Heterogeneous Particle Swarm Optimization

Andries P. Engelbrecht

Department of Computer Science, University of Pretoria, South Africa
engel@cs.up.ac.za

Abstract. Particles in the standard particle swarm optimization (PSO) algorithms, and most of its modifications, follow the same behaviours. That is, particles implement the same velocity and position update rules. This means that particles exhibit the same search characteristics. A heterogeneous PSO (HPSO) is proposed in this paper, where particles are allowed to follow different search behaviours selected from a behaviour pool, thereby efficiently addressing the exploration–exploitation trade-off problem. A preliminary empirical analysis is provided to show that much can be gained by using heterogeneous swarms.

1 Introduction

Particle swarm optimization (PSO) [3,8] is a stochastic, population-based optimization method. PSO algorithms maintain a *swarm* of candidate solutions, called *particles*. Each particle adjusts its position in search space by adding to its current position a step size, called the *velocity*. Step sizes are computed based on how far a particle is from the best position that the particle found during the search process, and how far the particle is from the best solution found by its neighborhood. The standard PSO and most of its modifications [4] make use of homogeneous swarms where all of the particles follow exactly the same behaviour. That is, particles implement the same velocity and position update rules. The effect is that particles have the same exploration and/or exploitation characteristics.

A very important aspect of optimization is the ability of an optimization algorithm to balance exploration and exploitation. Initially, the algorithm should focus on exploration, while preferring exploitation as the search process converges on an optimum. It is however difficult to determine at which point should the algorithm switch from an explorative behaviour to an exploitative behaviour. It may therefor be of an advantage to rather use heterogeneous swarms, where particles are allowed to implement different velocity and position update rules. By allowing particles to implement different update rules, a swarm may consist of explorative particles as well as exploitative particles. The optimization algorithm therefor has the ability to explore and exploit throughout the search process.

This paper proposes a heterogeneous PSO (HPSO), where particles in a swarm will be allocated different search behaviours by randomly selecting velocity and position update rules from a behaviour pool. The formal concept of heterogeneous swarms was introduced by Engelbrecht in [5], where the model investigated

M. Dorigo et al. (Eds.): ANTS 2010, LNCS 6234, pp. 191–202, 2010.

in this paper has been proposed. However, the idea of heterogenous swarms is not new. Examples of existing approaches where particles are allowed to implement different behaviours include, amongst others,

– The division of labor PSO [16], where particles are allowed to switch to a local search near the end of the search process.
– The life-cycle PSO [9], where particles follow a life-cycle, changing from a PSO particle, to a genetic algorithm individual, to a stochastic hill-climber. At any time, individuals may follow different behaviours.
– The predator-prey PSO [13], where the swarm contains predator and prey particles. Predator particles are attracted only to the global best position, thereby exploiting. Prey particles implement the standard PSO velocity update rule, but with an additional term added to repel prey particles from the position of predator particles.
– The guaranteed convergence PSO [15], where the global best particle follows a different, exploitative search behaviour than all the other particles.
– The NichePSO [2], developed to locate multiple solutions. A main swarm of particles is used, where particles implement a cognitive-only velocity update. Sub-swarms are formed around optima, with particles following the guaranteed convergence PSO.
– The charged PSO [1], where some particles have a charge and others not. Non-charged particles implement the standard velocity update rule, while charged particles add an additional repelling force to the velocity update rule.

Recent models that nake use of a more generic concept of heterogeneous behaviours include the heterogeneous cooperative algorithms developed by Olorunda and Engelbrecht [11], the heterogeneous PSO algorithms of Montes de Oca *et al* [10], and the adaptive heterogeneous PSO proposed by Spanevello and Montes de Oca [14]. The heterogeneous cooperative algorithm allows sub-swarms in a cooperative coevolutionary model to implement different meta-heuristic algorithms exhibiting different search behaviours. Montes de Oca *et al* considered different levels of heterogeneity, using the term update-rule heterogeneity to mean PSO algorithms where particles use different position and velocity update rules. In their work, ony two different update rules were used. Spanevello and Montes de Oca proposed that behaviours change during the optimization process.

The HPSO proposed in this paper differ from the above PSO algorithms, in that behaviours are randomly assigned from a pool of behaviours. Two strategies are proposed, one where the randomly selected behaviours remain static, and the other where behaviours change at each iteration by randomly selecting new behaviours from the behaviour pool. A preliminary empirical analysis is provided to show that the proposed HPSO has potential.

The rest of the paper is organized as follows: Section 2 discusses a few homogeneous PSO algorithms which differ only in the implemented position and velocity update rules. The HPSO is presented in Section 3, while some empirical results are provided in Section 4.

2 Homogeneous Particle Swarm Optimizers

This section provides a very compact overview of homogeneous PSO algorithms which will be used by the HPSO proposed in Section 3. PSO models are included that differ in their exploration and exploitation behaviours. Please note that this is not an extensive review of homogeneous PSO algorithms.

2.1 Traditional Position and Veclosity Updates

The traditional PSO particle velocity and position updates, assuming a star neighborhood topology, are given as follows [3,8]:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)(y_{ij}(t) - x_{ij}(t)) + c_2 r_{2j}(t)(\hat{y}_j(t) - x_{ij}(t)) \quad (1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (2)$$

where $x_{ij}(t), y_{ij}(t)$ and $\hat{y}_j(t)$ refer respectively to particle i's position, personal best position, and global best position in dimension j at time step t. The constants c_1 and c_2 are the acceleration coefficients, and $r_{1j}(t), r_{2j}(t) \sim U(0,1)$. In the above, w is the inertia weight.

Equation (1) result in particles with a balance in exploration and exploitation depending on the values of the parameters, w, c_1 and c_2. For the purposes of this paper, c_1 is initially set to a value larger than c_2. Over time, c_1 is linearly decreased, while c_2 is linearly increased [12]. This will focus on exploration during the initial search steps, moving towards more exploitation as the number of iterations increases.

2.2 Cognitive-Only Model

The cognitive-only velocity update [6] removes the social component from equation (1), to result in

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)(y_{ij}(t) - x_{ij}(t)) \quad (3)$$

The same position update as in equation (2) is used. The cognitive-only model results in more exploration, due to the fact that each particle becomes a hill-climber.

2.3 Social-Only Model

The social-only velocity update [6] removes the cognitive component from equation (1), to result in

$$v_{ij}(t+1) = wv_{ij}(t) + c_2 r_{2j}(t)(\hat{y}_j(t) - x_{ij}(t)) \quad (4)$$

The same position update as in equation (2) is used. The cognitive-only model results in faster exploitation, as the entire swarm is one stochastic hill-climber.

2.4 Barebones PSO

Kennedy [7] developed the barebones PSO, where the velocity update is replaced with

$$v_{ij}(t+1) \sim N\left(\frac{y_{ij}(t) + \hat{y}_j(t)}{2}, \sigma\right) \tag{5}$$

where $\sigma = |y_{ij}(t) - \hat{y}_j(t)|$. The position update changes to

$$x_{ij}(t+1) = v_{ij}(t+1) \tag{6}$$

Note that the velocity no longer serves as a step size, but is the actual new position of the particle, sampled from the above Gaussian distribution. The barebones PSO facilitates initial exploration, due to large deviations (initially, personal best positions will be far from the global best position). As the number of iterations increases, the deviation approaches zero, focussing on exploitation of the average of the personal best and global best positions.

2.5 Modified Barebones PSO

Kennedy [7] modified the barebones velocity equation to improve its exploration abilities. The new velocity update is

$$v_{ij}(t+1) = \begin{cases} y_{ij}(t) & \text{if } U(0,1) < 0.5 \\ N\left(\frac{y_{ij}(t)+\hat{y}_j(t)}{2}, \sigma\right) & \text{otherwise} \end{cases} \tag{7}$$

Exploration is increased during the initial stages of the search process by focusing 50% of the time on personal best positions. Recall that personal best positions will initially differ significantly due to uniform random initialization of particles. As the process converges, the focus will move to exploitation, as all personal best positions will converge towards the global best position.

3 Heterogeneous Particle Swarm Optimization

The HPSO proposed in this paper selects random behaviours for particles from a pool of behaviours. Each behaviour consists of a pair containing a position update and a velocity update. For the purposes of this paper, the pool of behaviours include the five models summarized above in Section 2. Two different HPSO models are proposed, namely

- The **static** HPSO (sHPSO), where behaviours are randomly assigned to particles during initialization. The assigned behaviours do not change during the search process.
- The **dynamic** HPSO (dHPSO), where paricle behaviours can randomly change during the search process. For the purposes of this paper, a particle randomly selects new behaviours from the behaviour pool when the particle fails to improve its personal best position over a window of recent

iterations. If the personal best position does not change, it may indicate early stagnation, which can be addressed by assigning a new search behaviour to the particle.

The only changes to the algorithm flow of the standard PSO are therefor a step to initialize the behaviours of particles, and for the dynamic HPSO, to assign new behaviours for particles that stagnate. More elaborate schemes to trigger reinitialization of behaviours can be implemented. However, as this is a preliminary study, additional mechanisms will be explored in future research.

4 Empirical Results

This section provides preliminary results to indicate that the HPSO model shows potential to be further explored. For the purposes of this paper, the two heterogeneous models were compared with the homogeneous models summarized in Section 2, in order to determine if any gain can be achieved with a heterogeneous model of these behaviours. Each algorithm is tested on the following functions:

- **Ackley**: $f(\mathbf{x}) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{j=1}^{n} x_j^2}} - e^{\frac{1}{n}\sum_{j=1}^{n}\cos(2\pi x_j)} + 20 + e$, where $x_j \in [-30, 30]$.
- **Quadric**: $f(\mathbf{x}) = \sum_{l=1}^{n}\left(\sum_{j=1}^{l} x_j\right)^2$, where $x_j \in [-100, 100]$.
- **Rastrigin**: $f(\mathbf{x}) = 10n + v\sum_{j=1}^{n}\left(x_j^2 - 10\cos(2\pi x_j)\right)$, where $x_j \in [-5.12, 5.12]$.
- **Rosenbrock**: $f(\mathbf{x}) = \sum_{j=1}^{n-1}\left(100(x_{j+1} - x_j^2)^2 + (x_j - 1)^2\right)$, where $x_j \in [-30, 30]$.
- **Salomon**: $f(\mathbf{x}) = v - \cos(2\pi\sum_{j=1}^{n} x_j^2) + 0.1\sqrt{\sum_{j=1}^{n} x_j^2} + 1$, where $x_j \in [-600, 600]$.
- **Griewank**: $f(\mathbf{x}) = 1 + \frac{1}{4000}\sum_{j=1}^{n} x_j^2 - \prod_{j=1}^{n}\cos\left(\frac{x_j}{\sqrt{j}}\right)$, where $x_j \in [-600, 600]$.

This results in two unimodal (Quadric and Rosenbrock) and four multimodal (Ackley, Rastrigin, Salomon, Griewank) functions. Note that Rosenbrock, Griewank, and Salomon are not separable.

The functions were used in 10, 30, 50, and 100 dimensions. Each algorithm was executed on each function for 30 independent runs of 1000 iterations. Swarm sizes of 50 particles were used. The inertia weight was set to $w = 0.72$ for all the velocity update rules. For the velocity update rule in equation (1), c_1 started at 2.5, linearly reduced to 0.5; c_2 started at 0.5, linearly increased to 2.5. For all other velocity updates, $c_1 = c_2 = 2.5$.

All algorithms were implemented using CIlib (http://www.cilib.net).

Tables 1 and 2 present the fitness of the best solution found at the end of the 1000 iterations, averaged over the 30 simulations, together with standard deviations. Table 3 summarizes these results by ranking the algorithms based on average best fitness. Figure 1 illustrates the performance of the algorithms over time for all functions in 50 dimensions, and figure 2 illustrates the scalability of all algortihms.

Table 1. Comparative results, showing average best solutions found for Ackley, Quadric, Rastrigin

Algorithm	Ackley	Quadric	Rastrigin
		10 Dimensions	
Standard PSO	2.25E+00±9.09E-01	2.17E+01±3.61E+01	1.67E+01±7.46E+00
Social PSO	3.27E+00±1.37E+00	6.92E+01±9.11E+01	1.85E+01±8.32E+00
Cognitive PSO	1.95E+01±4.24E-01	1.48E+04±3.56E+03	1.19E+02±9.88E+00
Barebones PSO	4.23E-15±9.01E-16	6.33E-19±1.71E-18	5.34E+00±3.14E+00
Modified Barebones	3.40E-15±0.00E+00	1.01E-07±1.90E-07	6.63E-02±2.52E-01
Static HPSO	3.99E-15±0.00E+00	1.34E-11±3.44E-11	1.47E+00±2.59E+00
Dynamic HPSO	4.44E-16±0.00E+00	2.07E-08±6.96E-08	2.02E+00±2.09E+00
		30 Dimensions	
Standard PSO	8.71E+00±9.52E-01	2.78E+03±1.20E+03	1.28E+02±2.81E+01
Social PSO	9.95E+00±1.49E+00	5.13E+03±2.32E+03	1.08E+02±2.08E+01
Cognitive PSO	2.04E+01±2.47E-01	1.28E+05±4.15E+04	4.32E+02±2.79E+01
Barebones PSO	2.05E-08±2.94E-08	8.55E+02±5.23E+02	6.02E+01±1.44E+01
Modified Barebones	4.10E-07±2.80E-07	7.10E+03±4.42E+03	1.51E+01±5.67E+00
Static HPSO	1.20E+00±7.79E-01	8.71E+00±1.66E+01	1.75E+01±2.85E+01
Dynamic HPSO	1.08E-10±1.64E-10	3.65E-01±1.03E+00	1.62E+00±6.35E+00
		50 Dimensions	
Standard PSO	1.01E+01±9.32E-01	9.89E+03±4.97E+03	2.81E+02±4.58E+01
Social PSO	1.17E+01±1.00E+00	1.34E+04±4.55E+03	2.82E+02±4.03E+01
Cognitive PSO	2.06E+01±1.12E-01	3.47E+05±1.03E+05	7.67E+02±4.12E+01
Barebones PSO	1.14E-01±3.57E-01	2.79E+04±1.05E+04	1.46E+02±2.43E+01
Modified Barebones	1.23E-02±5.18E-03	8.19E+04±3.63E+04	7.84E+01±3.77E+01
Static HPSO	2.87E+00±4.55E+00	1.29E+03±2.13E+03	4.47E+01±7.08E+01
Dynamic HPSO	4.65E-09±1.19E-08	2.47E+01±8.48E+01	6.64E-02±3.635E-01
		100 Dimensions	
Standard PSO	1.22E+01±8.25E-01	4.06E+04±1.45E+04	6.72E+02±5.14E+01
Social PSO	1.32E+01±7.83E-01	6.15E+04±2.53E+04	7.21E+02±6.61E+01
Cognitive PSO	2.08E+01±7.39E-02	1.42E+06±5.19E+05	1.62E+03±4.10E+01
Barebones PSO	1.20E+01±9.30E-01	2.70E+05±1.00E+05	1.30E+03±3.61E+02
Modified Barebones	9.38E+00±2.29E+00	5.77E+05±2.15E+05	5.81E+02±1.30E+02
Static HPSO	2.87E+00±4.55E+00	2.41E+04±3.87E+04	1.24E+02±1.93E+02
Dynamic HPSO	1.60E-07±7.13E-07	1.28E+01±2.28E+03	1.78E-12±5.69E-12

Figure 1 shows that dHPSO provided the best fitness for 50 dimensions over all of the functions. This is confirmed in Table 3, which shows an average rank of 1 for dHPSO over all functions. This observation of dHPSO's best performance is also for 100 dimensions. For 30 dimensions, dHPSO has the best rank of 1.5, due to 3 functions (Rastrigin, Rosenbrock, Griewank) where dHPSO was the second best performer (after sHPSO and the modified barebones PSO). For 10 dimensions dHPSO has the second best rank of 2.17 after the modified barebones PSO which obtained a rank of 2. The average rank over all dimensions for each function show that dHPSO has the best rank for all but one function (Rosenbrock), where sHPSO has the best rank. When considering the overall rank, as the average rank over all functions and all dimensions, the HPSO models performed best, with dHPSO having a rank of 1.42 followed by sHPSO with a

Table 2. Comparative results, showing average best solutions found for Rosenbrock, Salomon, Griewank

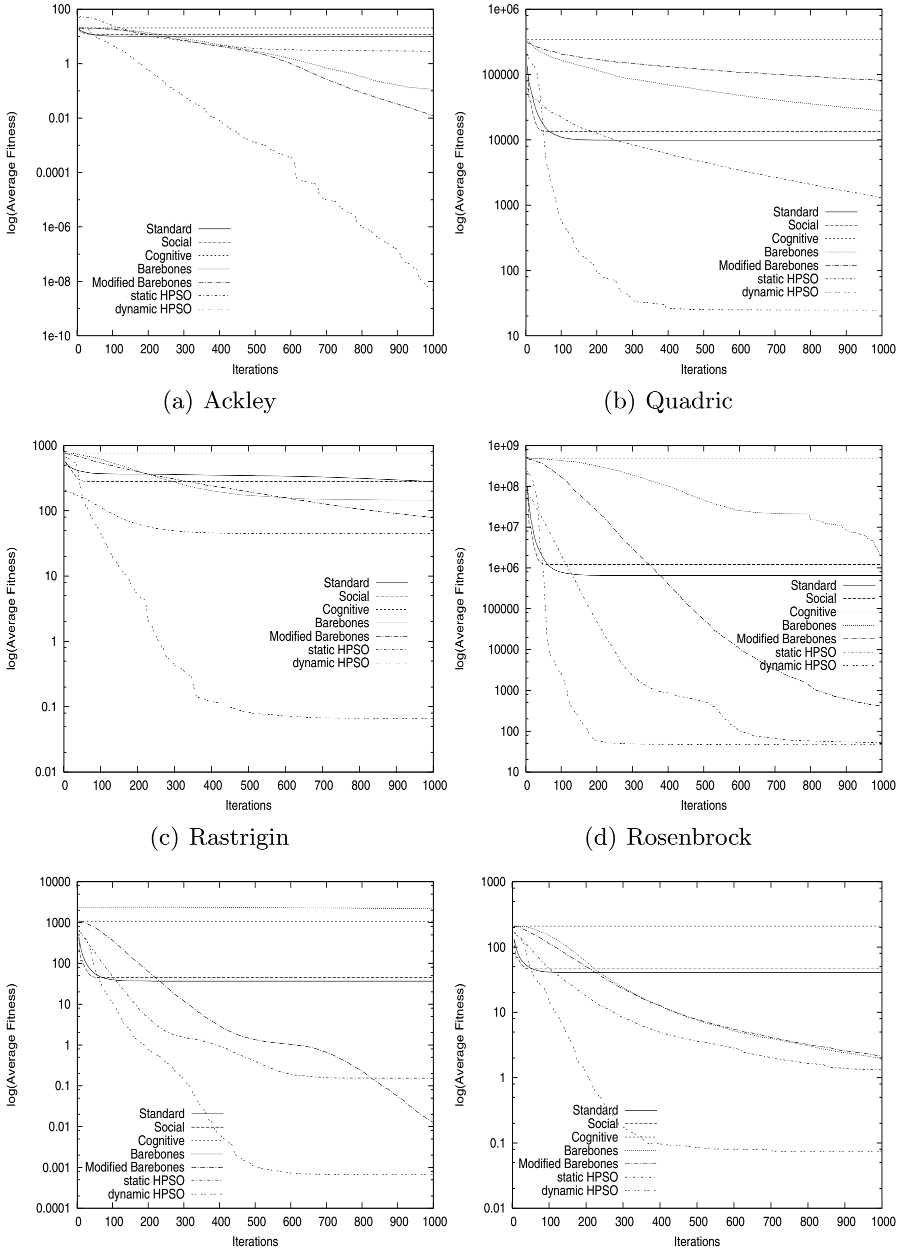
Algorithm	Rosenbrock	Salomon	Griewank
	10 Dimensions		
Standard PSO	7.39E+01±1.25E+02	2.01E+00±1.41E+00	3.33E-01±2.52E-01
Social PSO	2.16E+03±7.69E+03	3.09E+00±1.79E+00	4.40E-01±2.46E-01
Cognitive PSO	2.81E+07±1.58E+07	7.21E+01±8.22E+00	1.24E+02±3.28E+01
Barebones PSO	9.09E+00±1.72E+01	1.30E-01±4.66E-02	7.48E-02±4.25E-02
Modified Barebones	4.82E+00±6.19E+00	9.99E-02±1.12E-07	1.43E-02±1.55E-02
Static HPSO	1.89E+00±9.15E+00	1.47E-01±5.07E-02	7.82E-02±4.68E-02
Dynamic HPSO	4.91E+00±3.12E+00	4.99E-02±5.08E-02	5.59E-02±4.38E-02
	30 Dimensions		
Standard PSO	1.52E+05±9.99E+04	2.47E+01±4.09E+00	1.18E+01±3.73E+00
Social PSO	2.11E+05±1.43E+05	2.75E+01±4.40E+00	1.49E+01±4.21E+00
Cognitive PSO	2.28E+08±3.88E+07	1.55E+02±7.61E+00	5.79E+02±5.99E+01
Barebones PSO	7.19E+01±1.02E+02	4.13E-01±7.30E-02	8.12E-03±8.12E-03
Modified Barebones	8.83E+01±4.45E+01	4.02E-01±5.90E-02	7.52E-04±2.27E-03
Static HPSO	1.49E+01±3.89E+01	4.20E-01±8.86E-02	4.07E-02±1.29E-01
Dynamic HPSO	2.64E+01±4.15E-01	7.67E-02±4.28E-02	3.04E-03±7.17E-03
	50 Dimensions		
Standard PSO	6.52E+05±2.60E+05	4.08E+01±4.44E+00	3.67E+01±8.50E+00
Social PSO	1.21E+06±5.08E+05	4.63E+01±4.73E+00	4.50E+01±8.82E+00
Cognitive PSO	4.92E+08±5.55E+07	2.09E+02±7.78E+00	1.08E+03±8.55E+01
Barebones PSO	1.90E+06±1.04E+07	1.97E+00±4.60E-01	1.27E-02±1.93E-02
Modified Barebones	4.01E+02±3.03E+02	2.13E+00±3.66E-01	1.26E-02±1.39E-02
Static HPSO	5.25E+01±1.36E+02	1.31E+00±1.42E+00	1.54E-01±5.57E-01
Dynamic HPSO	4.67E+01±3.94E-01	7.33E-02±4.48E-02	6.65E-04±2.50E-03
	100 Dimensions		
Standard PSO	3.76E+06±2.21E+06	7.34E+01±5.53E+00	1.17E+02±2.85E+01
Social PSO	5.74E+06±1.69E+06	8.08E+01±6.65E+00	1.48E+02±1.68E+01
Cognitive PSO	1.13E+09±8.88E+07	3.11E+02±7.79E+00	2.39E+03±1.27E+02
Barebones PSO	1.15E+09±7.57E+07	2.97E+02±5.43E+01	2.18E+03±5.90E+02
Modified Barebones	4.71E+08±3.56E+08	4.38E+02±9.12E+00	4.39E+01±3.43E+01
Static HPSO	3.38E+03±7.70E+03	1.86E+01±7.94E+00	3.61E+00±4.08E+00
Dynamic HPSO	9.87E+01±5.92E+00	7.99E-02±4.04E-02	1.43E-07±7.84E-07

rank of 2.92. This is a clear illustration that, on average, the HPSO models, specifically the dHPSO, performed better than the homogenous models used within the HPSO models.

Figure 2 shows that the HPSO algorithms were by far the most scalable. In rank order from best scalable to worst scalable (ranks are given in parentheses): dHPSO (1), sHPSO (1.8), modified barebones PSO (3.5), standard gbest PSO (4.2), barebones PSO (5), social-only PSO (5.3), and cognitive-only (6.8). Important to note is that dHPSO was not significantly affected by an increase in dimensions, whereas sHPSO showed a small deterioration in performance compared to the other PSO algorithms. Note that, in general, the homogeneous PSO algorithms' performance deteriorate significantly as the number of dimensions increases.

Table 3. Performance Ranks for All Experiments; Ack (Ackley), Quad (Quadric), Ras (Rastrigin), Ros (Rosenbrock), Sal (Salomon), Grie (Griewank)

Algorithm	Ack	Quad	Ras	Ros	Sal	Grie	Average
	10 Dimensions						
Standard PSO	5	5	5	5	5	5	5
Social PSO	6	6	6	6	6	6	6
Cognitive PSO	7	7	7	7	7	7	7
Barebones PSO	4	1	4	4	3	3	3.17
Modified Barebones	2	4	1	2	2	1	2
Static HPSO	3	2	2	1	4	4	2.67
Dynamic HPSO	1	3	3	3	1	2	2.17
	30 Dimensions						
Standard PSO	5	4	4	5	5	5	4.67
Social PSO	6	5	3	6	6	6	5.33
Cognitive PSO	7	7	6	7	7	7	6.83
Barebones PSO	2	3	7	3	3	3	3.5
Modified Barebones	3	6	1	4	2	1	2.83
Static HPSO	4	2	5	1	4	4	3.33
Dynamic HPSO	1	1	2	2	1	2	1.5
	50 Dimensions						
Standard PSO	5	3	3	4	5	5	4.17
Social PSO	6	4	4	5	6	6	5.17
Cognitive PSO	7	7	6	7	7	7	6.83
Barebones PSO	3	5	2	6	3	3	3.67
Modified Barebones	2	6	7	3	4	2	4
Static HPSO	4	2	5	2	2	4	3.17
Dynamic HPSO	1	1	1	1	1	1	1
	100 Dimensions						
Standard PSO	5	3	3	3	3	4	3.5
Social PSO	6	4	4	4	4	5	4.5
Cognitive PSO	7	7	2	5	6	7	5.67
Barebones PSO	4	5	6	6	5	6	5.33
Modified Barebones	3	6	7	7	7	3	5.5
Static HPSO	2	2	5	2	2	2	2.5
Dynamic HPSO	1	1	1	1	1	1	1
	Average over all Dimensions						
Standard PSO	5	3.75	3.75	4.25	4.5	4.75	4.33
Social PSO	6	4.75	4.25	5.25	5.5	5.75	5.25
Cognitive PSO	7	7	5.25	6.5	6.75	7	6.58
Barebones PSO	3.25	3.5	4.75	4.75	3.5	3.75	3.92
Modified Barebones	2.5	5.5	4	4	3.75	1.75	3.58
Static HPSO	3.25	2	4.25	1.5	3	3.5	2.92
Dynamic HPSO	1	1.5	1.75	1.75	1	1.5	1.42

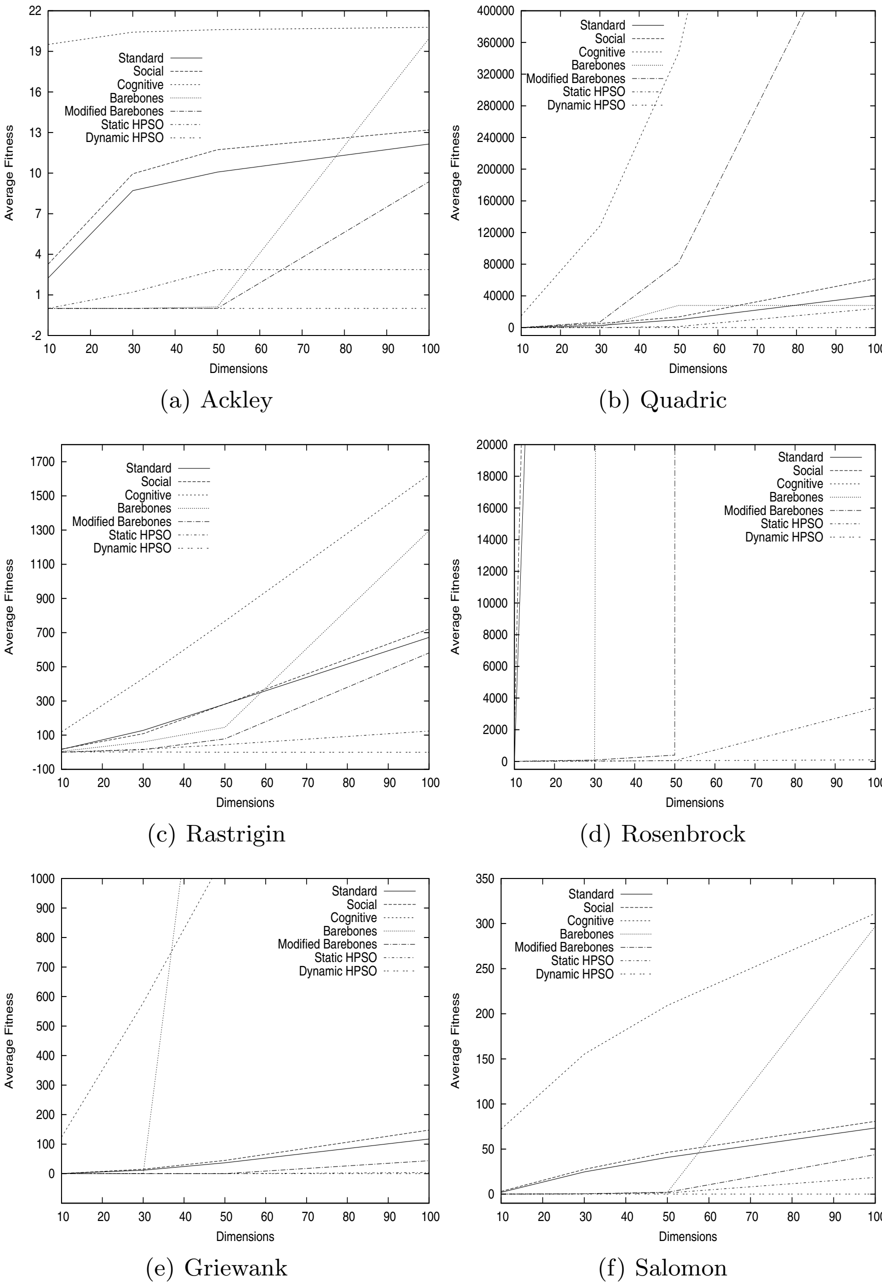


(a) Ackley (b) Quadric

(c) Rastrigin (d) Rosenbrock

(e) Griewank (f) Salomon

Fig. 1. Accuracy Profile for All Functions in 50 Dimensions



(a) Ackley (b) Quadric

(c) Rastrigin (d) Rosenbrock

(e) Griewank (f) Salomon

Fig. 2. Algorithm Scalability

5 Conclusions

This paper proposed a heterogeneous PSO (HPSO), where the particles of the swarm implements different behaviours in terms of the particle position and velocity updates. Two versions of the HPSO were proposed, the one static, where behaviours do not change, and a dynamic version where a particle may change its behaviour during the search process if it can not improve its personal best position. Both versions initialize particle behaviours by randomly selecting behaviours from a behaviour pool. When a particle in the dynamic HPSO has to change its behaviour, a new behaviour is randomly selected from the pool of behaviours.

For the purposes of this preliminary study, five different behaviours were included in the behaviour pool. These behaviours differ in the degree of exploration and exploitation, and how exploration is balanced with exploitation.

The empirical results have shown that the dynamic HPSO significantly outperformed the other two approaches in terms of the quality of the optima found and in terms of scalability. The static HPSO was the second best performer. This indicates that the HPSO has potential to be further explored. Future research will develop different models for behaviour change and will investigate other mechanisms to trigger a behaviour change.

References

1. Blackwell, T., Bentley, P.: Dynamic Search with Charged Swarms. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 19–26 (2002)
2. Brits, R., Engelbrecht, A., van den Bergh, F.: A Niching Particle Swarm Optimizer. In: Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning, pp. 692–696 (2002)
3. Eberhart, R., Kennedy, J.: A New Optimizer using Particle Swarm Theory. In: Proceedings of the Sixth International Symposium on Micromachine and Human Science, pp. 39–43 (1995)
4. Engelbrecht, A.: Fundamentals of Computational Swarm Intelligence. Wiley & Sons, Chichester (2007)
5. Engelbrecht, A.: CIlib: A Component-based Framework for Plug-and-Simulate. In: International Conference on Hybrid Computational Intelligence Systems, Barcelona, Spain (2008) (Invites Talk)
6. Kennedy, J.: The Particle Swarm: Social Adaptation of Knowledge. In: Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 303–308 (1997)
7. Kennedy, J.: Bare Bones Particle Swarms. In: Proceedings of the IEEE Swarm Intelligence Symposium, pp. 80–87 (2003)
8. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of the IEEE International Joint Conference on Neural Networks, pp. 1942–1948 (1995)
9. Krink, T., vberg, M.L.: The Life Cycle Model: Combining Particle Swarm Optimisation, Genetic Algorithms and Hill Climbers. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 621–630. Springer, Heidelberg (2002)

10. de Oca, M.M., Pena, J., Stuetzle, T., Pinciroli, C., Dorigo, M.: Heterogeneous Particle Swarm Optimizers. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 689–709 (2009)
11. Olorunda, O., Engelbrecht, A.: An Analysis of Heterogeneous Cooperative Algorithms. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1562–1569 (2009)
12. Ratnaweera, A., Halgamuge, S., Watson, H.: Particle Swarm Optimiser with Time Varying Acceleration Coefficients. In: Proceedings of the International Conference on Soft Computing and Intelligent Systems, pp. 240–255 (2002)
13. Silva, A., Neves, A., Costa, E.: An Empirical Comparison of Particle Swarm and Predator Prey Optimisation. In: O'Neill, M., Sutcliffe, R.F.E., Ryan, C., Eaton, M., Griffith, N.J.L. (eds.) AICS 2002. LNCS (LNAI), vol. 2464, pp. 103–110. Springer, Heidelberg (2002)
14. Spanevello, P., de Oca, M.M.: Experiments on Adaptive Heterogeneous PSO Algorithms. In: Proceedings of the Doctoral Symposium on Engineering Stochastic Local Search Algorithms, pp. 36–40 (2009)
15. van den Bergh, F., Engelbrecht, A.: A New Locally Convergent Particle Swarm Optimizer. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 96–101 (2002)
16. Vesterstrøm, J., Riget, J., Krink, T.: Division of Labor in Particle Swarm Optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1570–1575. IEEE Press, Los Alamitos (2002)