Dynamic Heterogeneous Particle Swarm Optimization

Shiqin YANG†, Student Member and Yuji SATO††, Nonmember

SUMMARY Recently, the Static Heterogeneous Particle Swarm Optimization (SHPSO) has been studied by more and more researchers. In SHPSO, the different search behaviours assigned to particles during initialization do not change during the search process. As a consequence of this, the inappropriate population size of exploratory particles could leave the SHPSO with great difficulties of escaping local optima. This motivated our attempt to improve the performance of SHPSO by introducing the dynamic heterogeneity. The self-adaptive heterogeneity is able to alter its heterogeneous structure according to some events caused by the behaviour of the swarm. The proposed triggering events are confirmed by keeping track of the frequency of the unchanged global best position \((p_g)\) for a number of iterations. This information is then used to select a new heterogeneous structure when \(p_g\) is considered stagnant. According to the different types of heterogeneity, DHPSo-d and DHPSo-p are proposed in this paper. In, particles dynamically use different rules for updating their position when the triggering events are confirmed. In DHPSo-p, a global gbest model and a pairwise connection model are automatically selected by the triggering configuration. In order to investigate the scalability of and DHPSo-p, a series of experiments with four state-of-the-art algorithms are performed on ten well-known optimization problems. The scalability analysis of and DHPSo-p reveals that the dynamic self-adaptive heterogeneous structure is able to address the exploration-exploitation trade-off problem in PSO, and provide the excellent optimal solution of a problem simultaneously.

key words: Particle Swarm Optimization (PSO), dynamic heterogeneity, the gbest topology, the pairwise connection topology

1. Introduction

Particle Swarm Optimization (PSO) [1] is a stochastic, population-based optimization method, which has been successfully applied to a number of applications. In the vast majority of PSO models, it is assumed that the swarm is composed of homogeneous individuals. Models that consider populations of homogeneous individuals are attractive because of their conceptual simplicity. However, heterogeneity is ubiquitous in nature. Recently, heterogeneous systems have drawn the attention of researchers working in different areas of swarm intelligence because designing heterogeneous models more accurately and approximately resembles real circumstances.

In general, a particle swarm is heterogeneous if it has at least a pair of particles that differ in any of the four aspects: the neighbourhood size, the model of influence, the update rules and their parametrization [2]. In this paper, we only empirically study two types of heterogeneity, update-rule heterogeneity and model-of-influence heterogeneity. If different particles use different rules for updating their position in the search space, then the swarm exhibits update-rule heterogeneity. When particles in a swarm use different mechanisms for choosing their informers, we say that the swarm exhibits model-of-influence heterogeneity. It was shown in [3], [4] that the Heterogeneous PSO (HPSO) model produced significantly better solutions than a selection of homogeneous PSO models.

To the best of our knowledge, the Static HPSO (SHPSO) has been studied by some researchers, while the Dynamic HPSO (DHPSo) is seldom systematically investigated based on real problems. This paper extends the analysis of DHPSo models to study their scalability to different dimensional optimization problems. The scalability analysis is performed in comparison with two homogeneous PSO models, a well-known Firefly Algorithm (FA) [5], [6] and one SHPSO model (HGLPSO). For the purpose of this study, 10 classical unconstrained benchmark problems were used in dimensions of 50 and 100. The results reported in this paper show that our proposed DHPSo models have excellent scalability properties.

2. Related Work

2.1 Standard Particle Swarm Optimization (SPSO)

The original PSO algorithm [1] was inspired by the social behaviour of biological organisms, e.g. birds flocking while searching for a food source in a given area. Due to its conceptual simplicity and excellent global optimization capability, PSO has been successfully applied to a number of applications, such as DNA sequence compression [7], resource allocation [8] and water distribution network design [9]. In this paper, we mainly focus on the two versions of standard PSO algorithms [10]: a constricted GBest PSO algorithm using a global topology and a constricted LBest PSO algorithm using a local topology.

In the gbest topology, often referred to as a global gbest model, every particle is able to obtain information from the best particle in the entire swarm population. The local lbest topology, often referred to as an lbest model, connects each particle to only its neighbouring particles in the swarm. Figure 1 (a) shows the full connections with four particles in

a) E-mail: shiqin.yang.jk@stu.hosei.ac.jp
DOI: 10.1587/transinf.2016EDP7219

Copyright © 2017 The Institute of Electronics, Information and Communication Engineers


In population-based algorithms, finding the optimal solution of a problem is based on the right balance between exploration and exploitation of the search space. The premise of SHPSO is that a better balance of exploration and exploitation can be achieved by having particles that follow different search behaviours, which should result in more accurate solutions. However, empirical study shows that the nature of different search behaviours between particles is not the only factor that determines the performance of SHPSO [2]. The relative composition of the swarm plays a major role in this respect. In SHPSO, the proportions of particles of different kinds are assigned during initialization, moreover, these assignments do not change during the optimization process. As a consequence of this, the inappropriate composition of the swarm will lead the particles near the local minimum to be trapped in the local optima. In this case, escaping from the local optima becomes difficult and SHPSO suffers from the premature convergence problem.

This motivated our attempt to improve the performance of SHPSO by introducing adaptive particle swarms as a response to some events caused by the behaviour of the swarm, thus “guiding” an appropriate balance between exploration and exploitation in the search space. The dynamic self-adaptive heterogeneity is able to automatically alter its heterogeneity type by the triggering configuration. The proposed triggering configuration keeps track of the frequency of the unchanged best position of the entire swarm (\(p_g\)) for a number of iterations. This information is then used to select a new heterogeneous structure when \(p_g\) is considered stagnant. According to different types of heterogeneity, we propose two different DHPSO models, namely, DHPSO-d (\(-d\) represents different rules) and DHPSO-p (\(-p\) means polymorphic models).

### 3.2 Proposal of DHPSO-d

Conceptually, we say that DHPSO-d exhibits dynamic update-rule heterogeneity. In DHPSO-d, different particles dynamically use different rules for updating their position by the triggering configuration in the search space. There are three rules (rule A, rule B and rule C) used in DHPSO-d. To rule A, the velocity and position of each particle are updated by the following equations [10]:

\[
\begin{align*}
    v_i &= \chi(v_i + c_1 e_1(p_i - x_i) + c_2 e_2(p_g - x_i)) \quad (1) \\
    x_i &= x_i + v_i \quad (2)
\end{align*}
\]

In Eq. (1), \(\chi\) is a constriction factor, \(e_1\) and \(e_2\) are independent random numbers uniquely generated at every update for each individual dimension in the range [0, 1] according to references [10], [15]. In order to ensure convergence, the values \(\chi \approx 0.72984\) and \(c_1 = c_2 = 2.05\) are preferred in most cases. \(p_i\) is the best position of the particle \(i\), \(x_i\) is the current position of the particle \(i\) and \(p_g\) is the best position of the entire swarm. To rule B, \(p_g\) can be reset by:

\[
    p_g = \frac{1}{n} \sum_{i=1}^{n} p_i \quad (3)
\]

To rule C, all particles in swarm update their positions by the following equations:

\[
\begin{align*}
    v_i &= \psi_1 v_i + \psi_2(p_i - x_i) \quad (4) \\
    x_i &= x_i + v_i \quad (5)
\end{align*}
\]

In Eq. (4) \(\psi_1\) and \(\psi_2\) are random numbers uniquely generated at every update for each individual dimension in the range [0, 1].
Algorithm 1 Main function of DHPSO-d with different rules

1: Initialize: population n, threshold $e_0$, $c_1 = c_2 = 2.05, \chi = 0.7298$
2: Initialize each $x_i$: $\text{rand}(x_i) \in (x_{\min}, x_{\max}), i \in [1,n]$
3: Initialize each $p_i$ and the global best position $p_g$, flag = 0
4: Initialize: $\text{rand}(\psi_i) \in [0,1], \text{rand}(\psi_2) \in [0,1], \text{rand}(\psi) \in (x_{\min}, x_{\max})$
5: Repeat
6: for all particles do
7: Update particle $x_i$ with rule $A$
8: Calculate fitness value of the updated particle $f(x'_i)$
9: if the new position $x'_i$ is better than $p_i$ then
10: $p_i = x'_i, f(p_i) = f(x'_i)$
11: end if
12: if the new position $x'_i$ is better than $p_g$ then
13: $p_g = x'_i, f(p_g) = f(x'_i), \text{flag} = 0$
14: end if
15: end for
16: Count the number of unchanged $p_y$: flag = flag + 1
17: if flag $\geq e_0$ then
18: Reset $p_y$ as mean of all particle’s $p_i$ with rule $B$
19: Re-initialize each $p_i$: $\text{rand}(p_i) \in (x_{\min}, x_{\max})$
20: Update each particle $x_i$ with rule $C$, flag = 0
21: end if
22: Until maximum iterations are attained

Figure 2 demonstrates the dynamic heterogeneity configuration in DHPSO-d during the optimization process, where $t$ is the current iteration number. To DHPSO-d, at each iteration, all particles use rule $A$ for updating their positions in the search space. flag is a counter to record the number of iterations of unchanged $p_y$. When the number of iterations is equal or greater than a threshold $e_0$, triggering event $a$ is confirmed ($\text{flag} \geq e_0$), then $p_y$ is reset by rule $B$ and all particles update their positions by rule $C$. After that, all particles will be passed to the next iteration. The pseudocode of main function of DHPSO-d can be summarized in Algorithm 1. For each iteration, the number of Fitness Evaluations (FEs) is $(n \cdot D)$, where $n$ is the population size and $D$ is the dimensionality of the problem. The computational complexity of DHPSO-d is $O(n \cdot D \cdot T)$, where $T$ is the maximum iteration number.

3.3 Proposal of DHPSO-p

Conceptually, DHPSO-p is a dynamic heterogeneous structure combined with the $gbest$ topology and the pairwise connection topology. Figure 3 (a) shows full connections with four particles in the $gbest$ topology, which is the topology type used in model $\Upsilon$. Without loss of generality, the population of DHPSO-p is denoted as an even number $n$. In Fig. 3 (b), each particle only has one specified neighbourhood, which is the topology type used in model $\Omega$. In model $\Omega$, neither $p_i$ nor the $p_g$ is involved in updating the particles. Instead, all particles are divided into two teams: an elite team and a loser team. Each particle of the loser team will be randomly connected with a particle from the elite team. The pseudocode of main function of DHPSO-p is summarized in Algorithm 2.

Algorithm 2 Main function of DHPSO-p with polymorphic models

1: Initialize population $n$, threshold $e_0$, $c_1 = c_2 = 2.05, \chi = 0.7298$
2: Initialize each $x_i$: $\text{rand}(x_i) \in (x_{\min}, x_{\max}), i \in [1,n]$
3: Initialize each $p_i$ and global best position $p_g$, flag = 0, trap = 0
4: Initialize: $\text{rand}(\psi_i) \in [0,1], \text{rand}(\psi_2) \in [0,1], \text{rand}(\psi) \in (x_{\min}, x_{\max})$
5: Repeat
6: for all particles (Use model $\Upsilon$) do
7: Update particle $x_i$ with rule $A$
8: Calculate fitness value of each updated particle $f(x'_i)$
9: if the new position $x'_i$ is better than $p_i$ then
10: $p_i = x'_i, f(p_i) = f(x'_i)$
11: end if
12: if the new position $x'_i$ is better than $p_g$ then
13: $p_g = x'_i, f(p_g) = f(x'_i), \text{flag} = 0$
14: end if
15: end for
16: Count the number of unchanged $p_y$: flag = flag + 1
17: if flag $\geq e_0$ then
18: Reset $p_y$ as mean of all particle’s $p_i$ with rule $B$
19: Re-initialize each $p_i$: $\text{rand}(p_i) \in (x_{\min}, x_{\max})$
20: Update each particle $x_i$ with rule $C$, flag = 0
21: Count the number of trapped into local optimum: trap = trap + 1
22: end if
23: if trap $\geq e_1$ then
24: Switch to Model $\Omega$
25: Break
26: end if
27: Until maximum iterations are attained

Algorithm 3 demonstrates the pseudocode of subfunction of DHPSO-p in model $\Omega$. In model $\Omega$, the fitness values of $x_i$ are sorted by the ascending order and afterwards, the particles that correspond to the top half of the sequence are selected as an elite team. All particles in the elite
team will be passed to the next iteration, while the loser will update its position and velocity by learning from the randomly selected elite. After learning from the elite by rule $D$, the loser, will also be passed to the next iteration. The main FEs in Algorithm 3 is to calculate the updated loser particle $x_i^*$. For each iteration, the number of FEs is $0.5 \cdot n \cdot D$. The computational complexity of model $\Omega$ is $O(n \cdot D \cdot T)$. Combination of model $I'$ and model $\Omega$ shown in Algorithm 2, the computational complexity of DHPPO-p is $O(n \cdot D \cdot T)$.

**Algorithm 3** Sub-function of DHPPO-p in model $\Omega$

1: Re-initialize $p_i$ of all particles, all particle’s positions $x_i$ and $p_g$
2: Repeat
3: for all particles do
4: Sort the $x_i$ by the ascending order
5: Select the top half of $x_i$ as an elite team
6: The remaining of particles are assigned to the loser team
7: Randomly select an elite and a loser, update the loser with rule $D$
8: end for
9: for all particles do
10: if the new position $x_i^*$ is better than $p_i$ then
11: $p_i = x_i^*$
12: end if
13: end for
14: Update the global optimum: $p_g = argmin(p_i)$
15: Until maximum iterations are attained

To rule $D$, all loser particles update their positions by the following equations:

$$v_i = \psi_1 v_i + \psi_2 (x_k - x_i)$$

(6)

$$x_i^* = x_i + v_i$$

(7)

Where, $x_k$ is the position of a randomly selected elite, $x_i$ is the position of a randomly selected loser and $x_i^*$ is the new updated position of the loser $x_i$.

$\psi_1$ and $\psi_2$ are random numbers uniquely generated at every update for each individual dimension in the range $[0, 1]$.

4. **Experiments**

4.1 **Evaluation Method**

The experiments are implemented on a PC with an Intel Core i7 860, 2.8 GHz CPU and Microsoft Windows 7 Professional SP1 64-bit operating system, and all algorithms are written in Matlab language. The bounds and global minimums of ten benchmark functions are shown in Table 1. These benchmarks were chosen for their variety. Functions $f_1 - f_6$ are highly complex multimodal problems with many local minima, $f_7 - f_{10}$ are continuous, convex and unimodal test problems. The four state-of-the-art algorithms are also chosen for their variety. First of all, the proposed DHPPO-d and DHPPO-p are based on GBestPSO [10]. So the GBestPSO algorithm is used to compare the performance of our proposed algorithms. Secondly, LBBestPSO [10] and Firefly Algorithm (FA) [16] are well-known algorithms for solving the highly complex multimodal problems. The global searching capability of DHPPO-d and DHPPO-p for multimodal problems can be confirmed by empirical comparison with LBBestPSO and FA. The last comparative algorithm, HGLPSO, is a type of SHPSO, where particles will be allocated different search behaviours by selecting different topologies from GBestPSO and LBBestPSO. The control parameters of all algorithms are shown in Table 2.

| Table 1 | The bounds and global minimums of benchmark functions. |
|---------|--------------------------------------------------------|
| Equation | Bounds | Optimum |
| Rosenbrock $f_1(x) = \sum_{j=1}^{D} \left( 100 \cdot (x_{j+1} - x_j^2) + (x_j - 1)^2 \right)$ | $(-100, 100)$ | $f(x^*) = 0, x^* = 1.0^D$ |
| Rastrigin $f_2(x) = \sum_{j=1}^{D} \left( x_j^2 - 10 \cdot \cos(2\pi x_j) \right) + 10 \cdot D$ | $(-5.12, 5.12)$ | $f(x^*) = 0, x^* = 0.0^D$ |
| Griewank $f_3(x) = \frac{1}{4000} \sum_{j=1}^{D} x_j^2 - \prod_{j=1}^{D} \cos \left( \frac{x_j}{\sqrt{j}} \right) + 1$ | $(-5, 10)$ | $f(x^*) = 0, x^* = 0.0^D$ |
| Ackley $f_4(x) = -a \cdot \exp(-0.02 \cdot \left( \sum_{j=1}^{D} x_j^2 \right)^{1/2})$ | $(-32, 32)$ | $f(x^*) = 0, x^* = 0.0^D$ |
| $+ \exp(D - \sum_{j=1}^{D} \cos(2\pi x_j)) + a \cdot \exp(a) = 20$ | | |
| Levy $f_5(x) = \sum_{j=1}^{D} \left( \left[ \sum_{j=1}^{D} \left( \sum_{j=1}^{D} \left( (1 + 10 \cdot \sin(x_j)) \right)^{-\alpha} \right) \right] + \sum_{j=1}^{D} \left( \sum_{j=1}^{D} \left( \sum_{j=1}^{D} \left( \sin(x_j) \right)^{-\alpha} \right) \right) \right)$ | $(-10, 10)$ | $f(x^*) = 0, x^* = 1.0^D$ |
| $+(\pi - 1)^2 \cdot \left( 1 + \sin^2(\pi \omega D) \right)$, $\omega = 1 + \frac{x_j}{\sqrt{D}}$ | | |
| Schwefel $f_6(x) = 418.9829 \cdot D - \sum_{j=1}^{D} j \sin(\sqrt{|x_j|})$ | $(-500, 500)$ | $f(x^*) = 0, x^* = 420.9687^D$ |
| Sphere $f_7(x) = \sum_{j=1}^{D} x_j^2$ | $(-5.12, 5.12)$ | $f(x^*) = 0, x^* = 0.0^D$ |
| Rotated hyper-ellipsoid $f_8(x) = \sum_{j=1}^{D} \sum_{j=1}^{D} x_j^2$ | $(-65, 65, 65)$ | $f(x^*) = 0, x^* = 0.0^D$ |
| Sum of different powers $f_9(x) = \sum_{j=1}^{D} |x_j|^{1.1}$ | $(-1, 1)$ | $f(x^*) = 0, x^* = 0.0^D$ |
| Zakharov $f_{10}(x) = \sum_{j=1}^{D} x_j^2 + (\sum_{j=1}^{D} 0.5 j x_j)^2 + (\sum_{j=1}^{D} 0.5 j x_j)^3$ | $(-1, 1)$ | $f(x^*) = 0, x^* = 0.0^D$ |
4.2 Evaluation Results

The collecting experimental results are including the best optimum solution for the test functions, averaged optimum solution and running time of each algorithm after 30 times independent trials. These experimental results are displayed in Table 3 and Table 4.

4.2.1 Analysis of Results on Multimodal Functions $f_1 - f_6$

Table 5 shows the results of pairwise comparisons between DHPSO-d and four compared algorithms by $t$-test. In Table 5, the symbol (+) represents the performance ob-

Table 2 The control parameters setting of algorithms.

| Algorithm | Control Parameters |
|-----------|--------------------|
| GBestPSO  | $c_1 = c_2 = 2.05, \epsilon = 0.7298$ |
| LBBestPSO | $c_1 = c_2 = 2.05, \epsilon = 0.7298$ |
| FA        | $\alpha = 0.25, y = 1, \beta_{min} = 0.20, \beta_0 = 1$ |
| HGLPSO    | $c_1 = c_2 = 2.05, \epsilon = 0.7298$ |
| DHPSO-d   | $c_1 = c_2 = 2.05, \epsilon = 0.7298, \epsilon_1 = 80$ |
| DHPSO-p   | $c_1 = c_2 = 2.05, \epsilon = 0.7298, \epsilon_1 = 80, \epsilon_2 = 2$ |

Table 3 The statistical results of optimization errors on multimodal functions: $f_1 - f_6$

| Algorithm | 50-D(100p) | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ |
|-----------|------------|-------|-------|-------|-------|-------|-------|
| GBestPSO  | Best       | 2.07e+01 | 1.30e+02 | 9.86e-03 | 0 | 4.90e+00 | 4.35e+03 |
|           | Mean       | 1.16e+05 | 2.25e+02 | 7.44e-01 | 4.77e+00 | 2.65e+01 | 6.48e+03 |
|           | Time(s)    | 1.59e+02 | 1.53e+02 | 3.71e+02 | 1.54e+02 | 2.33e+02 | 1.86e+02 |
| LBBestPSO | Best       | 1.14e+02 | 1.05e+02 | 0 | 0 | 4.60e-01 | 7.00e+03 |
|           | Mean       | 1.81e+02 | 1.90e+02 | 0 | 0 | 4.83e+02 | 4.77e+02 |
|           | Time(s)    | 3.82e+02 | 3.97e+02 | 4.26e+02 | 3.64e+02 | 4.83e+02 | 4.77e+02 |
| FA        | Best       | 4.70e+01 | 4.97e+00 | 0 | 1.61e-06 | 0 | 7.82e+03 |
|           | Mean       | 8.96e+02 | 3.90e+01 | 5.21e-07 | 4.70e-03 | 7.56e-06 | 9.07e+03 |
|           | Time(s)    | 9.34e+03 | 2.88e+05 | 9.73e+03 | 9.40e+03 | 6.02e+04 | 9.41e+03 |
| HGLPSO    | Best       | 3.07e-04 | 1.61e+02 | 0 | 0 | 5.37e+01 | 0 | 5.12e+03 |
|           | Mean       | 2.76e+01 | 2.13e+02 | 2.47e-04 | 3.66e+01 | 5.73e+00 | 6.69e+03 |
|           | Time(s)    | 1.99e+02 | 2.62e+02 | 2.99e+02 | 8.50e+01 | 3.41e+02 | 3.13e+02 |
| DHPSO-d   | Best       | 2.31e+01 | 5.37e+01 | 0 | 0 | 0 | 0 | 2.01e+03 |
|           | Mean       | 8.86e+04 | 1.35e+02 | 8.51e-01 | 0 | 0 | 0 | 3.92e+03 |
|           | Time(s)    | 2.05e+02 | 1.78e+02 | 1.43e-01 | 8.51e-01 | 0 | 0 | 1.96e+02 |
| DHPSO-p   | Best       | 1.05e+02 | 8.95e+00 | 0 | 7.38e-07 | 0 | 2.57e+03 |
|           | Mean       | 2.01e+03 | 1.77e+01 | 4.35e-06 | 2.86e-01 | 2.44e-01 | 3.93e+03 |
|           | Time(s)    | 2.50e+03 | 3.04e+02 | 3.41e+02 | 3.04e+02 | 3.71e+02 | 3.25e+02 |

| Algorithm | 100-D(200p) | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ |
|-----------|------------|-------|-------|-------|-------|-------|-------|
| GBestPSO  | Best       | 4.56e+01 | 4.30e+02 | 5.79e-01 | 3.72e+00 | 1.13e+00 | 1.15e+04 |
|           | Mean       | 2.00e+09 | 5.82e+02 | 1.00e+00 | 1.38e+01 | 1.20e+02 | 1.50e+04 |
|           | Time(s)    | 4.08e+02 | 4.42e+02 | 5.07e+02 | 4.71e+02 | 5.77e+02 | 4.96e+02 |
| LBBestPSO | Best       | 4.03e+02 | 4.55e+02 | 0 | 1.21e+02 | 3.18e+01 | 1.41e+04 |
|           | Mean       | 5.63e+02 | 5.21e+02 | 0 | 2.41e+02 | 5.85e+01 | 1.38e+04 |
|           | Time(s)    | 8.41e+02 | 9.32e+02 | 8.98e+02 | 8.79e+02 | 1.20e+03 | 1.14e+03 |
| FA        | Best       | 1.01e+02 | 4.88e+01 | 1.69e-07 | 3.20e-03 | 4.98e-05 | 2.01e+04 |
|           | Mean       | 8.96e+02 | 9.05e+01 | 4.36e-06 | 1.16e-02 | 9.30e-05 | 2.16e+04 |
|           | Time(s)    | 4.48e+04 | 4.23e+05 | 5.22e+05 | 6.29e+04 | 3.67e+05 | 1.39e+05 |
| HGLPSO    | Best       | 3.85e+01 | 4.46e+02 | 0 | 1.27e+00 | 4.32e+01 | 1.29e+04 |
|           | Mean       | 1.49e+02 | 5.39e+02 | 0 | 2.63e+00 | 6.49e+01 | 1.53e+04 |
|           | Time(s)    | 5.90e+02 | 8.22e+02 | 8.60e+02 | 8.12e+02 | 9.28e+02 | 9.57e+02 |
| DHPSO-d   | Best       | 7.34e+01 | 3.89e+02 | 5.09e-01 | 0 | 2.14e+01 | 7.74e+03 |
|           | Mean       | 1.66e+09 | 4.68e+02 | 8.96e-01 | 1.20e+01 | 6.04e+01 | 1.10e+04 |
|           | Time(s)    | 4.82e+02 | 4.92e+02 | 5.59e+02 | 5.29e+02 | 6.46e+02 | 5.09e+02 |
| DHPSO-p   | Best       | 1.71e+02 | 1.39e+01 | 0 | 8.79e-03 | 1.59e-01 | 7.26e-03 |
|           | Mean       | 4.10e+02 | 2.38e+01 | 0 | 8.79e-03 | 1.59e-01 | 7.26e-03 |
|           | Time(s)    | 5.71e+02 | 6.75e+02 | 7.25e+02 | 6.23e+02 | 8.98e+02 | 7.42e+02 |
Table 4 The statistical results of optimization errors on 50-Dimension, 100-Dimension unimodal functions with a population of 100 and 200 after 30 trials of $1 \times 10^4$ function evaluations.

| Dim | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ |
|-----|-------|-------|-------|---------|
| 50-D | 5.24e+00 | 6.10e+04 | 0 | 2.55e+01 |
| 1.25e+02 | 3.46e+02 | 3.32e+02 | 1.61e+02 | 4.03e-05 |
| 1.80e+02 | 7.98e+02 | 9.10e+02 | 3.22e+02 | 2.55e+00 |
| FA | 6.57e-07 | 4.79e-02 | 0 | 0 |
| 4.21e-06 | 1.15e-01 | 1.15e-06 | 0 |
| 9.50e+03 | 1.66e+04 | 2.73e+03 | 1.04e+04 | 0 |
| HGLPSO | 0 | 0 | 0 | 1.27e+01 |
| 1.50e+02 | 5.82e+02 | 6.43e+02 | 2.13e+02 |
| DHPSO-d | 0 | 0 | 0 | 1.50e+02 |
| 1.45e+02 | 4.01e+02 | 4.03e+02 | 2.23e+02 |
| 0 | 0 | 0 | 1.17e-04 |
| 2.22e-01 | 0 | 2.15e-04 |
| 1.99e+02 | 4.93e+02 | 3.54e+02 | 2.00e+02 |
| 100-D | 0 | 0 | 0 | 1.29e+00 |
| 4.54e+01 | 1.49e+03 | 8.81e+02 | 4.00e+02 |
| LBestPSO | 0 | 0 | 0 | 1.29e+00 |
| 4.72e+02 | 3.85e+03 | 2.23e+03 | 6.31e+02 |
| FA | 3.23e+05 | 4.41e+05 | 0 | 4.33e+05 |
| 3.70e+05 | 6.26e+04 | 3.37e+03 | 6.78e+04 |
| HGLPSO | 0 | 0 | 0 | 2.76e+01 |
| 4.22e+02 | 1.18e+03 | 2.67e+03 | 5.39e+02 |
| DHPSO-d | 8.74e+00 | 2.03e+05 | 0 | 9.85e+02 |
| 3.82e+02 | 1.87e+03 | 1.08e+03 | 5.55e+02 |
| DHPSO-p | 0 | 0 | 0 | 1.29e-04 |
| 4.42e+02 | 2.22e+03 | 9.42e+02 | 5.45e+02 |

Table 5 Performance comparison between DHPSO-d and four comparative algorithms on $f_7-f_{10}$ by $t$-test with a significance level of $\alpha = 0.05$.

| Dim | DHPSO-d VS | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ |
|-----|------------|-------|-------|-------|---------|
| 50-D | VS–GBestPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 3/3ori | VS–LBestPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 2/2ori | VS–FA | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 1/2ori | VS–HGLPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 3/1ori | DHPSO-d–HGBestPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 2/1ori | DHPSO-d–LBestPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/2ori | DHPSO-d–FA | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/2ori | DHPSO-d–HGLPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–FA | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–LBestPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–HGLPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–FA | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–LBestPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–HGLPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–FA | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–LBestPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–HGLPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–FA | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–LBestPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–HGLPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–FA | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–LBestPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–HGLPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–FA | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–LBestPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–HGLPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–FA | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–LBestPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–HGLPSO | $\approx$ | $\approx$ | $\approx$ | $\approx$ |
| 0/3ori | DHPSO-d–FA | $\approx$ | $\approx$ | $\approx$ | $\approx$ |

4.2.2 Analysis of Results on Unimodal Functions $f_7 - f_{10}$

As described comparisons in Table 7, DHPSO-d performs better than GBestPSO on all unimodal functions with dimensions of 50 and 100. Compared with LBestPSO, FA and HGLPSO, the performance of DHPSO-d is poorer than them. It reveals that DHPSO-d could not perform the superior performance for unimodal problems.

In Table 8, only to LBestPSO, the performance of DHPSO-p is poorer than the compared algorithm on $f_7$ function with the dimensions of 50. Specifically, when the dimensionality of the problem is increased to 100, DHPSO-p successfully demonstrates the superior scalability over all compared algorithms.

4.2.3 Analysis of Results on All Functions $f_1 - f_{10}$

In this part, the scalability analysis is performed in the comparison between DHPSO models (DHPSO-d and DHPSO-p) and SHPSO model (HGLPSO) to all test functions $f_1 - f_{10}$. There is a problem in conducting multiple significance tests. Because they are probabilistic, it is possible that some results are easy to chance, even random data generated from the same distribution will differ significantly sometimes. This problem can be addressed by a modified Bonferroni procedure which manipulates the $\alpha$ value in a way that protects against capitalization on chance [17]. It should be noted that the test function $f_9$ is excluded from the modified Bonferroni procedure in Table 9 and Table 10, because all six algorithms could find the optimum solution.
Table 9 A modified Bonferroni procedure for DHPSO-d VS HGLPSO.

| 50-D | p-value | Rank | Rank' | New α | Significant |
|------|---------|------|-------|-------|-------------|
| f6  | 2.89e-19 | 1    | 9     | 5.56e-03 | Yes (+) |
| f2  | 6.07e-11 | 2    | 8     | 6.25e-03 | Yes (+) |
| f1  | 9.35e-09 | 3    | 7     | 7.14e-03 | Yes (−) |
| f8  | 4.24e-04 | 4    | 6     | 8.33e-03 | Yes (−) |
| f3  | 5.08e-04 | 5    | 5     | 1.00e-02 | Yes (−) |
| f5  | 5.51e-04 | 6    | 4     | 1.25e-02 | Yes (+) |
| f7  | 2.00e-01 | 7    | 3     | 1.67e-02 | No (−) |
| f4  | 4.34e-01 | 8    | 2     | 2.50e-02 | No (−) |
| f10 | 6.81e-01 | 9    | 1     | 5.00e-02 | No (−) |

+/−+/− 3/3/3

| 100-D | p-value | Rank | Rank' | New α | Significant |
|-------|---------|------|-------|-------|-------------|
| f3   | 3.97e-24 | 1    | 9     | 5.56e-03 | Yes (−) |
| f6   | 3.32e-13 | 2    | 8     | 6.25e-03 | Yes (+) |
| f4   | 2.93e-12 | 3    | 7     | 7.14e-03 | Yes (−) |
| f8   | 6.37e-09 | 4    | 6     | 8.33e-03 | Yes (−) |
| f2   | 1.68e-08 | 5    | 5     | 1.00e-02 | Yes (+) |
| f10  | 8.95e-04 | 6    | 4     | 1.25e-02 | Yes (−) |
| f7   | 2.31e-03 | 7    | 3     | 1.67e-02 | No (−) |
| f1   | 2.26e-02 | 8    | 2     | 2.50e-02 | No (−) |
| f5   | 4.14e-01 | 9    | 1     | 5.00e-02 | No (−) |

+/−+/− 2/1/6

Table 10 Modified Bonferroni procedure for DHPSO-p VS HGLPSO.

| 50-D | p-value | Rank | Rank' | New α | Significant |
|------|---------|------|-------|-------|-------------|
| f2   | 6.18e-25 | 1    | 9     | 5.56e-03 | Yes (+) |
| f6   | 2.49e-14 | 2    | 8     | 6.25e-03 | Yes (+) |
| f5   | 8.67e-04 | 3    | 7     | 7.14e-03 | Yes (−) |
| f10  | 2.48e-02 | 4    | 6     | 8.33e-03 | No (−) |
| f8   | 2.80e-02 | 5    | 5     | 1.00e-02 | No (−) |
| f1   | 1.05e-02 | 6    | 4     | 1.25e-02 | No (−) |
| f7   | 2.61e-01 | 7    | 3     | 1.67e-02 | No (−) |
| f3   | 3.34e-01 | 8    | 2     | 2.50e-02 | No (−) |
| f4   | 7.64e-01 | 9    | 1     | 5.00e-02 | No (−) |

+/−+/− 3/6/0

| 100-D | p-value | Rank | Rank' | New α | Significant |
|-------|---------|------|-------|-------|-------------|
| f2   | 1.20e-34 | 1    | 9     | 5.56e-03 | Yes (+) |
| f6   | 6.38e-31 | 2    | 8     | 6.25e-03 | Yes (+) |
| f5   | 1.32e-24 | 3    | 7     | 7.14e-03 | Yes (+) |
| f10  | 5.72e-05 | 4    | 6     | 8.33e-03 | Yes (−) |
| f1   | 8.78e-05 | 5    | 5     | 1.00e-02 | Yes (−) |
| f10  | 4.28e-02 | 6    | 4     | 1.25e-02 | No (−) |
| f3   | 1.68e-01 | 7    | 3     | 1.67e-02 | No (−) |
| f7   | 3.24e-01 | 8    | 2     | 2.50e-02 | No (−) |
| f8   | 6.81e-01 | 9    | 1     | 5.00e-02 | No (−) |

+/−+/− 4/4/1

and have the same performances on f0.

In Table 9, on the dimensions of 50, the number of (3/3/3) shown in the last column means that there is no significantly difference in performance between DHPSO-d and HGLPSO to all test functions. On the high dimensions of 100, 2/1/6 means that the performance of DHPSO-d is poorer than HGLPSO.

In Table 10, on the dimensions of 50, 3/6/0 means that DHPSO-p is significantly superior on 3 test functions. On the dimensions of 100, 4/4/1 means that DHPSO-p is significantly superior on 4 functions, while HGLPSO is only significantly superior on 1 function (f1). It is clear that in many of the test cases, DHPSO-p has better performance than the HGLPSO, especially in the high dimensional cases.

4.3 Discussion

Based on the previous empirical analysis, the scalability of proposed DHPSO-d and DHPSO-p will be examined from a practical point of view, namely, by evaluating the optimum performance and running time on each algorithm. In general, if there is no significantly difference in optimal solution, we will prefer the algorithm with less running time. Following this viewpoint, an overall picture of the scalability of DHPSO-d and DHPSO-p is depicted as follows.

To the multimodal functions: f1 - f6

From Fig. 4 it can be observed that the computation time of DHPSO-d is less than the most of algorithms except GBestPSO. As shown in Fig. 4 and Fig. 5, the computation time of DHPSO-p is less than LBestPSO and FA. To summarize, on 50 dimensional functions, DHPSO-d exhibits the superior solution accuracy and faster convergence speed over two homogeneous PSO algorithms (GBestPSO and LBestPSO) and one SHPSO algorithm (HGLPSO). This implies that the method of different particles dynamically using different rules for updating their positions is beneficial to the solution improvement during the optimization process. However, DHPSO-d did show some deterioration in performance for 100 dimensions on test functions, which illustrates that the stagnation of the best-so-far solution couldn’t be eliminated only by the dynamic update-rule heterogeneity. On the other hand, we note that DHPSO-
p has the best scalability over all five algorithms on multimodal functions, even in the high dimensional cases. It reveals that the strategy of dynamic model-of-influence heterogeneity in DHPSO-p could successfully improve the performance of DHPSO-d for multimodal function, especially in the high dimensional cases.

- To the unimodal functions: $f_7 - f_{10}$

Table 7 reveals that DHPSO-d could not demonstrate the desirable performance for the unimodal functions. Although dynamic update-rule heterogeneity is efficient for searching the global optimum of multimodal problems, tuning the algorithm to find the optimal solution on unimodal problem is ineffective. Different with DHPSO-d, DHPSO-p exhibits the superior scalability over all algorithms except LBestPSO. These results point towards the idea that mechanisms allowing the swarm or sub-swarms to choose their informers during the optimization process could be more beneficial to address the exploration and exploitation trade-off problem than designing sophisticated update rules to make particles capable of doing both things. This means that dynamic heterogeneous structure enables us to move design complications from the individual level to the swarm level. The results show that DHPSO-p has the best scalability over all five algorithms as the dimensionality of unimodal problem increased to 100.

- To all functions: $f_1 - f_{10}$

According to Fig. 4 and Fig. 6, it can be seen that the computation time of DHPSO-d is less than HGLPSO, and there is no significantly difference in computation time between DHPSO-p and HGLPSO. In terms of solution quality and computation time, DHPSO-d outperforms HGLPSO on 50 dimensional functions, while HGLPSO has better scalability than DHPSO-d on the high dimensions of 100. According to the previous analysis of DHPSO-d, the stagnation of optimal solution can’t be eliminated only by the dynamic update-rule heterogeneity as the dimensionality of the problem is increased to 100. Another result of the experimental analysis of DHPSO-d is that the strategy of dynamic model-of-influence heterogeneity is more beneficial to solution improvement than designing sophisticated update rules. We note that HGLPSO is a static model-of-influence heterogeneity combined with the $g_{best}$ model and the $l_{best}$ model.

In conclusion, the results presented in this study have shown that the model-of-influence heterogeneity algorithm is more scalable to higher dimensions compared to the update-rule heterogeneity algorithm. In addition, DHPSO-p has better scalability than HGLPSO on all functions with dimensions of 50 and 100. Our results show that the dynamic self-adaptive heterogeneity is able to address the exploration-exploitation trade-off problem, and provide the excellent optimal solution of a problem.

5. Conclusion

In this paper, we propose two DHPSO models, namely, DHPSO-d with dynamic update-rule heterogeneity and DHPSO-p with dynamic model-of-influence heterogeneity. In order to systematically investigate the performances of DHPSO-d and DHPSO-p, four state-of-the-art algorithms are performed in comparison based on 10 classical unconstrained problems. One of the results of DHPSO-d analysis is that the method of dynamic update-rule heterogeneity is beneficial to the solution improvement during the optimization process. Another result analysis of DHPSO-d shows that the stagnation of optimal solution can’t be eliminated only by the dynamic update-rule heterogeneity as the dimensionality of the problem increases. Moreover, empirical study of DHPSO-p shows the adaptive model-of-influence heterogeneity algorithm is more scalable to higher dimensions than dynamic update-rule heterogeneity algorithm. In conclusion, the dynamic self-adaptive heterogeneous structure is able to effectively address the exploration-exploitation trade-off problem, and provides the excellent optimal solution of a problem. The enlarged adaptive behaviours of particle swarms may lead to a more efficient DHPSO model and our future work is focusing on this issue.

References

[1] R.C. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” Proc. 6th International Symposium on Micro Machine and Human Science, pp.39–43, 1995.
[2] M.A.M. De Oca, J. Peña, T. Stützle, C. Pinciroli, and M. Dorigo, “Heterogeneous particle swarm optimizers,” Proc. IEEE Congr. Evolutionary Computation, IEEE, pp.698–705, 2009.
[3] A.P. Engelbrecht, “Heterogeneous particle swarm optimization,” Swarm Intelligence, vol.6234, pp.191–202, Springer, 2010.
[4] N. Lynn and P.N. Suganthan, “Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation,” Swarm and Evolutionary Computation, vol.24, pp.11–24, 2015.
[5] X.S. Yang and X. He, “Firefly algorithm: recent advances and applications,” International Journal of Swarm Intelligence, vol.1, no.1, pp.36–50, 2013.
[6] I. Fister, I. Fister Jr., X.S. Yang, and J. Brest, “A comprehensive review of firefly algorithms,” Swarm and Evolutionary Computation, vol.13, pp.34–46, 2013.
[7] Z. Zhu, J. Zhou, Z. Ji, and Y.-H. Shi, “Dna sequence compression using adaptive particle swarm optimization-based memetic algorithm,” Evolutionary Computation, IEEE Transactions on, vol.15, no.5, pp.643–658, 2011.
[8] Y.-J. Gong, J. Zhang, H.S.-H. Chung, W.-N. Chen, Z.-H. Zhan, Y. Li,
and Y.-H. Shi, “An efficient resource allocation scheme using particle swarm optimization,” Evolutionary Computation, IEEE Transactions on, vol.16, no.6, pp.801–816, 2012.

[9] J. Montalvo, J. Izquierdo, R. Pérez, and P.L. Iglesias, “A diversity-enriched variant of discrete pso applied to the design of water distribution networks,” Engineering Optimization, vol.40, no.7, pp.655–668, 2008.

[10] D. Bratton and J. Kennedy, “Defining a standard for particle swarm optimization,” Swarm Intelligence Symposium, IEEE, pp.120–127, 2007.

[11] J. Kennedy, “The behavior of particles,” Evolutionary Programming VII, Springer, vol.1447, pp.579–589, 1998.

[12] J. Kennedy, J.F. Kennedy, and R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann, 2001.

[13] J. Kennedy, “Small worlds and mega minds: effects of neighborhood topology on particle swarm performance,” Proc. IEEE Congr. Evolutionary Computation, IEEE, pp.1931–1938, 1999.

[14] J. Kennedy and R. Mendes, “Population structure and particle swarm performance,” Proc. IEEE Congr. Evolutionary Computation, IEEE, pp.1671–1676, 2002.

[15] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” IEEE Congress on Evolutionary Computation, IEEE, pp.69–73, 1998.

[16] X.S. Yang, Nature-inspired metaheuristic algorithms, Luniver Press, 2010.

[17] C. Hommel, “A comparison of two modified bonferroni procedures,” Biometrika, vol.76, no.3, pp.624–625, 1989.

Shiqin Yang received the M.Eng degree from JiangNan University, Wuxi, China, in 2008. She is currently pursuing the Ph.D. degree with the Graduate School of Computer and Information Sciences, Hosei University, Tokyo, Japan. Her research interests include swarm intelligence, continuous dynamic constrained optimization and fuzzy clustering algorithms.

Yuji Sato received the B.E and Ph.D. degrees in engineering from the University of Tokyo, Japan in 1981 and 1997, respectively. He is a professor of Faculty of Computer and Information Sciences at Hosei University, Japan. From 2007 to 2008, he was a visiting scholar at Illinois Genetic Algorithms Laboratory (IlliGAL). His research interests include evolutionary computation on many-core architecture, swarm intelligence for multimodal problems and evolution of machine learning techniques in design. He has won the 2015 highly commended paper award in International Journal of Intelligent Computing and Cybernetics. He co-organized a special session EC on Many-core Architecture to Solve Large-scale Problems in CEC2011 at New Orleans, Computational Intelligence on Consumer Games and Graphics Hardware CIGPU in WCCI2012 at Brisbane, Parallel and Distributed Evolutionary Computation in the Cloud Era in CEC2013 at Cancun and CEC2015 at Sendai, and a member of program committee of GECCO since 1999.