Hierarchical heterogeneous particle swarm optimization: algorithms and evaluations

Xinpei Ma\textsuperscript{a,b} and Hiroki Sayama\textsuperscript{a,b}

\textsuperscript{a}Department of Systems Science & Industrial Engineering, Binghamton University, State University of New York, Binghamton, NY, USA; \textsuperscript{b}Center for Collective Dynamics of Complex Systems, Binghamton University, State University of New York, Binghamton, NY, USA

ABSTRACT
Particle swarm optimization (PSO) has recently been extended in several directions. Heterogeneous PSO (HPSO) is one of such recent extensions, which implements behavioural heterogeneity of particles. In this paper, we propose a further extended version, Hierarchical Heterogeneous PSO (HHPSO), in which heterogeneous behaviors of particles are enforced through interactions among hierarchically structured particles. Two algorithms have been developed and studied: multi-layer HHPSO (ml-HHPSO) and multi-group HHPSO (mg-HHPSO). In each HHPSO algorithm, stagnancy and overcrowding detection mechanisms were implemented to avoid premature convergence. The algorithm performance was measured on a set of benchmark functions and compared with performances of standard PSO (SPSO) and HPSO. The results demonstrated that both ml-HHPSO and mg-HHPSO performed well on all testing problems and significantly outperformed SPSO and HPSO in terms of solution accuracy, convergence speed and diversity maintenance. Further computational experiments revealed the optimal frequencies of stagnation and overcrowding detection for each HHPSO algorithm.

1. Introduction
Particle swarm optimization (PSO), a population-based computational meta-heuristic, has an increasing impact on many areas of science and engineering \cite{1,15,29,31}. PSO is effective at solving a wide range of optimization problems \cite{29}. The algorithm starts with a randomly generated population of particles moving in a search space. Using the positions of the global best solution and the personal best solution each particle has achieved so far, the particles accelerate toward better solutions in the search space \cite{22}. Unlike the majority of other search heuristics, PSO continually maintains a population of potential solutions to the problem instead of only one solution. The standard PSO algorithm (SPSO) uses the same updating rules for all particles. PSO’s algorithmic design is simple and computationally efficient \cite{5,12}.

Many variants of PSO have demonstrated that the performance of PSO algorithms is largely determined by particles’ searching behaviors and interactions \cite{20,39}. Among recent extensions of PSO, introduction of hierarchical interactions and particle heterogeneity is of our particular interest. To the extent of our knowledge, none of the earlier works combined hierarchical structures and heterogeneous behaviors in PSO.
In this paper, we propose a new PSO algorithm, \textit{Hierarchical Heterogeneous PSO} (HHPSO), to integrate hierarchical structures and heterogeneous behaviors in PSO. The initial ideas of HHPSO and preliminary results have been published in [26]. This paper is an extension of the HHPSO framework, especially focusing on algorithm development and performance evaluations.

Our rationale in combining hierarchical structure and heterogeneous behaviors is as follows. First, many collective systems from both natural and engineered realms possess multi-level structures to enhance functional connectivity, system stability and information exchange [6,9,11]. Adding a hierarchical structure to PSO algorithm develops multi-level cooperative interactions between particles [24].

Second, behavior heterogeneity is a ubiquitous property in many biological systems [6]. By making the PSO heterogeneous, particles track the global optima using diverse velocity and position updating rules rather than being fixed at the beginning. Having functionally specialized particles may alleviate the swarm’s premature convergence and thereby the search performance may go beyond that of a swarm with homogeneous behaviors [14,35].

Third, combining hierarchical structures and heterogeneous behaviors links particles’ heterogeneous behaviors to population structures [26]. Particles’ behavior modes are dynamically adjusted based upon the searching performance of the swarm [41]. Implementing a structure-driven behavioural heterogeneity to PSO algorithm may therefore be of an advantage to balance exploration and exploitation in the search process.

The rest of this paper is organized as follows. In Section 2, we review two relevant developments in PSO algorithms, which are population structure based PSO and heterogeneous PSO. In Section 3, we describe the two proposed new HHPSO algorithms (ml-HHPSO and mg-HHPSO). Two experiments and corresponding results are described in Sections 4 and 5. In Section 4, a set of benchmark functions is used to test the two HHPSO algorithms and to compare them with the standard PSO and heterogeneous PSO. In Section 5, we further explore the parameter space of the two HHPSO algorithms. Conclusions, discussion and future work are presented in Section 6.

2. Related work

2.1. PSO algorithms with population structures

Population topologies and interaction modes are important aspects of population-based optimizations. Over the last decade, designs of different population topologies, both static and dynamic (i.e. time varying), have been actively studied in PSO research [8,23,27].

Kennedy and Mendes systematically explored the effects of static population structures on PSO [23]. They found that topologies in which some particles were highly connected while others were relatively isolated could improve the algorithm performance. For instance, PSO with a ‘von Neumann’ configuration significantly outperformed the standard PSO.

A number of variants of PSO have been developed using dynamic population structures. Hu and Eberhart proposed a dynamic neighborhood PSO (DNPSO) to solve multi-objective optimization problems [16]. Veeramachaneni proposed a Fitness-Distance ratio PSO (FDR-PSO) to combine the effects of neighbors, where particles are attracted towards the personal best positions visited by the neighboring particles, which have better fitness values [30].

Multi-swarm PSO algorithms have also been developed, which divide the population of particles into a number of interacting sub-swarms. Yan et al. proposed a multi-swarm PSO to alleviate premature convergence [18], in which each sub-swarm performs optimization independently. After every certain number of iterations, the whole swarm is shuffled, and particles are reassigned to equally sized sub-swarms to improve the information exchange. Blackwell and Branke proposed a multi-swarm PSO with interacting sub-swarms [4]. Interactions between sub-swarms are achieved by allowing particles to move towards attractors from other sub-swarms.
Figure 1. Schematic diagrams of multi-layer HHPSO (ml-HHPSO) (a) and multi-group HHPSO (mg-HHPSO) (b). In panel (a), particles are structured into multiple layers based on their current fitness values; particles in each layer are attracted to those in the layer above (left). An illustrative example of particles in a search space is shown (right), with attractive forces shown by arrows of different colours. In panel (b), particles are randomly split into multiple groups, inside each of which they are further structured into three layers based on their fitness values (red: head, blue: tail, yellow: body). Groups are spatially clustered in this figure just for visual clarity, but they can overlap in real cases. Body particles are attracted to others with higher fitness values within its group (black arrows). Heads are attracted to other heads with higher fitness values (red arrows). Tails are attracted to all heads (blue arrows).

Although the existing literature demonstrated that effective population structures and information exchange mechanisms can improve the performance of algorithms, the searching strategies and mechanisms typically remain static during the whole searching process.

2.2. PSO Algorithms with Heterogeneous Behaviors

Heterogeneity of particles’ behaviors in PSO algorithms has started to draw the attention of researchers recently. Several efforts have been made to improve the algorithm performance by introducing behavioural heterogeneity.

Jie et al. [20] developed a self-organization PSO (SOPSO). The algorithm aims to maintain population diversity throughout the search. Three different searching behaviors, i.e. global search, local search and mutation, are selected by particles. Particles switch their behaviors based on changes of population diversity.

Silva et al. proposed a predator-prey PSO [36]. Their algorithm tries to better control the balance between exploration and exploitation by introducing predators and prey particles. Having prey particles repelled by predators facilitates exploration, while particles’ regrouping behaviors promotes exploitation.

Engelbrecht proposed two versions of heterogeneous PSO (HPSO) [13]. The first one is named static HPSO (sHPSO), where behaviors are randomly assigned to particles during the searching process. The second is dynamic HPSO (dHPSO), where particles select different search behaviors once they become stagnant, i.e. they fail to develop their personal best solutions in a certain window of iterations.

Nepomuceno and Engelbrecht proposed a self-adaptive HPSO [28]. The algorithm is inspired by the foraging behavior of ants. The behavior selection process is similar but more adaptive compared with either sHPSO or dHPSO. For each behavior in the behavior pool, the amount of pheromone concentration is calculated based on the behavior’s contribution to the development of particles’ solutions. Instead of randomly selecting a behavior from a predefined behavior pool, particles tend to select an optimal behavior with a high pheromone concentration.

For the heterogeneous PSO algorithms presented in the literature, different searching behaviors were designed to adapt to changes [10,13,28]. Their results were promising in terms of accuracy and convergence speed. However, these algorithms have shortcomings. First, none of them relate heterogeneity of behaviors to heterogeneity in population structures. Second, how to coordinate different heterogeneous behaviors during the search remains unexplored.
Algorithm 1 ml-HHPSO In this algorithm, $S$ represents the swarm. $S_i$ represents particle $i$. $n_s$ is the number of particles in the swarm. $n_x$ is the number of dimensions of the search space. $f$ represents the fitness function. $y_i$ represents the personal best position of particle $i$ at time $t$. $\hat{y}$ represents the global best position. $L_j$ represents layer $j$. $m$ and $k$ are the number of layers and the number of particles in a layer, respectively.

1: for each iteration do
2:   # Update personal best position and global best position
3:   for each $S_i$, $i = 1, 2, \ldots, n_s$ do
4:     if $f(x_i) < f(y_i)$ then
5:       $y_i = x_i$
6:     $\hat{y} = \min\{y_i | i = 1, 2, \ldots, n_s\}$
7:   # Organize particles into $m$ layers
8:   $(R_1, R_2, \ldots, R_{n_s}) \leftarrow \text{Sort}(S_1, S_2, \ldots, S_{n_s})$
9:   in an increasing order of $y_1, \ldots, y_{n_s}$
10: for $j \in [0, m)$ do
11:   $L_j \leftarrow \{R_{js_k+1} \mid R_{js_k+2} \ldots, R_{js_k+k}\}$
12: # Update velocities and positions
13: for each $S_i$, $i = 1, 2, \ldots, n_s$ do
14:   if $S_i$ has detected early stagnation or overcrowding then
15:     Randomly select a new behavior from the behavior pool
16:     else
17:       if $S_i \in L_0$ then
18:         Attractors $\leftarrow \{S_i \in L_0 \mid y_i < y_j\}$
19:       else
20:         Attractors $\leftarrow \{y_{i-1}\}$
21: Update velocity and position using Equations (2) and (3)

3. Proposed algorithms

In the original PSO algorithm, the searching behavior of each particle is guided by two pieces of information: personal best solution and global best solution. At each time step, the particles are updated by the following equations:

$$V_{i,j}^{t+1} = V_{i,j}^{t} \times \omega + c_1 r_{1,i,j}^{t}(\hat{y}_{i,j}^{t} - x_{i,j}^{t}) + c_2 r_{2,i,j}^{t}(y_{i,j}^{t} - x_{i,j}^{t}) \quad (1)$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + V_{i,j}^{t+1} \quad (2)$$

Here, $V_{i,j}^{t}$ denotes the velocity value of particle $i$ at time $t$, $x_{i,j}^{t}$ is the particle $i$’s current position at time $t$, $y_{i,j}^{t}$ is the personal best solution of particle $i$ at time $t$, and $\hat{y}_{i,j}^{t}$ is the global best solution known at time $t$. Subscript $j$ is the index of the spatial dimension. $\omega$ is a parameter called inertia weight representing how much particles’ memory can influence the new position, $r_{1,i,j}^{t}$ and $r_{2,i,j}^{t}$ are two random numbers, and $c_1$ and $c_2$ are two constant acceleration coefficients. They are used to balance exploration and exploitation behaviors.

In this paper, we revise the above algorithm to propose Hierarchical Heterogeneous PSO (HHPSO) algorithms that integrate heterogeneous behaviors of particles and hierarchical interaction patterns into PSO. The algorithm design is a two-step procedure. First, all particles are arranged into a hierarchical structure based on their current fitness values (particles with higher fitness values move to a higher level, while particles with lower fitness values move to a lower level). Second, particles are assigned different searching behaviors based on their ranks in the hierarchy and their performances. Each particle’s fitness value is evaluated through an $n$-dimensional objective function $f(x_1, x_2, \ldots, x_n)$, $f : R^n \rightarrow R$. In this study, the fitness functions are defined so that lower fitness values represent better solutions. PSO tries to find solutions that minimize the given fitness function.

In our HHPSO algorithms, particles are attracted toward the personal best positions of other attractor particles in the swarm as well as their own personal best and the global best solutions. Therefore, in the new algorithm design, the velocity updating formula (Equation 1) was modified by...
Algorithm 2 mg-HHPSO

In this algorithm, \( S \) represents the swarm. \( S_i \) represents particle \( i \). \( n_x \) is the number of particles in the swarm. \( n_x \) is the number of dimensions of the search space. \( H, T \) and \( B \) represent the three sets, namely, head particles, tail particles and body particles, respectively. \( y_i \) represents the personal best position of particle \( i \) at time \( t \). \( \hat{y} \) represents the global best position, and \( f \) represents the fitness function. \( G_n \) represents group \( n \). The number of groups is \( m \). The number of particles in each group is \( k \).

1: for each iteration do  
2:    # Update personal best position and global best position  
3:    for each \( S_i, i = 1, 2, \ldots, n_x \) do  
4:        if \( f(x_i) < f(y_i) \) then  
5:            \( y_i = x_i \)  
6:        end if  
7:    end for  
8:    \( \hat{y} = \min\{y_i | i = 1, 2, \ldots, n_x \} \)  
9:    # Identify head, tail and body particles  
10:   Randomly arrange particles into \( m \) groups  
11:   for each group \( G_i, i = 1, 2, \ldots, m \) do  
12:        \( \{R_{i,1}, R_{i,2}, \ldots, R_{i,k}\} \leftarrow \text{Sort} \{G_{i,1}, G_{i,2}, \ldots, G_{i,k}\} \) in order of increasing personal best fitness values  
13:        \( H_i \leftarrow \{R_{i,1}\} \)  
14:        \( T_i \leftarrow \{R_{i,k}\} \)  
15:        \( B_i \leftarrow \{R_{i,2}, R_{i,3}, \ldots, R_{i,k-1}\} \)  
16:    \( H \leftarrow H_1 \cup H_2 \cup \ldots \cup H_m \)  
17:    \( T \leftarrow T_1 \cup T_2 \cup \ldots \cup T_m \)  
18:    \( B \leftarrow B_1 \cup B_2 \cup \ldots \cup B_m \)  
19:    # Update velocities and positions  
20:    for each \( S_i, i = 1, 2, \ldots, n_x \) do  
21:        if \( S_i \) has detected early stagnation or overcrowding then  
22:            Randomly select a new behavior from the behavior pool  
23:        else  
24:            if \( S_i \in H \) then  
25:                Attractor\( \leftarrow \{S_{i'} \in H | y_{i'} < y_i\} \)  
26:            else if \( S_i \in B_x, x = 1, 2, \ldots, m \) then  
27:                Attractor\( \leftarrow \{S_{i'} \in G_x | y_{i'} < y_i\} \)  
28:            else  
29:                Attractor\( \leftarrow \{S_{i'} \in H\} \)  
30:        end if  
31:    end for  
32:    Update velocity and position using Equations (2) and (3)

\[
V_{i,j}^{t+1} = V_{i,j}^t + \omega + c_1r_{1,ij}^t (\hat{y}_{i,j}^t - x_{i,j}^t) + c_2r_{2,ij}^t (y_{i,j}^t - x_{i,j}^t)
+ \sum_{a=1}^{A_i^t} c_3r_{3,ij}^t (x(i)_{a,j}^t - x_{i,j}^t) \tag{3}
\]

Here, \( x(i)_{a,j}^t \) is the position of attractor particle \( a \) of particle \( i \) in dimension \( j \) at time \( t \). \( A_i^t \) is the total number of attractors of particle \( i \) at time \( t \). \( c_3 \) is a constant acceleration coefficient and \( r_{3,ij}^t \) is a random number, which are similar with SPSO.

In order to study the effects of varying population structures and communication channels on PSO, we have tested two hierarchical structures, and thus two corresponding HHPSO algorithms, multi-layer HHPSO (ml-HHPSO) and multi-group HHPSO (mg-HHPSO), have been proposed. Multi-layer HHPSO (ml-HHPSO) focuses on establishing vertical interactions between multiple layers (Figure 1). Its algorithm is shown in Algorithm 1. The population structure consists of equally sized layers. In each iteration, particles are sorted by their current fitness values and arranged into a hierarchical structure. Particles in the upper layer always have better fitness values than particles in the lower layer. In general, particles are attracted by particles in their immediate superior layer, and also attract particles in their intermediate inferior layer. However, particles in the uppermost layer are exceptions, they are attracted toward other particles in the same layer but have better fitness values.

Adding an additional term of attraction from the particle’s attractors, as follows:

\[
V_{i,j}^{t+1} = V_{i,j}^t + \omega + c_1r_{1,ij}^t (\hat{y}_{i,j}^t - x_{i,j}^t) + c_2r_{2,ij}^t (y_{i,j}^t - x_{i,j}^t) + \sum_{a=1}^{A_i^t} c_3r_{3,ij}^t (x(i)_{a,j}^t - x_{i,j}^t) \tag{3}
\]
Additionally, all particles are also attracted to the global best and their personal best positions. This mechanism is quite similar with SPSO.

The second HHPSO algorithm, multi-group HHPSO (mg-HHPSO), aims to establish both vertical (i.e. between-layer) and horizontal (i.e. between-subpopulation) interactions of particles. Its algorithm is shown in Algorithm 2. The population structure in mg-HHPSO is composed of multiple groups, each of which exhibits a three-level hierarchical structure. At the beginning of each iteration, particles are randomly allocated to equally sized groups. Within each group, particles are further arranged into the hierarchical structure based on their current fitness values. Within each group, head, body and tail particles are established to represent the three-level hierarchy. The head particle and the tail particle are the particles with the lowest (best) and highest (worst) fitness values in the group, respectively. The rest of particles are designated as the body particles. The three types of particles adopt different behaviors to update their positions and velocities. Head particles are attracted by other head particles that have better fitness values. Body particles are responsible for enhancing the group’s fitness. They are attracted to their heads and other body particles with better fitness values. Tail are attracted towards all head particles, which contribute to the improvement of the group’s fitness level from the bottom up. In this algorithm, all particles are also attracted to the global and personal bests, like in SPSO.

In both ml- and mg-HHPSO, we monitor the population to detect two signals of premature convergence: early stagnation and overcrowding. Early stagnation refers to the situation in which a particle fails to develop its personal best solution for a certain number of iterations [12,13,26]. Overcrowding refers to the situation in which some areas in the search space have high particle density [40]. Previous studies have proven that both phenomena are closely related to ineffective searching behaviors and/or susceptibility to local minima [37]. In HHPSO algorithms, if either overcrowding or early stagnation is detected, the relevant particle will randomly select a new updating equation from the behavoural pool, rather than abide by the behavior decided by the hierarchical structure.
Similar with the work done by Engelbrecht [13], six different behavioural rules – SPSO, Social-Only PSO, Cognitive-Only PSO, Barebones PSO [21] and Modified Barebones PSO [21] – are used to form the behavior pool. This strategy keeps track of the particles’ performances during the search, and allows particles to change their behaviors when they have a tendency to get trapped into local minima. Engelbrecht’s HPSO already had such detection mechanisms [13]. In the meantime, this detection mechanism would not be implementable in SPSO because it lacks the behavior pool.

In order to further optimize the combined impact of early stagnation detection and overcrowding detection, we introduced two parameters, $p$ and $q$, to control the frequencies for particles to detect stagnancy and overcrowding, respectively. $p$ and $q$ are defined as the probability of these detections occurring in each iteration. For example, if $p = q = 1$, every particle always checks if it is stagnant, and if it is in an overcrowded area. The algorithm performance with specific values of $p$ and $q$ were evaluated by measuring the quality of solutions and the speed of convergence.

### 4. Experiment I

To evaluate the performance of HHPSO algorithms, we conducted numerical experiments on several benchmark functions. Each HHPSO algorithm starts with a swarm of 50 particles. Particles in PSO are assigned random positions and velocities in a 50-dimensional search space. Five layers and ten groups are used in ml- and mg-HHPSO, respectively. The parameter values used in the experiments are listed in Table 1.

In Experiment I, we measured the performance of HHPSO algorithms without controlling the frequencies of the two detection mechanisms, which is equivalent to holding both $p$ and $q$ constant.

---

**Figure 2.** Average of the best fitness values of SPSO, HPSO, ml-HHPSO and mg-HHPSO for nine benchmark functions: (a) Ackley; (b) Quadric; (c) Rastrigin; (d) Rosenbrock; (e) Griewank; (f) Salomon; (g) Rotated Ackley; (h) Rotated Griewank and (i) Rotated Rastrigin. The shaded regions represent the standard errors. Lower fitness value means better results. Each swarm is made of 50 particles. Results are averaged over 50 independent simulation runs. (b), (c), (e), (h) and (i) are plotted in a symlog scale of matplotlib [17] to show convergence to zero.
Figure 3. Average population diversity of SPSO, HPSO, ml-HHPSO and mg-HHPSO for nine benchmark functions: (a) Ackley; (b) Quadric; (c) Rastrigin; (d) Rosenbrock; (e) Griewank; (f) Salomon; (g) Rotated Ackley; (h) Rotated Griewank and (i) Rotated Rastrigin. Each swarm is made of 50 particles. Results are averaged over 50 independent simulation runs.

Figure 4. Contour plots of the performance of ml-HHPSO (a) and mg-HHPSO (b) under different parameter values for $p$ and $q$. On the left is the algorithm performance of ml-HHPSO applying different frequency values and on the right is the algorithm performance of mg-HHPSO applying different frequency values. The plotted function is the cumulative convergence time of nine benchmark functions. Red points represent the parameter combinations, under which the algorithm failed to converge to the global minima on at least one benchmark. We identified optimal $p$ and $q$ values, which yield high solution quality and fast convergence speed, for ml-HHPSO and mg-HHPSO, respectively. For each algorithm, a white star indicates the pair of selected $p$ and $q$ values ($p = 0.3, q = 0.7$ for ml-HHPSO and $p = 0.2$ and $q = 0.6$ for mg-HHPSO) used for the experiment of performance comparison. The x-axis represents the values of parameter $p$, which is the possibility that early stagnation detection occurs. The y-axis represents the values of parameter $q$, which is the possibility that overcrowding detection occurs.
Figure 5. Average best fitness values of ml-HHPSO ($p = q = 1.0$), ml-HHPSO ($p = q = 0.3, q = 0.7$) and mg-HHPSO ($p = 0.2, q = 0.6$) for nine benchmark functions: (a) Ackley; (b) Quadric; (c) Rastrigin; (d) Rosenbrock; (e) Griewank; (f) Salomon; (g) Rotated Ackley; (h) Rotated Griewank and (i) Rotated Rastrigin. The shaded regions represent the standard errors. Lower fitness means better results. Each swarm is made of 50 particles. Results are averaged over 50 independent simulation runs. (b), (c), (e), (f), (h) and (i) are plotted in a symlog scale of matplotlib [17] to show convergence to zero.

The results were evaluated through comparisons with SPSO and HPSO. The performance of algorithms was measured on nine benchmark functions: (a) Ackley, (b) Quadric, (c) Rastrigin, (d) Rosenbrock, (e) Griewank, (f) Salomon, (g) Rotated Ackley, (h) Rotated Griewank and (i) Rotated Rastrigin [25,38]. Table 2 lists the benchmark functions used in this paper. Each algorithm was run 50 times for each benchmark function.

The results of the first experiment are presented in Figure 2. Both HHPSO algorithms demonstrated better search efficiency than SPSO and HPSO. The quality of solutions was improved significantly for all nine benchmark functions in terms of solution accuracy. For benchmark functions (c) Rastrigin, (d) Rosenbrock, (e) Griewank, (h) Rotated Griewank and (i) Rotated Rastrigin, both HHPSO algorithms quickly converged to a stable point. For the other benchmark functions, the algorithms took a relatively long time to converge to a solution.

We measured the population diversity of swarm $S$ at time $t$ for each of the four algorithms using the following metric [12]:

$$\text{Diversity}(S)^t = \frac{1}{n} \sum_{i=1}^{n} \left( \sum_{j=1}^{n_x} \left( x_{ij}^t - \bar{x}_j^t \right)^2 \right)$$

(4)

Here, $x_{ij}^t$ represents the position of particle $i$, and $\bar{x}_j^t$ is the average position over all particles in dimension $j$ at time $t$. $n_s$ is the swarm size, and $n_x$ is the dimensionality of the problem. Using this measure, we found that both ml-HHPSO and mg-HHPSO maintained greater population diversity over time than other algorithms, especially in the beginning of the optimization (Figure 3).
5. Experiment II

In the second experiment, we aimed at finding the optimal frequencies for stagnation detection and overcrowding detection that ensure high solution quality and short convergence time. Parameters $p$ and $q$ were each varied from 0 to 1, and the number of iterations that the algorithm needed to converge to a stable point for each benchmark function was measured. A total of 180 pairs of $p$ and $q$ were tested for ml-HHPSO and mg-HHPSO (if parameter values with which the algorithms failed to converge until the end, the convergence time can be seen as 1000 iterations). These sets covered a two-dimensional $p$-$q$ parameter space. For each set, we ran the experiment 30 times to average the results. These results allowed us to select optimal values of $p$ and $q$ for each HHPSO algorithm. We then conducted the same experiment as Experiment I using newly obtained optimal values of $p$ and $q$, and compared the algorithm performance with the original one with $p = q = 1$.

The results of the second experiment are shown in Figure 4, which exhibits rectangular regions at the centre of the parameter space for ml-HHPSO and mg-HHPSO where the performances of the algorithms were optimal. The results indicated that, by controlling the frequencies of early stagnation detection and overcrowding detection, HHPSO algorithms were able to further optimize their search performances.

Figure 5 illustrates the optimization results before and after applying an optimal frequency values for $p$ and $q$ on both HHPSO algorithms in a 50-dimensional search space. After applying the optimal values for $p$ and $q$, the quality of best solutions and the convergence speed were further improved substantially for most benchmark functions. The solution quality was significantly improved on (b)

![Figure 6. Average population diversity of ml-HHPSO ($p = q = 1.0$), mg-HHPSO ($p = q = 1.0$), ml-HHPSO ($p = 0.3$, $q = 0.7$) and mg-HHPSO ($p = 0.2$, $q = 0.6$) for nine benchmark functions: (a) Ackley; (b) Quadric; (c) Rastrigin; (d) Rosenbrock; (e) Griewank; (f) Salomon; (g) Rotated Ackley; (h) Rotated Griewank and (i) Rotated Rastrigin. Each swarm is made of 50 particles. Results are averaged over 50 independent simulation runs.](image-url)
Quadric and (f) Salomon functions, and the convergence speed was improved on (a) Ackley, (b) Quadric, (c) Rastrigin, (e) Griewank, (f) Salomon, (g) Rotated Ackley, (h) Rotated Griewank and (i) Rotated Rastrigin. For benchmark function (d) Rosenbrick, after applying optimal $p$ and $q$ values, the algorithm performance was not significantly improved. These results confirmed that the algorithm performance of the two proposed HHPSO algorithms can be further improved by tuning the frequency values for the two detection mechanisms.

We conducted the population diversity analysis again to compare the population diversity changing patterns between HHPSOs before and after applying optimal parameters $p$ and $q$. In Figure 6, we found that after applying the optimal frequency parameters $p$ and $q$, population diversity was enhanced for both ml-HHPSO and mg-HHPSO on all benchmark functions. Tuning parameters for the two types of premature convergence detections provided a simple yet effective way to preserve population diversity.

6. Conclusions

In this paper, we proposed two HHPSO algorithms that combine hierarchical population structures and heterogeneous searching behaviors together. HHPSO with multi-layers population structure focuses on vertical communication (between groups), while HHPSO with multi-group population structure focuses on both vertical and horizontal communication (within and between groups). The communication channels within and/or between subpopulations were precisely designed by allowing some particles which exhibit superior capacity to influence more on others, while letting the bad performers to obtain extra help from others [3]. In both HHPSO algorithms, we applied stagnation and overcrowding detection mechanisms during the search to preclude premature convergence.

The results built by ml-HHPSO and mg-HHPSO algorithms demonstrated remarkable improvements in both solution accuracy and convergence speed. Stagnation and overcrowding detections are effective approaches that prevent particles from falling into local minima. Applying the optimal frequencies for these two detections, both two HHPSO algorithms further improved in terms of the solution accuracy and convergence speed.

The significant improvements achieved by HHPSO algorithms can be attributed to some underlying principles, which include non-centralised control [34], hierarchical communication and self-adaptive behavioural heterogeneity [32]. Compared to SPSO and HPSO algorithms, the two HHPSO algorithms are more flexible and adaptive in terms of cooperative modes and behavior selections.

Implementing dynamic hierarchical structures facilitates the establishment of a multitude of interactions between the exploration and the exploitation particles [2,3]. Hence, the optimizer can explore the search space more efficiently. Introduction of heterogeneous behaviors to the PSO algorithm maximises the behavior diversity and discourages premature convergence. Monitoring early stagnation and overcrowding for each particle improves the effectiveness of individual searching behaviors. A high degree of freedom makes particles strong enough to jump out of local minima, so the swarm is more likely to avoid premature convergence [7].

The present study has several limitations. First, the algorithm performances were examined on a limited number of benchmark functions. While it was a reasonable first step in the investigation of evaluating the performances of new algorithms, it would be desirable to compare algorithm performance using more diverse, real-world problems. In addition, different kinds of performance measurements, such as reliability and robustness, should also be taken into account in the future study [12,19,37]. Second, some important parameters we used to establish the new algorithm, such as the number of groups and the number of hierarchies, were decided based on limited numbers of experiments without substantial justification. A study of parameter sensitivity will be our future work. Last but not the least, further study will pay more attention to reduce the computational cost by removing/combining/modifying parameters involved in HHPSO algorithms and simplify the algorithm design.
Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This material is based upon work supported by the US National Science Foundation [grant number 1319152].

ORCID

Xinpei Ma http://orcid.org/0000-0001-8119-8511
Hiroki Sayama http://orcid.org/0000-0002-2670-5864

References

[1] M.R. AlRashidi, and M.E. El-Hawary, A survey of particle swarm optimization applications in electric power systems, IEEE Trans. Evol. Comput. 13 (2009), pp. 913–918.
[2] N. Bellomo, C. Bianca, and M. Delitala, Complexity analysis and mathematical tools towards the modelling of living systems, Phys. Life Rev. 6 (2009), pp. 144–175.
[3] F. BerghVan den, and A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Trans. Evol. Comput. 8 (2004), pp. 225–239.
[4] T. Blackwell, and J. Branke, Multi-swarm optimization in dynamic environments, in EvoWorkshops G.R. Raidl, S. Cagnoni, J. Branke, D.W. Corne, R. Drechsler, Y. Jin, C.G. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G.D. Smith, G. Squillero, Vol. 3005, London: Springer, 2004, pp. 489–500.
[5] D. Bratton and J. Kennedy, Defining a standard for particle swarm optimization, Swarm Intelligence Symposium, SIS 2007. IEEE 2007, Honolulu, HI, 2007, pp. 120–127.
[6] S. Camazine, Self-organization in Biological Systems, Princeton University Press, 2003.
[7] A. Carlisle and G. Dozier, Adapting particle swarm optimization to dynamic environments, Proceedings of the International Conference on Artificial Intelligence 1, Athens, 2000, pp. 429–434.
[8] D. Chen, and C. Zhao, Particle swarm optimization with adaptive population size and its application, Appl. Soft Comput. 9 (2009), pp. 39–48.
[9] B.J. Crespi, The evolution of social behavior in microorganisms, Trends Ecol. Evol. 16 (2001), pp. 178–183.
[10] M.A.M. De Oca, J. Peña, T. Stützle, C. Pinciroli and M. Dorigo, Heterogeneous particle swarm optimizers, IEEE Congress on Evolutionary Computation, 2009. CEC09, Trondheim, 2009, pp. 698–705.
[11] A. De Palma, and C. Lefevre, Individual decision-making in dynamic collective systems, J. Math. Soc. 9 (1983), pp. 103–124.
[12] A.P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, John Wiley & Sons, Pretoria, 2006.
[13] A.P. Engelbrecht, Heterogeneous particle swarm optimization, in Swarm Intelligence, M. Dorigo, M. Birattari, G.A. Di Caro, R. Doursat, A.P. Engelbrecht, D. Floreano, L.M. Gambardella, R. Groß, E. Şahin, H. Sayama, T. Stützle, eds., Springer, Pretoria, 2010, pp. 191–202.
[14] A., Hara, K., Shiraga, and T., Takahama, Heterogeneous particle swarm optimization including predator-prey relationship, 2012 Joint 6th International Conference on Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), Kobe, 2012, pp. 1368–1373.
[15] Q. He and L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (2007), pp. 89–99.
[16] X. Hu and R. Eberhart, Multiobjective optimization using dynamic neighborhood particle swarm optimization, WCCI, West Lafayette, IN, 2002, pp. 1677–1681.
[17] J. Hunter and D. Dale, The matplotlib users guide, Matplotlib 0.90. 0 Users Guide, Barcelona, 2007.
[18] Y. Jiang, T. Hu, C. Huang, and X. Wu, An improved particle swarm optimization algorithm, Appl. Math. Comput. 193 (2007), pp. 231–239.
[19] M. Jiang, Y. Luo, and S. Yang, Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm, Inform. Process. Lett. 102 (2007), pp. 8–16.
[20] J. Jie, J. Zeng, and C. Han, Self-organization particle swarm optimization based on information feedback, in Advances in Natural Computation, Springer, Xi’an, 2006, pp. 913–922.
[21] J. Kennedy, Bare bones particle swarms, Proceedings of the 2003 IEEE Swarm Intelligence Symposium, SIS’03, 2003, Indianapolis, IN, 2003, 80–87.
[22] J. Kennedy, J.F. Kennedy, R.C. Eberhart, and Y. Shi, Swarm intelligence, Morgan Kaufmann, 2001.
[23] J. Kennedy, and R. Mendes, Population structure and particle swarm performance, Washington, DC: IEEE computer Society, 2002.
[24] X. Li, and X. Yao, *Cooperatively coevolving particle swarms for large scale optimization*, IEEE Trans. Evol. Comput. 16 (2012), pp. 210–224.

[25] J.J. Liang, A.K. Qin, P.N. Suganthan, and S. Baskar, *Comprehensive learning particle swarm optimizer for global optimization of multimodal functions*, IEEE Trans. Evol. Comput. 10 (2006), pp. 281–295.

[26] X. Ma and H. Sayama, *Hierarchical heterogeneous particle swarm optimization*, ALIFE 14: The fourteenth conference on the synthesis and simulation of living systems Vol. 14, Binghamton, NY, pp. 3–5.

[27] R. Mendes, *Population topologies and their influence in particle swarm performance*, Ph.D. thesis, Citeseer, Guimaraes, 2004.

[28] F.V. Nepomuceno and A.P. Engelbrecht, *A self-adaptive heterogeneous pso inspired by ants*, in *Swarm Intelligence*, M. Dorigo, M. Birattari, C. Blum, A.L. Christensen, A.P. Engelbrecht, R. Groß, T. Stützle, eds., Springer, Pretoria, 2012, pp. 188–195.

[29] K.E. Parsopoulos, Particle Swarm Optimization and Intelligence: Advances and Applications, IGI Global, Epirus, 2010.

[30] T. Peram, K. Veeramachaneni, and C.K. Mohan, *Fitness-distance-ratio based particle swarm optimization*, Proceedings of the 2003 IEEE Swarm Intelligence Symposium, 2003. SIS’03, Indianapolis, IN, 2003, pp. 174–181.

[31] R. Poli, *Analysis of the publications on the applications of particle swarm optimization*, J. Artif. Evol. Appl. 2008 (2008), p. 3.

[32] A. Ratnaweera, S.K. Halgamuge, and H.C. Watson, *Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients*, IEEE Trans. Evol. Comput. 8 (2004), pp. 240–255.

[33] R. Salomon, *Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. a survey of some theoretical and practical aspects of genetic algorithms*, BioSystems 39 (1996), pp. 263–278.

[34] H. Sayama, *Decentralized control and interactive design methods for large-scale heterogeneous self-organizing swarms*, in *Advances in Artificial Life*, F. Almeida e Costa, L.M. Rocha, E. Costa, I. Harvey, A. Coutinho, eds., Springer, Binghamton, NY, 2007, pp. 675–684.

[35] H. Sayama, *Four classes of morphogenetic collective systems*, preprint (2014), arXiv:1405.6296.

[36] A. Silva, A. Neves, and E. Costa, *An empirical comparison of particle swarm and predator prey optimization*, in *Artificial Intelligence and Cognitive Science*, Springer, Castelo Branco, Portugal, 2002, pp. 103–110.

[37] I.C. Trelea, *The particle swarm optimization algorithm: convergence analysis and parameter selection*, Inform. Process. Lett. 85 (2003), pp. 317–325.

[38] J. Vesterstrøm and R. Thomsen, *A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems*, Congress on Evolutionary Computation, 2004. CEC2004, Portland, 2004 Vol 2, pp. 1980–1987.

[39] X.F. Xie, W.J. Zhang, and Z.I. Yang, *Overview of particle swarm optimization*, Control Decis. 18 (2003), pp. 129–134.

[40] S. Yang, and C. Li, *A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments*, IEEE Trans. Evol. Comput. 14 (2010), pp. 959–974.

[41] Z.H. Zhan, J. Zhang, Y. Li, and H.S.H. Chung, *Adaptive particle swarm optimization*, IEEE Trans, Syst. Man Cybern. Part B: Cybern. 39 (2009), pp. 1362–1381.