Hybrid Particle Guide Selection Methods in Multi-Objective Particle Swarm Optimization

Hybrid Particle Guide Selection Methods in Multi-Objective Particle Swarm Optimization

David Ireland [1], Andrew Lewis [2], Sanaz Mostaghim [3] and Jun Wei Lu [4]

[1,2] *Institute of Integrated and Intelligent Systems*
Griffith University, Brisbane, Australia
{D.Ireland, A.Lewis}@griffith.edu.au

[3] *Institute of Applied Informatics and Formal Description Methods*
University of Karlsruhe, Karlsruhe, Germany
smo@aifb.uni-karlsruhe.de

[4] *School of Engineering*
Griffith University, Brisbane, Australia
J.Lu@griffith.edu.au

Abstract

This paper presents quantitative comparison of the performance of different methods for selecting the guide particle for multi-objective particle swarm optimization (MOPSO). Two principal methods are compared: the recently described Sigma method, and a new "Centroid" method. Drawing on the different dominant behaviors exhibited by the different selection methods, a variety of hybridizations of these is proposed to develop a more robust optimization algorithm. Statistical analysis of the hybrid methods demonstrates their contribution to improved performance of the optimization algorithm.

1. Introduction

Biologically-inspired optimization algorithms have recently attracted wide interest, both for their ability to provide solutions for a diverse set of complex optimization problems without needing extensive *a priori* knowledge of the problem, and their ready adaptation to parallel and distributed computing resources [1][2][3]. A relatively new class of algorithms is Particle Swarm Optimization (PSO), introduced by Eberhart and Kennedy in 1995 [4]. This technique is based upon the social interaction of members of a population or swarm, usually referred to as particles. Applied to optimization, the swarm of particles travels through the problem parameter space, each particle being given a computed velocity based upon its own performance and the performance of another selected (guide) particle. In most cases the guide particle is chosen to be the current best performing particle in the swarm.

These methods have been widely applied to problems involving a single objective [2][5][6]. At present active research is mainly being conducted on the organizational topology of the swarm [7]. However, recent efforts have been made in developing PSO algorithms for multi-objective optimization problems requiring the simultaneous solution of a multiple set of objectives [8][9]. In the multi-objective PSO (MOPSO), choosing the guide particle has become more complex, as the output of a MOPSO algorithm consists of a set of solutions defining the trade-off between the different objectives.

For example, a simplistic approach might be to randomly select the guide particle. While this may increase the diversity of the solution set it may also decrease the rate of convergence by destroying the "memory" of the system during its evolution. For this reason this paper examines several different methods in choosing guide particles. Different methods are then combined in order to promote convergence, diversity and efficiency. It is believed no selection scheme will give adequate performance on all test problems, however the motivation behind this work is to investigate the possibility of a MOPSO with a hybrid selection scheme that will get an average performance

Proceedings of the Second IEEE International
Conference on e-Science and Grid Computing (e-Science'06)
0-7695-2734-5/06 $20.00 © 2006 IEEE

on a wide range of test functions. This paper is organized as follows: the remainder of this section briefly introduces multi-objective problems. In Section 2 the Particle Swarm Optimization (PSO) algorithm and multi-objective PSO variation are introduced. In Section 3, a proposal is made for selecting the guide particle, referred to as the Centroid method. Sections 4 and 5 are dedicated to experiments and statistical comparison of different selection methods, Section 6 their discussion, and Section 7 contains concluding remarks.

1.1 Multi-Objective Optimization

A multi-objective optimization problem with *m* objectives maybe defined by an equation of the form,

$$\text{minimize } \vec{F}=\left\{f_1(\vec{x}), f_2(\vec{x}), ..., f_m(\vec{x})\right\} \quad (1)$$

Where the decision vector $\vec{x}=\left\{x_1, x_2, ..., x_n\right\}$ belongs to a feasible region defined by $\vec{x} \in R^n$. R^n may be bounded, depending on the constraints of the problem.

In multi-objective optimization, to determine whether one decision vector is more attractive than another, a domination relation is used. For decision vectors $\vec{x}_1$ and $\vec{x}_2$, when the following conditions are met:

- $\vec{x}_1$ is at least as good as $\vec{x}_2$ for all the objectives, and
- $\vec{x}_1$ is strictly better then $\vec{x}_2$ for at least one objective

then $\vec{x}_1$ is said to "dominate" $\vec{x}_2$ (denoted $\vec{x}_1 \prec \vec{x}_2$).

In the case where $\vec{x}_1$ and $\vec{x}_2$ dominate others but not each other they are deemed mutually optimal solutions and referred to as *Pareto-optimal*. The output of a multi-objective optimization is a set of Pareto-optimal solutions which reflect the trade-off surfaces between the different objectives. This set of Pareto-optimal solutions is referred to as the *Pareto-front*. The shape of the *Pareto-front* is dependent on the particular problem.

2. Particle Swarm Optimization

The formulation of a particle swarm algorithm begins with the definition of an *N* sized population, or swarm, of decision vectors, denoted P^t where *t* represents the generation. Each *i–th* particle in the swarm has a position and velocity defined in parameter space at time *t* as $\vec{x}_i^t=\left\{x_1, x_2, ..., x_n\right\}$ and $\vec{v}_i^t=\left\{v_1, v_2, ..., v_n\right\}$ respectively. After each generation these vectors are routinely updated using:

$$\vec{v}_i^{t+1} = w\vec{v}_i^t + c_1R_1\left(\vec{p}_i^t - \vec{x}_i^t\right) + c_2R_2\left(p_g^t - \vec{x}_i^t\right)$$
$$\vec{x}_i^{t+1} = \vec{x}_i^t + \vec{v}_i^i \quad (2)$$

Position $\vec{p}_i^t$ is defined as the best position found by the *i*-th particle so far. This vector is updated after each generation. $\vec{p}_g^t$ is the global best vector of all the particles in the swarm. Both these positions may contribute in determining the direction and velocity of the particular particle. The c_1 and c_2 terms are positive constants referred to as the social and cognitive components respectively. Typically these values are set to 1.5 to allow the particle to *overshoot* either the $\vec{p}_i^t$ or $\vec{p}_g^t$ particles in parameter space and allow effective exploring around these areas. The w term is referred to as the inertial weight which limits the effect of updates on the velocity vector, effectively controlling the trade off between global and local exploration of the particles. R_1 and R_2 are randomly generated values in the range of $[0,1]$ used to add a stochastic element when updating the velocity.

2.1 Multi-Objective PSO

The single objective PSO algorithm can be transformed to include multiple objectives with little difficulty. Selecting a suitable guide particle $\vec{p}_g^t$ however becomes a more difficult task. For single-objective optimization this particle can simply be the best globally performing particle in the swarm. However, in multi-objective optimization problems a set of Pareto-optimal particles from which the algorithm must choose, or use to construct an appropriate global guide. In MOPSO the guide for the *i-th* member in the swarm maybe different to that for other members of the swarm. Thus it will be denoted $\vec{p}_{i,g}^t$ in this article. Another consideration is the method of maintaining and keeping the Pareto-optimal solutions. Typically these solutions are stored in an archive-like database which is updated after each generation in order to maintain a pure set of non-dominated solutions. The algorithm must then also maintain the archive by adding new non-dominated solutions and removing solutions which themselves become dominated.

Proceedings of the Second IEEE International
Conference on e-Science and Grid Computing (e-Science'06)
0-7695-2734-5/06 $20.00 © 2006 IEEE

3. Guide Particle Selection Methods

This section will first present the Sigma and Centroid methods for selecting the guide particle and then discuss means of developing hybrid methods to capture desirable behaviors of each method.

3.1 Sigma Method

The Sigma method involves choosing the guide particle based on the similarity of the angular position in objective space of an archive member and a particle in the swarm. This method was proposed by Mostaghim and Teich [10] and was shown to promote convergence and diversity. However, it was noted by the authors that a larger swarm size was necessary in order for the Sigma method to facilitate convergence.

The Sigma method chooses the guide particle by assigning a σ value to the particles in the archive and swarm. For each particle σ is defined as the gradient of the line connecting the location of the particle and the origin of the objective space, i.e. when two particles lie on the line $f_2 = af_1$ (considering 2 objectives) their σ values are equal. For each particle in the swarm the algorithm selects the guide particle from the archive by finding the archive member with the closest σ compared to σ_i of the swarm particle. This methodology allows the particles to move directly towards the Pareto-optimal front [10] by selecting the guide particle that orientates the particular particles to the origin in objective space. For 2-dimensional objective space σ_i is defined as,

$$\sigma = \frac{f_1^2 - f_2^2}{f_1^2 + f_2^2} \tag{6}$$

While for 3 dimensional objective space σ is a vector defined as,

$$\vec{\sigma} = \begin{pmatrix} f_1^2 - f_2^2 \\ f_2^2 - f_3^2 \\ f_3^2 - f_1^2 \end{pmatrix} / \left(f_1^2 + f_2^2 + f_3^2 \right) \tag{7}$$

The algorithm of the Sigma selection method is given in Figure 1. The function *sigma* calculates the σ value for the *i*-th swarm member and the members of a Pareto-optimal archive set denoted $\vec{A}$ which has a length of $|A|$. The algorithm then calculates the Euclidian distance between the two σ values of the swarm member and each of the archive members using the *calcdist* functions. The archive member with the smallest distance in terms of the σ value is the chosen guide particle.

BEGIN

Input: $\vec{x}_i, \vec{A}$

Output: $\vec{p}_{i,g}^{\,t}$

$\sigma_i^{Swarm} = sigma\left(\vec{F}\left(\vec{x}_i\right)\right)$

For j = 1 to $|A|$

$\sigma_j^{Archive} = sigma\left(\vec{F}\left(\vec{A}_j\right)\right)$

End

$dist = calcdist\left(\sigma_i^{Swarm}, \sigma_1^{Archive}\right)$

For j = 2 to $|A|$

$tempdist = calcdist\left(\sigma_i^{Swarm}, \sigma_j^{Archive}\right)$

If $tempdist \le dist$

$dist = tempdist$

$g = j$

End

End

$\vec{p}_{i,g}^{\,t} = \vec{A}_g$

END

Figure 1: Sigma Selection Algorithm

3.2 Centroid Method

Instead of selecting a single guide particle from the archive the Centroid method constructs the guide based upon a distance-weighted average of the archive members. For each particle in the swarm a weight vector, $\vec{\varphi}_i$, is derived by the summation of the inverse euclidian distances between the particular particle and each member in the archive given as,

$$\vec{\varphi}_j = \sum_{j=1}^{|A|} \left\| \vec{x}_i - \vec{A}_j \right\|^{-1} \tag{3}$$

The particle guide is then derived using:

$$\vec{p}_{i,g}^{\,t} = \sum_{j=1}^{|A|} \frac{\vec{\varphi}_j \vec{A}_j}{s_i} \tag{4}$$

Where s_i is defined as,

$$s_i = \sum_{j=1}^{|A|} \vec{\varphi}_j \quad (5)$$

As equation (4) shows, each member of the archive is used to compute the guide particle where the closer the archive member to the swarm member the more influence it will have. This bears a resemblance to the *Fully Informed Particle Swarm* (FIPS) algorithm proposed by Mendes and Kennedy for single objective optimization [11]. Rather then using the best performing particle in the swarm as the $\vec{p}_g^t$ particle the FIPS algorithm uses information from all of its neighbors in computing this particle. In determining the $\vec{p}_g^t$ particle the FIPS algorithm weights each particle using a relevant parameter such as fitness or distance from the current individual. Particles which have a higher weight have more contribution in computing $\vec{p}_g^t$. Hitherto this approach has received little attention for MOPSO.

The Sigma method constructs motion vectors directed at the origin of objective space, tending to "funnel" the swarm and possibly lead to a restricted approximation to the Pareto-optimal front. In contrast the Centroid method, by using constructed guides from distance-weighted averages of the archive, has the potential to expand the included arc across all constructed motion vectors, spreading the particles across a wider approximation to the Pareto-optimal front. The implemented Centroid selection algorithm is outlined in Figure 2. In this instance the weight vector was limited to a maximum and minimum value defined as the size and reciprocal size of the archive of Pareto optimal solutions respectively.

3.3 Proposed Hybrid Methods

Initial research was conducted comparing the performance and identifying the rate of convergence and diversity of the solved Pareto-optimal fronts of the Sigma and Centroid methods. This led to the consideration of creating hybrid selection methods. Early observation indicated the Centroid method was capable of achieving good diversity in the Pareto-front but displayed slow convergence. Conversely the Sigma method showed a tendency to converge too rapidly resulting in a solved Pareto-front with very little diversity. If the swarm particles were to use different or exchangeable selection methods it was considered possible that the best properties exhibited by two selection methods might combine, resulting in a more effective and robust MOPSO algorithm

BEGIN

Input: $\vec{x}_i, \vec{A}$

Output: $\vec{p}_{i,g}^t$

$\vec{s}_i = 0$

For j = 1 to $|A|$

$\vec{\varphi}_j = calcdist\left(\vec{x}_i, \vec{A}_j\right)^{-1}$

If $\vec{\varphi}_j < |A|^{-1}$

$\vec{\varphi}_j = |A|^{-1}$

Else if $\vec{\varphi}_j > |A|$

$\vec{\varphi}_j = |A|$

End

$\vec{s}_i = \vec{s}_i + \vec{\varphi}_i$

End

$\vec{p}_{i,g}^t = 0$

For j = 1 to $|A|$

$\vec{p}_{i,g}^t = \vec{p}_{i,g}^t + \vec{A}_j \cdot \vec{\varphi}_i / \vec{s}_i$

End

END

Figure 2: Weighted Centroid Selection Algorithm

Three base selection strategies were employed, including Sigma and Centroid algorithms and a method that randomly selected the guide particle from the archive, subsequently termed the *Random* method. By merging these base methods a further three hybrid MOPSO algorithms were constructed which each contained two particle guide selection schemes.

- Hybrid I – (Centroid/Random)
- Hybrid II – (Centroid/Sigma)
- Hybrid III – (Random/Sigma)

With hybrids, selection schemes for each particle were chosen using roulette wheel selection where a particular selection scheme had an eual probability of being selected.

4. Performance Metric

4.1 HyperVolume Ratio

In order to compare the different selection methods the HyperVolume ratio performance metric was used.

Initially the total volume occupied by the solved Pareto front (SPF) and the known global Pareto-front (GPF) are calculated. Figure 3 shows an illustration of an example two objective Pareto Front and the area that it occupies.

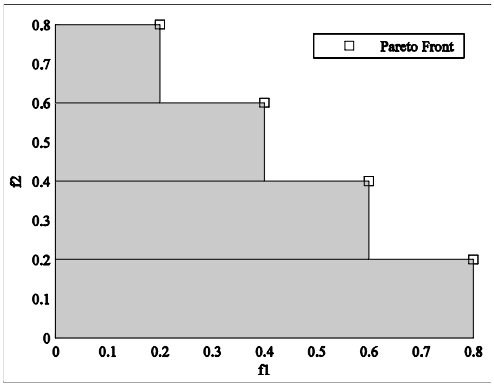

Figure 3: Shaded regions represent the area "occupied" by the particular Pareto Front.

The HyperVolume ratio (HR) is then defined as the ratio of the volumes occupied by the SPF and GPF [12]. If the HR is smaller than 1 then the volume beneath the SPF is smaller than beneath the GPF, suggesting the Pareto Front is not spread over the GPF. Conversely, if the HR is greater than 1, the SPF has a finite distance from the GPF, indicating a failure to converge. The HR metric thus gives information about the spread of the SPF and the amount of convergence in a simple way.

4.2 Test Functions

Four well known test problems [13] were implemented in the experiment, each had the form,

$$f_1(\vec{x}) = x_1 \tag{6}$$

$$f_2(\vec{x}) = g(\vec{x}) \cdot h(f_1, g) \tag{7}$$

Where f_1 and f_2 are the functions to be minimized and functions *g* and *h* are given in Table 1. These functions allow the user to design various shaped global Pareto Fronts and difficulties. The parameter space dimensions *N* for each of the test functions are also given in Table1.

Test function 1 exhibits a convex shaped global Pareto-front, test function 2 exhibits a concaved shape Pareto-front referred to as non-convex, and test function 3 represents a multi-modal problem with 21^{N-1} local Pareto-fronts. Test function 4 gives a discontinuity of the global pareto-front which is not produced in parameter space. For test functions 1, 2 and 4 $x_i \in [0,1]$ while for test function 3 $x_1 \in [0,1]$ and $x_i \in [-5,5],\ i \neq 1$.

Multi-Objective Test Functions	
1.	$g(\vec{x}) = 1 + 9\left(\sum_{n=1}^{N} x_n\right)/(N-1)$ $h(f_1, g) = 1 - \sqrt{f_1 / g}$ $N = 30$
2.	$g(\vec{x}) = 1 + 9\left(\sum_{n=1}^{N} x_n\right)/(N-1)$ $h(f_1, g) = 1 - (f_1 / g)^2$ $N = 5$
3.	$g(\vec{x}) = 1 + 10(N-1) + \sum_{n=2}^{N}\left(x_n^2 - 10\cos(4\pi x_n)\right)$ $h(f_1, g) = 1 - (f_1 / g)^2$ $N = 5$
4.	$g(\vec{x}) = 1 + 9\left(\sum_{n=1}^{N} x_n\right)/(N-1)$ $h(f_1, g) = 1 - \sqrt{f_1 / g} - \sin(10\pi f_1)$ $N = 30$

Table 1: Multi-Objective Test Functions

4.3 Parameter Settings

The MOPSO algorithms were implemented with a swarm size of 100 particles which was updated for 250 generations for all test functions except for 3, where for this test function a swarm size of 250 particles and 500 generations was used as the inherent multimodality of the test function required a larger swarm size and more iterations to get a valid approximated Pareto Front.

The size of the archive was maintained at a maximum of 20 Non-dominated particles, by using clustering methods [13]. The initial swarm was generated using a uniformly distributed random number generator. The inertial weight was initialized at 0.8 and was decreased after each generation by 0.00125.

5. Statistical Results

Each selection scheme was run 30 times in order to supply a valid sample for statistical analysis. The results are presented in a *box-and-whisker* plot and the mean, μ, and standard deviation, σ, of the solutions tabulated. The box-and-whisker plot was chosen because it readily displays key measures: the enclosed box depicts the lower quartile, median and upper quartile while the arms extending from the box (*whiskers)* show the smallest and largest observation of the statistical data.

Figure 4 and Table 2 give the experimental results for test function 1 which exhibited a convex shaped global Pareto Front. These results show the Hybrid III method which consisted of the Random and Sigma selection schemes performed the best with a μ of 1.2103 and σ of 0.047 followed by the Random and Hybrid I schemes. The Centroid method performed the worst which had the largest deviation and poorest mean.

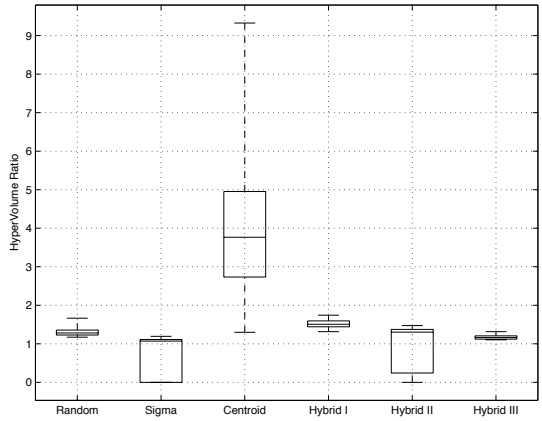

Figure 4: Box-Whisker Plot for the Convex Test Functions HyperVolume Ratio

Selection Method	μ	σ
Random	1.3527	0.1497
Sigma	0.8675	0.4491
Centroid	4.1619	1.4369
Hybrid I	1.5916	0.1558
Hybrid II	1.0273	0.5244
Hybrid III	1.2103	0.0749

Table 2: Mean and Standard Deviation of the Convex Test Functions HyperVolume Ratio

The results for test function 2 are given in Figure 5 and Table 3. For this test function the most successful selection scheme was the Sigma method which had a μ of 0.7482 and σ of 0.4199 followed by the Random and Hybrid III methods. In this instance all the selection methods gave similar deviations. The box-and-whisker plot shows at times, that all selection methods produced approximated Pareto Front which occupied very little area.

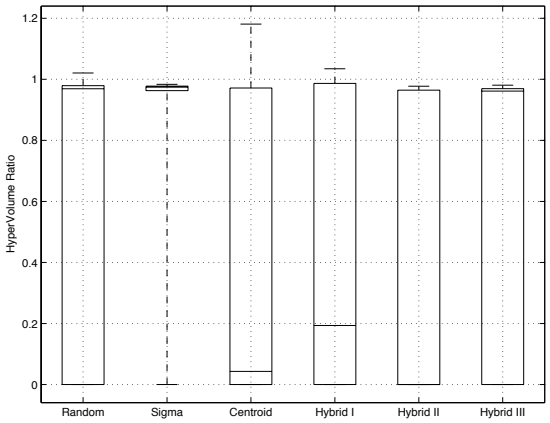

Figure 5: Box-and-Whisker Plot for the Non-Convex Test Functions HyperVolume Ratio

Selection Method	μ	σ
Random	0.6841	0.4557
Sigma	0.7482	0.4199
Centroid	0.4228	0.4847
Hybrid I	0.4737	0.4938
Hybrid II	0.4496	0.4889
Hybrid III	0.6130	0.4744

Table 3: Mean and Standard Deviation of the Non-Convex Test Functions HyperVolume Ratio

Test function 3 proved to be the most difficult function to optimize due to its multi-modal nature as seen from the low means in Figure 6 and Table 4. It would appear the Random and Centroid selection methods gave the best performance with μ values of 0.8452 and 0.4418, and σ values of 1.3487 and 0.6558 respectively. The nearest performing hybrid method was Hybrid I which also contained the Centroid and Random selection schemes. The poorest performing selection scheme was the Sigma method.

Proceedings of the Second IEEE International
Conference on e-Science and Grid Computing (e-Science'06)
0-7695-2734-5/06 $20.00 © 2006 IEEE

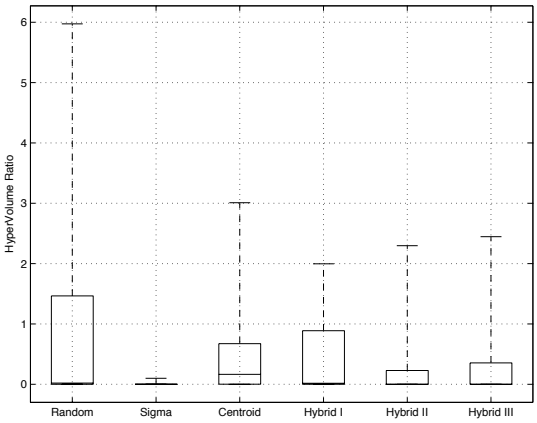

Figure 6: Box-and-Whisker Plot for the Multi-Modal Test Functions HyperVolume Ratio

The experimental test results for test function 4 are given in Figure 7 and Table 5. The most successful selection scheme was the Hybrid III method with a μ of 1.6684 and σ of 0.2511, followed by the Hybrid II and Random methods. Again the Centroid method performed the worst with a large deviation and the poorest mean.

Selection Method	μ	σ
Random	0.8452	1.3487
Sigma	0.0083	0.0233
Centroid	0.4418	0.6558
Hybrid I	0.4002	0.5707
Hybrid II	0.2677	0.5581
Hybrid III	0.2892	0.6030

Table 4: Mean and Standard Deviation of the Multimodal Test Functions HyperVolume Ratio

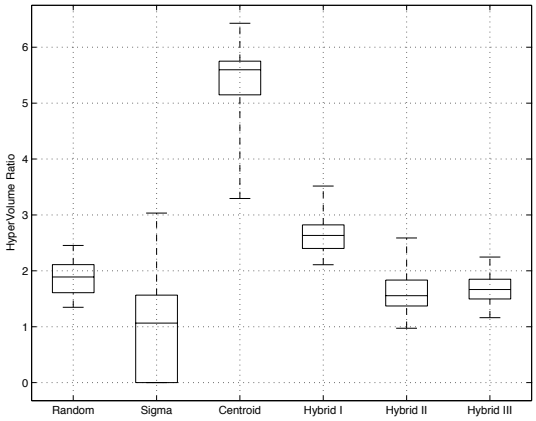

Figure 7: Box-Whisker Plot for the Discontinuous Test Functions HyperVolume Ratio

Selection Method	μ	σ
Random	1.8843	0.2863
Sigma	0.9736	0.9121
Centroid	5.433	0.6984
Hybrid I	2.6156	0.3225
Hybrid II	1.609	0.3569
Hybrid III	1.6684	0.2511

Table 5: Mean and Standard Deviation of the Discontinuous Test Functions HyperVolume Ratio

6 Discussion of Results

Although not completely conclusive, it is evident in the statistical plots of Section 5 that the roulette wheel hybridization of the particle guide selection schemes results in an algorithm that shares similar performance characteristics with the pure selection schemes. Overall it would appear the Hybrid III and Random selection schemes were the most robust for the implemented test functions. The reason behind this could be the combinatorial behavior of the Random selection scheme accelerating the swarm particle and the Sigma method conversely drawing the particle towards the origin creating an effective exploration and exploitation response.

The Centroid method appeared to consistently produce an approximated Pareto Front with a significantly higher Hyperarea ratio compared to the other methods; this occurred in test functions 1 and 4. The Hybrid I and II methods which contain the Centroid, Random and Sigma selection methods failed to produce any significant improvement or evidence of robustness. Although the performance of the Sigma method was relatively high in test function 2 it completely failed in the multi-modal test function and gave a large deviation in test function 4 which contained discontinuities in the objective space.

6. Conclusions

This paper investigated the performance of several particle guide selection schemes for MOPSO and proposed the concept of using hybrid selection methods to increase diversity and improve convergence to the solved Pareto-front for a series of test functions. The proposed Centroid method uses a selection strategy of finding the global best guides in the search space, where the other methods work in the objective space. This has great importance in MOPSO because the move strategy happens in the search space. Statistical analysis demonstrated hybrid selection

Proceedings of the Second IEEE International
Conference on e-Science and Grid Computing (e-Science'06)
0-7695-2734-5/06 $20.00 © 2006 IEEE

methods displayed significantly improved performance, but this was problem dependent. No selection scheme was dominant on all test functions, but results were sufficiently promising to warrant further investigation. In future, we wish to investigate the notion of online adjustment of the roulette wheel selection probabilities using a relative performance metric. Also, more extensive testing will be done on problems with a higher number of objectives and parameters.

7. References

[1] R.C Eberhart and Y. Shi, "*Particle Swarm Optimization: developments, applications and resources",* In Proc Congress on Evolutionary Computation Vol. 1 pp. 81 – 86 May 2001

[2] Y. Rahmat-Sammi, N. Jin and S. Xu, "*Particle Swarm Optimization (PSO) In Electromagnetics: Let the Bees Design Your Antennas*", 22nd Annual Review on Progress in Applied Electromagnetics, March 2006.

[3] J. Kennedy and R.C Eberhart, "*Swarm Intelligence",* Morgan Kaufmann 2001

[4] J. Kennedy and R.C. Eberhart, "*Particle Swarm Optimization*", In Proc. International Conference on Neural Networks Perth Australia 1995

[5] S.M. Guru, S.K. Halgamuge and S. Fernado, "*Particle Swarm Optimisers for Cluster Formation in Wireless Sensor Networks*", In Proc. 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference, pp 319 – 324 December 2005

[6] M.M Khodier, C.G Christodoulou, "*Linear Array Geometry Synthesis with Minimum Sidelobe Level and Null Control Using Particle Swarm Optimization*", IEEE Transactions on Antennas and Propagation Vol. 53 No. 8 pp 2674 – 2679 August 2005

[7] J. Kennedy and R. Mendes, "*Topological Structure and Particle Swarm Performance*", In Proc of the 4th Congress on Evolutionary Computation 2002

[8] K.E. Parsopoulos and M.N. Vrahatis, "*Particle Swarm Optimization Methods in Mutliobjective Problems",* In Proc. of the 2002 ACM Symposium on Applied Computing, pp 603—607 Madrid Spain 2002.

[9] C. A. Coello Coello and M. S. Lechuga, "*MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization",* In Proc. IEEE World Congress on Computational Intelligence, Hawaii 2002

[10] S. Mostaghim and J. Teich, "*Strategies for Finding Good Local Guides in Multi-Objective Particle Swarm Optimization (MOPSO)"*, In Proc. IEEE Conference on Swarm Intelligence Symposium pp. 26 – 33 April 2003

[11] R. Mendes and J. Kennedy and J. Neeves, "*The Fully Informed Particle Swarm: Simpler, Maybe Better*", IEEE Transactions on Evolutionary Computation, Vol. 8 No. 3 pp. 204-210 June 2004

[12] Y. Collette and P. Siarry, "*Multiobjective Optimization Principles and Case Studies",* Springer-Verlag Berlin Heidelberg 2003

[13] E. Zitzler, "*Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*", TIK-Schriftenreihe Nr. 30, Dissertation ETH No. 13398, Shaker Verlag, Germany, Swiss Federal Institute of Technology (ETH) Zurich 1999