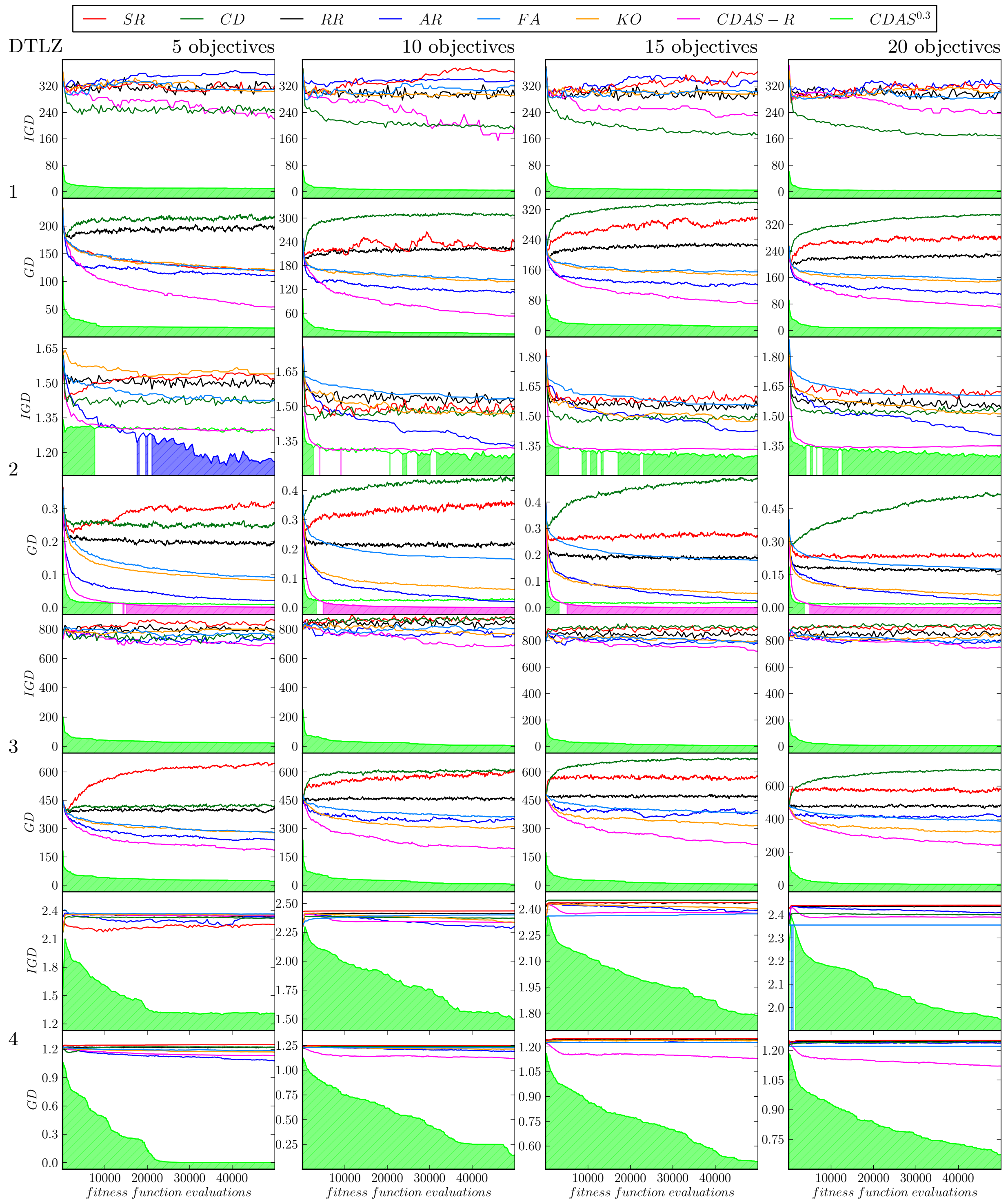


Figure 3: Archive maintenance results. Median IGD and GD values over 30 runs plotted for each method. The shaded area underneath a method indicates that it is significantly better over the range of the shaded region compared to *all* other methods (using pairwise comparisons using the non-parametric Mann-Whitney U test, at the 5% level). Note the rapid convergence of the CDAS$^{0.3}$ method, followed by flattening out, which is due to the mapping of the objective space focusing on key areas of the front, but also preventing solutions close to $\mathcal{F}$ in its central region being accepted – which puts a limit on the best IGD attainable.

Optimiser performance is tracked using the widely used generational distance (GD) and inverse generational distance (IGD) measures [3], which quantify the convergence to the Pareto front, and the spread and convergence to the Pareto front respectively. For the calculation of GD and IGD, 10000 samples from the relevant $\mathcal{F}$ were generated uniformly.

5. ANALYSIS OF RESULTS

Figures 2 and 3 provide the results of the two sets of experiments. An algorithm which is performing significantly better (as assessed by pairwise comparisons with all the other methods) is highlighted via a shaded area between its median performance line and the abscissa across the range of generations for which this is the case. Plots are arranged in both figures such that the number of objectives increases from left to right, and the DTLZ function increases numerically from top to bottom (with the IGD and GD plots being adjacent vertically).

From Figure 2 we can see that although the crowding distance tends to be lower than the other methods on IGD on DTLZ1, it is not often significantly so. From examining the GD plots, we can see that the improvement in the coverage of CD, is actually occurring simultaneously with a divergence on the GD, with SR finding significantly better converged solutions for 10+ objectives (albeit at a cost to *its* IGD). For DTLZ1 we therefore see there is a clear trade-off for these methods with respect to convergence and coverage for the numbers of objectives tested.

Even though the median across 30 runs is being plotted, the IGD is seen to oscillate considerably (especially for DTLZ1). As this is not seen to such an extent in Figure 3, it is most likely due to the achieve truncation approach used in the first set of experiments, which is stochastic in nature. It may therefore remove an archive element at random which is far from other archive members, causing a large variation in IGD.

The results tend to trend together on the two quality measures on DTLZ2, with AR, CDAS-R and SR all competitive on both IGD and GD, although CDAS-R tends to perform better in the early stages.

For DTLZ3, AR is seen to perform significantly better across the quality measures and objectives. For DTLZ4, SR generally out-performs the others, although in the earlier stages for 5 and 10 objectives, AR is better. We postulate that this is because as the objective number increases the distribution of points on the edge of the front also increases. Since AR only takes into account the ordering and not the geometric location of solutions it will then give greater and greater preference towards the edge in comparison with SR (as it prefers solutions that are in the centre of its ordering, rather than the geometric centre [12]).

In terms of Figure 3 and the maintenance approaches, the analysis is much simpler. CDAS$^{0.3}$ is seen to perform significantly and substantially better than all seven other approaches across DTLZ1, 3 and 4, and to rapidly converge for all problems bar DTLZ4, where the convergence tends to take longer. For DTLZ2, although the IGD value is seen to be significantly better for 10, 15 and 20 objectives, it loses out to CDAS-R on the GD measure. We note that there appears to be a limiting value apparent for CDAS$^{0.3}$ on across all of the problems – e.g., on DTLZ2 one can see that the IGD floors at about 1.35 across *all* objective cardinalities. The reason for this, is due to the mapping, as

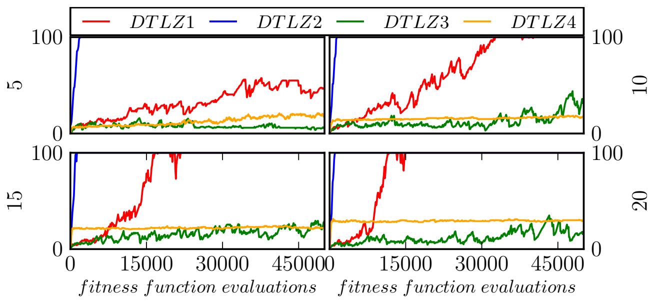


Figure 4: Archive size for CDAS$^{0.3}$ on different objectives over time.

shown in Figure 1a, where when solutions are found in the extremities of a convex $\mathcal{F}$, those in the centre are not non-dominated in the CDAS projection and therefore not stored when the values in $\mathbf{s}$ are small. As such, there is a limit on the IGD values that can be obtained (although not the GD). For DTLZ1, although the deceptive fronts are linear, CDAS still is seen to converge, as once one solution is discovered on a lower front than currently stored, a much larger number of the previous front will be discarded even if not Pareto dominated, due to the mapping of CDAS. This is supported by Figure 4, which shows the median global guide archive size as the MOPSO optimiser proceeds when using CDAS$^{0.3}$ archive maintenance. For DTLZ1, the archive size does not reach 100 at all with 5-objectives, and still takes quite a few generations to reach when in higher dimensions. Figure 4 also indicates another of the drivers of the CDAS$^{0.3}$ convergence, as we can see that for all bar DTLZ2, the archive is not truncated until later in the run, or not at all – therefore there is a persistent convergence pressure when using CDAS$^{0.3}$ from both the global selection and the personal guide selection (that is, solutions will persist from one generation to the next unless they are dominated under CDAS mapping, rather than removal due to storage constraints). For the other archiving maintenance methods the maximum size is typically reached within three generations (i.e. 400 function evaluations). CDAS-R does not do as well as CDAS$^{0.3}$ for archive maintenance, however like the other methods CDAS-R puts a rank on the non-dominated solutions found, and limits the guide archives to the 100 best of these, so there is not the same degree of convergence pressure as with CDAS$^{0.3}$ (i.e. the archive fills rapidly and requires truncation).

6. DISCUSSION

The general results with CDAS are seen to be in keeping with other recent work in the area [5], which use a CDAS-based MOPSO on DTLZ2 and DTLZ4, and compare it to a hybrid MOPSO combining AR and CD. We find here that CDAS as used for guide maintenance provides significantly better results across both the test functions, and the ranges of objective numbers we have assessed. Its only apparent weakness is manifested in DTLZ2. However, as we have discussed above, the reason for this is due to transformation of the objective space. CDAS with low values in $\mathbf{s}$ promotes convergence to the extremities of the Pareto front, but once the solutions stored are in the vicinity of $\mathcal{F}$, the centre of $\mathcal{F}$ is always projected to dominated locations in the new mapping when the front is convex (which puts a floor on the IGD achievable). With respect to the GD – we see that CDAS-R actually achieves slightly lower values than CDAS$^{0.3}$, and we conject that this may be due to it storing a greater range of guides – and therefore when these are in a reasonably converged position, the chances of getting even closer solutions anywhere across the front is improved (whereas CDAS$^{0.3}$ will only accept those on the extremity).

In relation to selection approaches, SR tends to perform the best, however we note that AR performs dramatically better on DTLZ3 for 10+ objectives (although interestingly not on DTLZ1, which has a similar deceptive front structure, but whose front *shape* is different). Corne & Knowles [4] concluded from their experiments that AR was better than SR (using a GA on a combinatorial problem), however the results here indicate a less-clear cut ordering, and a dependance on problem type, front shape and objective number – with AR tending to do better on fewer objectives.

The results indicate that archive maintenance has a lesser effect on the final quality of solutions returned, in terms of IGD and GD, compared with the effect of the selector choice *apart* from when using CDAS for archive maintenance, where the IGD and GD values are significantly lower than all selector results (barring DTLZ2).

Based upon the results here, we recommend those developing applying many-objective particle swarm optimisers strongly consider used CDAS-based archiving approaches.

7. REFERENCES

[1] P. Bentley and J. Wakefield. Finding acceptable solutions in the Pareto-optimal range using multiobjective genetic algorithms. *Soft Computing in Engineering Design and Manufacturing*, 5:231–240, 1997.

[2] C. Coello Coello and M. Lechunga. A Proposal for Multiple Objective Particle Swarm Optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation*, pages 1051–1056, 2002.

[3] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007.

[4] D. Corne and J. Knowles. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 773–780. ACM, 2007.

[5] A. B. de Carvalho and A. Pozo. Using Different Many-Objective Techniques in Particle Swarm Optimization for Many Objective Problems: An Empirical Study. *International Journal of Computer Information Systems and Industrial Management Applications*, 3:96–107, 2011.

[6] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *Lecture notes in computer science 1917*, pages 849–858. Springer, 2000.

[7] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Multi-Objective Optimization Test Problems. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 825–830, Piscataway, New Jersey, May 2002. IEEE Service Center.

[8] F. Di Pierro, S.-T. Khu, and D. Savic. An Investigation on Preference Order Ranking Scheme for Multiobjective Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 11(1):17 –45, 2007.

[9] N. Drechsler, R. Drechsler, and B. Becker. Multi-objective optimisation based on relation favour. In *Evolutionary Multi-Criterion Optimization*, pages 154–166. Springer, 2001.

[10] J. Fieldsend. Multi-Objective Particle Swarm Optimisation Methods. Technical Report 419, Department of Computer Science, University of Exeter, March 2004.

[11] J. Fieldsend and S.Singh. A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence. In *2002 UK Workshop on Computational Intelligence (UKCI'02)*, pages 37–44, Birmingham, UK, 2002.

[12] M. Garza-Fabre, G. Toscano-Pulido, and C. Coello. Two novel approaches for many-objective optimization. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1 –8, july 2010.

[13] X. Hu and R. Eberhart. Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation*, pages 1677–1681, 2002.

[14] J. Kennedy and R. Eberhart. Particle Swarm Optimization. In *IEEE International Conference on Neural Networks*, pages 1942–1948, Perth, Australia, 1995. IEEE Service Center.

[15] J. Moore and R. Chapman. Application Of Particle Swarm To Multiobjective Optimization, 1999. Unpublished, Auburn University.

[16] N. Padhye. Comparison of archiving methods in multi-objective particle swarm optimization (MOPSO): empirical study. In *GECCO'09 Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. ACM, 2009.

[17] N. Padhye, J. Branke, and S. Mostaghim. Empirical comparison of MOPSO methods: guide selection and diversity preservation. In *GECCO'09 Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. ACM, 2009.

[18] K. Parsopoulos and M. Vrahatis. Particle Swarm Optimization Method in Multiobjective Problems. In *Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002)*, pages 603–607, 2002.

[19] M. Reyes-sierra and C. A. C. 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.

[20] H. Sato, H. Aguirre, and K. Tanaka. Controlling Dominance Area of Solutions in Multiobjective Evolutionary Algorithms and Performance Analysis on Multiobjective 0/1 Knapsack Problems. *IPSJ Digital Courier*, 3:703–718, 2007.

[21] U. Wickramasinghe and X. Li. Using a Distance Metric to Guide PSO Algorithms for Many-Objective Optimization. In *GECCO'09 Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. ACM, 2009.