

Adaptive Parallel Particle Swarm Optimization Algorithm Based on Dynamic Exchange of Control Parameters

Masaaki Suzuki

Department of Industrial Administration, Tokyo University of Science, Chiba, Japan

Email: m-suzuki@rs.tus.ac.jp

How to cite this paper: Suzuki, M. (2016) Adaptive Parallel Particle Swarm Optimization Algorithm Based on Dynamic Exchange of Control Parameters. *American Journal of Operations Research*, 6, 401-413.

<http://dx.doi.org/10.4236/ajor.2016.65037>

Received: July 29, 2016

Accepted: September 25, 2016

Published: September 28, 2016

Copyright © 2016 by author and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Updating the velocity in particle swarm optimization (PSO) consists of three terms: the inertia term, the cognitive term and the social term. The balance of these terms determines the balance of the global and local search abilities, and therefore the performance of PSO. In this work, an adaptive parallel PSO algorithm, which is based on the dynamic exchange of control parameters between adjacent swarms, has been developed. The proposed PSO algorithm enables us to adaptively optimize inertia factors, learning factors and swarm activity. By performing simulations of a search for the global minimum of a benchmark multimodal function, we have found that the proposed PSO successfully provides appropriate control parameter values, and thus good global optimization performance.

Keywords

Swarm Intelligence, Particle Swarm Optimization, Global Optimization, Metaheuristics, Adaptive Parameter Tuning

1. Introduction

In the various aspects of optimization, there are cases where a globally optimal solution is not necessarily obtainable. In such cases, it is desirable to find instead a semi-optimal solution that can be computed within a practical timeframe. To achieve this goal, heuristic optimization techniques are popularly studied and used, typified by genetic algorithms (GA), simulated annealing (SA) and particle swarm optimization [1] (PSO). In addition, since multipoint search algorithms like GAs and PSO can determine a Pareto-optimal solution based on a one-time calculation, they are actively employed in applied research to handle multipurpose optimization problems.

If the objective function under consideration is multimodal, then heuristic optimization techniques are desired to have qualities including a global solution search ability, maintained by preservation of solution diversity; a local solution search ability, maintained conversely by centralization of the solution search; and a balance between these two. Solution diversification and centralization strategies are factors universally shared by heuristic optimization techniques, and influence their performance. However, there are few precise and universal guidelines for configuring the values of the parameters that control these strategies: their configuration is problem-specific. Additionally, tuning of these parameters is not simple, and generally requires many preliminary calculations. Furthermore, which parameter values are suitable may vary at every stage of the solution search; pertinent examples include the configuration of crossover rate and spontaneous mutation rate in GAs and the temperature cooling schedule in SA.

Focusing on PSO, a kind of multipoint search heuristic optimization technique, in this study we propose several parallel PSO algorithms in which control parameters are dynamically exchanged between a number of swarms and are adaptively adjusted during the solution search process. We also share our findings from an evaluation of algorithm performance on a minimum search problem for a multimodal objective function.

2. Particle Swarm Optimization

PSO is an evolutionary optimization calculation technique based on the concept of swarm intelligence. In PSO, the hypersurface of an objective function is searched as information is exchanged between swarms of search points, which simulate animals or insects. The next state of each individual is generated based on the optimal solution in its search history (“personal best”; ***pbest***), the optimal solution in the combined search history of all individuals in the swarm (“global best”; ***gbest***), and the current velocity vector. Briefly, assuming a population size N_p and problem dimension N_d , the position and velocity of an individual i (where $i = 1, \dots, N_p$) at the $t + 1^{\text{th}}$ step of the search, respectively $\mathbf{x}_i^{(t+1)}$ and $\mathbf{v}_i^{(t+1)}$, are:

$$\mathbf{x}_i^{(t+1)} = \left(x_{i,1}^{(t+1)}, \dots, x_{i,j}^{(t+1)}, \dots, x_{i,N_d}^{(t+1)} \right) \quad (1)$$

$$\mathbf{v}_i^{(t+1)} = \left(v_{i,1}^{(t+1)}, \dots, v_{i,j}^{(t+1)}, \dots, v_{i,N_d}^{(t+1)} \right) \quad (2)$$

These two variables can be updated by means of the following equation, using the position and velocity at the t^{th} step, $\mathbf{x}_i^{(t)}$ and $\mathbf{v}_i^{(t)}$:

$$v_{i,j}^{(t+1)} = \gamma \cdot v_{i,j}^{(t)} + c_1 \cdot \text{rand}1_{i,j} \cdot \left(pbest_{i,j}^{(t)} - x_{i,j}^{(t)} \right) + c_2 \cdot \text{rand}2_{i,j} \cdot \left(gbest_j^{(t)} - x_{i,j}^{(t)} \right) \quad (3)$$

$$x_{i,j}^{(t+1)} = x_{i,j}^{(t)} + v_{i,j}^{(t+1)} \quad (4)$$

Here, $pbest_i^{(t)}$ represents the optimal solution discovered during the search through the t^{th} step by individual i itself, while $gbest^{(t)}$ represents the optimal solution discovered during the search through the t^{th} step by the swarm to which individual i belongs. The term γ represents inertia, and takes a value between $[0, 1]$ (inertia factor); c_1 and c_2 are weighting factors, respectively called the cognitive learning factor and

the social learning factor (learning factors); and $rand1$ and $rand2$ are uniform random numbers in $[0, 1]$. The PSO solution search procedure is described below:

1. Decide population size and maximum number of search steps.
2. Set initial position and velocity of each individual.
3. Calculate objective function value for each individual.
4. Determine the optimal individual solution *pbest* for each individual and the optimal swarm solution *gbest*, and update these values.
5. Update the position and velocity of each individual according to Equation (3) and (4).
6. End search if desired solution accuracy is obtained or if maximum number of steps is reached.

The algorithm behind PSO is simpler than a GA, another multipoint search heuristic optimization technique, making it easier to code and tending to lead to faster solution convergence. On the other hand, PSO sometimes loses solution diversity during the search, which readily invites excessive convergence. In response, improved PSO techniques have begun to be proposed around the world. Examples include distributed PSO and hierarchical PSO, which search the solution space with multiple different swarms [2] [3]; a method that performs a global search for its initial calculations but intensively searches the area of suboptimal solutions thereafter similar to SA [4]; a technique that incorporates bounded rational randomness thereafter, like the “lazy ant” in ant colony optimization; and a method that avoids local solutions if the algorithm become caught in them for a while. As with many other heuristic optimization techniques, PSO includes several option control parameters that analysts can set. Because these settings can greatly influence search performance, theoretical research on stability and convergence due to parameter values [5] [6] and research and development on PSO with adaptive parameter tuning functions (e.g., [7] [8]) are underway. The tuning of the quantum PSO (QPSO) [9] is simpler compared to standard PSO since QPSO has only a single control parameter.

3. Proposed Method

In this study, we focus on several parameters to control diversification and centralization in solution search: the inertia factor γ , learning factors c_1 and c_2 , and swarm activity (described in Section 3.3). Introducing concepts similar to those employed in the replica-exchange method [10] and parallel SA method, we propose parallel PSO algorithms in which parameter values are adaptively adjusted via dynamic exchange of the above control parameters between multiple swarms during the solution search process.

The replica-exchange method was developed in response to problems like spin glass and protein folding, in which it is difficult to find the ground-energy state (a global optimum solution) because several semi-stable states (local optimum solutions) exist in the system. In the replica-exchange method, several replicas of the original system are prepared, which have different temperatures and never interact with each other. We

encourage readers to imagine “temperature” here as the temperature parameter in Metropolis Monte Carlo simulations, *i.e.*, it indicates the degree to which deterioration is permitted when making the decision to transition to a candidate in the next state. Solution searches in high-temperature systems exhibit behavior close to a random search, whereas solution searches in low-temperature systems exhibit behavior close to the steepest descent method. Solution search calculations are run independently and simultaneously for each replica, each at its respective constant temperature. At the same time, temperature is exchanged periodically after a certain number of search steps according to the exchange probability w in the following equations between a given replica pair (with respective states $X_{T=T_k}$ and $X'_{T=T_{k+1}}$) having adjacent temperatures (T_k and T_{k+1}).

$$w(\{X_{T=T_k}, X'_{T=T_{k+1}}\} \rightarrow \{X_{T=T_{k+1}}, X'_{T=T_k}\}) = \begin{cases} 1 & \text{for } \Delta \leq 0 \\ \exp(-\Delta) & \text{for } \Delta > 0 \end{cases} \tag{5}$$

$$\Delta \equiv \left(\frac{1}{T_k} - \frac{1}{T_{k+1}} \right) \{ E(X'_{T=T_{k+1}}) - E(X_{T=T_k}) \} \tag{6}$$

Here, $E(X)$ represents the energy of a replica at state X (*i.e.*, the objective function value). **Figure 1** shows a schematic diagram of the replica-exchange method. In the replica-exchange method, high-temperature calculations correspond to retention of solution diversity, while low-temperature calculations correspond to a local solution search. Moreover, we can argue that it has some qualities of heuristic optimization algorithms for multimodal objective functions, in that its calculations are repeated as temperatures are probabilistically exchanged. Unlike SA, in which temperature falls monotonically, in this technique the temperature meanders if we focus on a single given replica. Thus, one can use this method to search a large solution space without becoming caught in a semi-s state.

3.1. Inertia-Factor Parallel PSO

Here, we first propose a technique focusing on the inertia factor γ , a control parameter in Equation (3). The search trajectories of individuals with large γ are more curved,

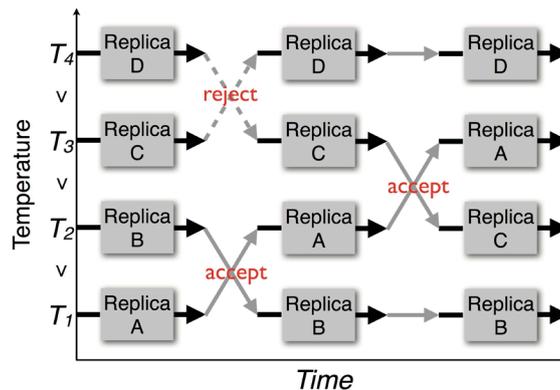


Figure 1. Schematic illustration of the replica-exchange method, with four replicas.

whereas those of individuals with small γ converge to an intermediate point between *pbest* and *gbest* (dependent on c_1 and c_2). Thus, efficient optimization should be achievable if, in the initial search, individuals are given large γ values and the solution search space is wide, while in its final stages, individuals are instead given small γ values and a solution is searched for intensively at *pbest*, *gbest*, and the area between them. For this reason, each individual's γ is typically reduced linearly with increasing search step t , according to the following equation:

$$\gamma^{(t)} = \gamma_{\max} - \frac{\gamma_{\max} - \gamma_{\min}}{t_{\max}} \cdot t \tag{7}$$

Here, γ_{\max} and γ_{\min} respectively represent the maximum and minimum inertia factors, and t_{\max} is the maximum number of search steps. Note that there is no single optimal reduction schedule one can choose for inertia factor γ : in truth, multiple techniques have been proposed besides the linear reduction described above [11], including exponential reduction [12] and stepwise reduction methods [13].

We consider N_s swarms with various different γ values in the Inertia-factor Parallel PSO (IP-PSO) proposed in this section. We assign $\gamma_k (k = 1, \dots, N_s)$ in this paper using the following equation:

$$\gamma_k = \frac{\gamma_{\max} - \gamma_{\min}}{N_s - 1} \cdot (k - 1) + \gamma_{\min} \tag{8}$$

In IP-PSO, each swarm has its own *gbest*, the optimal solution found across all individuals in the swarm. Periodically, after a certain number of search steps, the objective function $f(\textit{gbest})$ values of two swarms ($S_{\gamma=\gamma_k}$ and $S'_{\gamma=\gamma_{k+1}}$) having adjacent γ_k values are compared. Each γ_k value is then probabilistically exchanged (or not) according to the Metropolis decisions in Equations (9) and (10).

$$w(\{S_{\gamma=\gamma_k}, S'_{\gamma=\gamma_{k+1}}\} \rightarrow \{S_{\gamma=\gamma_{k+1}}, S'_{\gamma=\gamma_k}\}) = \begin{cases} 1 & \text{for } \Delta \leq 0 \\ \exp(-\Delta) & \text{for } \Delta > 0 \end{cases} \tag{9}$$

$$\Delta \equiv \left(\frac{1}{\gamma_k} - \frac{1}{\gamma_{k+1}} \right) \left\{ f(\textit{gbest}_{\gamma=\gamma_{k+1}}) - f(\textit{gbest}_{\gamma=\gamma_k}) \right\} \tag{10}$$

Figure 2 shows a schematic diagram of IP-PSO. The IP-PSO (and also the other proposed adaptive parallel PSOs described in Sections 3.2 - 3.4) employs the Metropolis criterion to determine the exchange acceptance of the control parameter rather than the move acceptance of each solution. The Metropolis decision will assign smaller γ_k value to the swarm having the superior $f(\textit{gbest})$ value with higher probability. (Note: “superior” here means “smaller”, since this paper is concerned with minimum search problems.) As a result, a more intensive search can be performed in the vicinity of *gbest*. On the other hand, it is also possible to escape local optimum solutions by a global search, because larger γ_k value is assigned to the swarm having the inferior $f(\textit{gbest})$ value with higher probability. In addition, unlike the related methods mentioned above in which γ decreases monotonically, this method can escape local optimum solutions, even if it becomes stuck during a search with a small γ value, be-

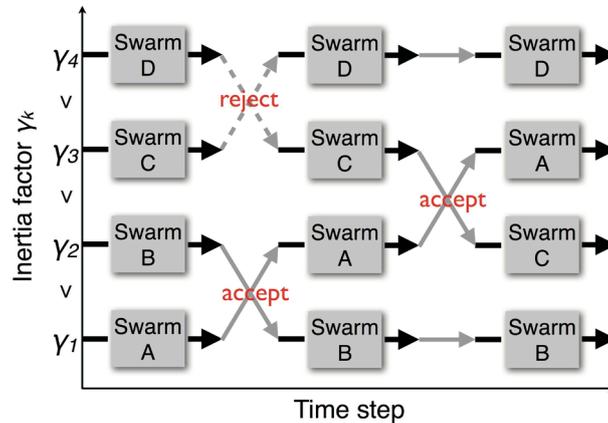


Figure 2. Schematic illustration of the Inertia-factor Parallel PSO (IP-PSO), with four swarms.

cause a larger γ value could be probabilistically assigned. The dynamic assignment of appropriate inertia factor γ values to each swarm according to the search conditions makes it unnecessary to configure a γ reduction schedule before carrying out optimization. The IP-PSO solution search procedure is described below:

1. Decide total population size, number of swarms, and maximum number of search steps.
2. Assign initial inertia factor values to each swarm according to Equation (8).
3. Set initial position and velocity of each individual.
4. Calculate objective function value for each individual.
5. Determine the optimal individual solution *pbest* for each individual and the optimal swarm solution *gbest*, and update these values.
6. Update the position and velocity of each individual according to Equation (3) and (4).
7. Periodically, after a certain number of search steps, compare objective function values between two swarms having adjacent inertia factor values; make the decision to exchange inertia factors according to Equations (9) and (10).
8. End search if desired solution accuracy is obtained or if maximum number of search steps is reached.

Because each swarm can be simulated independently and simultaneously with only a slight communication cost, the IP-PSO (and also the other proposed adaptive parallel PSOs described in Sections 3.2 - 3.4) are well suited for and very efficiently runs on massively parallel computers.

3.2. Learning-Factor Parallel PSO

Here, we propose a technique focusing on learning factors c_1 and c_2 , control parameters in Equation (3). For individuals with relatively large c_1 , PSO searches in the vicinity of the optimal solution in that individual's search history, *pbest*, whereas for individuals with large c_2 , it searches in the vicinity of the optimal solution in the search history of the swarm, *gbest*. Thus, efficient optimization should be realizable if, in the initial

search, individuals are given large c_1 and small c_2 values to ensure solution diversity, while in its final stages, individuals are instead given small c_1 and large c_2 values in an attempt to centralize the search in the vicinity of **gbest**. Some time-change schedules have learning factors c_1 and c_2 decrease (or increase) linearly with increasing number of search steps [14].

For the Learning-factor Parallel PSO (LP-PSO) proposed in this section, we introduce the allocation parameter α ($0 \leq \alpha \leq 1$), which regulates the balance between learning factors c_1 and c_2 , and define the LP-PSO learning factors c'_1 and c'_2 according to the following equation:

$$\begin{cases} c'_1 = 2\alpha c_0 \\ c'_2 = 2(1-\alpha)c_0 \end{cases} \quad (11)$$

Here, c_0 is a constant. This paper uses $c_0 = 1.4955$, a learning factor determined to be stable in a PSO stability analysis run by Clerc *et al.* [5]. When $\alpha = 1$ in Equation (11), $c'_2 = 0$ and the search is run in the vicinity of **pbest**; when $\alpha = 0$, $c'_1 = 0$ and the search is run in the vicinity of **gbest**. Similar to IP-PSO in Section 3.1, we consider N_s swarms having various different α values. We determine α_k ($k = 1, \dots, N_s$) in this paper using the following equation:

$$\alpha_k = \frac{\alpha_{\max} - \alpha_{\min}}{N_s - 1} \cdot (k - 1) + \alpha_{\min} \quad (12)$$

Thereafter, as in IP-PSO, each swarm has its own **gbest**, the optimal solution found across all individuals in the swarm. Periodically, after a certain number of search steps, the objective function $f(\mathbf{gbest})$ values of two swarms having adjacent α_k values are compared. Each α_k value is then probabilistically exchanged (or not) according to the Metropolis decisions in the same manner as Equations (9) and (10). The decision will assign smaller α_k value to the swarm having the superior $f(\mathbf{gbest})$ value (*i.e.*, small c'_1 and large c'_2) with higher probability. As a result, a more-intensive search can be performed in the vicinity of **gbest**. On the other hand, it is also possible to escape local optimum solutions using a global search based on the **pbest** of each individual, because larger α_k value (*i.e.*, large c'_1 and small c'_2) is assigned to the swarm having the inferior $f(\mathbf{gbest})$ value with higher probability. The assignment of appropriate learning factor c'_1 and c'_2 values to each swarm according to the search conditions actually makes it unnecessary to configure a time-change schedule before carrying out optimization.

3.3. Activity Parallel PSO

Here, we propose a technique focusing on the control parameter for swarm activity. Yasuda *et al.* [15] used molecular motion as an analogy for the movement of each individual in PSO, and defined the activity of a swarm Act as an index of the diversification/centralization of a solution search according to the following equation:

$$Act = \left(\frac{1}{N_p \cdot N_d} \sum_{i=1}^{N_p} \sum_{j=1}^{N_d} V_{i,j}^2 \right)^{1/2} \quad (13)$$

Swarms with high activity have many individuals with high velocities, and search over a wide solution space. Swarms with low activity, on the other hand, have many individuals with low velocities, and so search intensively for local solutions. Activity is observed moment-to-moment, because searches in which activity decreases gradually and continually can yield favorable solutions. In the event that measured activity is lower than the preset baseline activity, increasing the inertia factor γ of each individual promotes global searches; in the event that measured activity is higher than the preset baseline activity, decreasing the inertia factor promotes local searches. These behaviors thus constitute adaptive parameter regulation (of the inertia factor). However, the baseline activity reduction schedule must be set appropriately in advance, such that it decreases gradually with increasing search steps.

For the Activity Parallel PSO (AP-PSO) proposed in this section, we directly control swarm activity (*i.e.*, what was measured in [15], the past study mentioned above) in a manner similar to temperature control methods in molecular dynamics applications. Each individual's velocity should be appropriately scaled at each search step in order to control the measured, actual activity Act (defined by Equation (13)) at the target activity Act_0 . Briefly, the scaling factor s is calculated according to the following equation, where the velocity \mathbf{v}_i of each individual is converted to $\mathbf{v}'_i = s \cdot \mathbf{v}_i$:

$$s = Act_0 \cdot \left(\frac{N_p \cdot N_d}{\sum_{i=1}^{N_p} \sum_{j=1}^{N_d} v_{i,j}^2} \right)^{1/2} \quad (14)$$

We consider N_s swarms controlled by various different target activity Act_0 values. We determine $Act_{0,k}$ ($k = 1, \dots, N_s$) in this paper using the following equation:

$$Act_{0,k} = \frac{Act_{0,\max} - Act_{0,\min}}{N_s - 1} \cdot (k - 1) + Act_{0,\min} \quad (15)$$

Thereafter, as in IP-PSO and LP-PSO, each swarm has its own ***gbest***, the optimal solution found across all individuals in the population. Periodically, after a certain number of search steps, the objective function $f(\mathbf{gbest})$ values of two swarms having adjacent $Act_{0,k}$ values are compared. Each $Act_{0,k}$ value is then probabilistically exchanged (or not) according to the Metropolis decisions in the same manner as Equations (9) and (10). The decision will assign smaller $Act_{0,k}$ value to the swarm having the superior $f(\mathbf{gbest})$ value with higher probability. As a result, a more-intensive search can be performed in the vicinity of ***gbest***. On the other hand, it is also possible to escape local optimum solutions using a global search, because larger $Act_{0,k}$ value is assigned to the swarm having the inferior $f(\mathbf{gbest})$ value with higher-probability. Assigning an appropriate activity value to each swarm according to the search conditions makes it unnecessary to configure an activity reduction schedule in advance.

3.4. PSO with Simultaneous Exchange of Multiple Control Parameters

The proposed PSO techniques in Sections 3.1 - 3.3 above focus on only one kind of control parameter at a time, and assign parameter values that differ between each

swarm. Nonetheless, adaptive control is possible if several control parameters are simultaneously and dynamically exchanged between swarms. For example, we can consider N_s swarms each having a different inertia factor γ and target activity Act_0 ; we call this technique the Inertia-factor and Activity Parallel PSO (IAP-PSO). For IP-PSO in Section 3.1, a given γ_k value corresponds one-to-one with a given swarm; however, with IAP-PSO, a given $(\gamma_k, Act_{0,k})$ pair corresponds one-to-one with a given swarm. These control parameter pairs are exchanged between swarms. Figure 3 shows a schematic diagram of IAP-PSO. We can consider an Inertia-factor and Learning-factor Parallel PSO (ILP-PSO) the same way: in it, the inertia factor and learning factors are simultaneously exchanged.

4. Numerical Simulation and Discussion

We evaluate the performance of the proposed techniques using a minimum search problem for a Rastrigin function, a representative multimodal function. Our Rastrigin function is represented by the following equation:

$$f(\mathbf{x}) = \sum_{j=1}^{N_d} [x_j^2 - 10 \cos(2\pi x_j) + 10] \tag{14}$$

The Rastrigin function is multimodal, its variables are completely independent of each other, and it has a minimum value of $f(\mathbf{x}^*) = 0$ for $\mathbf{x}^* = (0, 0, \dots, 0)$. In the performance evaluation experiments, the number of dimensions was set at $N_d = 100$, and the initial coordinates and initial velocity of each individual were set according to uniform random numbers in the respective ranges $-100 \leq x_{i,j}^{(0)} \leq 100$ and $0 \leq v_{i,j}^{(0)} \leq 100$.

We first evaluated the proposed PSO techniques, in which only a single control parameter is exchanged in the search process. We observed the relationship between successful transitions in control parameter value and changes in objective function value, and compared its performance with other techniques. Specifically, we compared a Linearly decreasing Inertia factor PSO (LDI-PSO), in which the inertia factor linearly and continually decreases according to Equation (7) with increasing search step t , with the proposed techniques IP-PSO, LP-PSO, and AP-PSO. Table 1 shows the major simulation conditions for each technique.

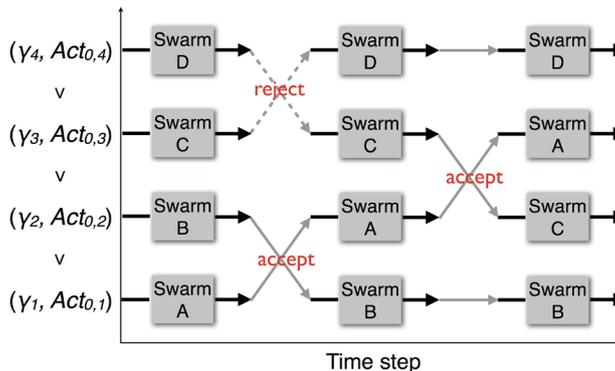


Figure 3. Schematic illustration of the Inertia-factor and Activity Parallel PSO (IAP-PSO), with four swarms.

Table 1. Summary of simulation conditions for performance evaluation of the proposed PSOs which exchange a single control parameter.

	LDI-PSO	IP-PSO	LP-PSO	AP-PSO
Dimension N_d			100	
Total number of individuals N_p			6400	
Number of swarms N_s	1	8	8	8
Inertia factor $\gamma = [0.4, 0.9]$	Linearly decreasing	Dynamic exchange	Linearly decreasing	Linearly decreasing
Allocation factor $\alpha = [0, 1]$	0.5	0.5	Dynamic exchange	0.5
Target activity $Act_0 = [1, 50]$	-	-	-	Dynamic exchange
Maximum number of steps t_{max}			3000	

Figure 4 shows time series data for the objective function $f(gbest)$ and the control parameter values obtained via each technique. The time series shown for LDI-PSO is data for the best of eight search attempts, assuming 6400 individuals \times 1 swarm. The time series shown for IP-PSO, LP-PSO, and AP-PSO are respective data for the best swarm within a representative search attempt (eight search attempts in total), assuming 800 individuals \times 8 swarms. In IP-PSO, adaptive control is realized through the dynamic exchange of inertia factor values, which occurs spontaneously without the need to configure a stepwise reduction schedule for the inertia factor, a condition seen in the aforementioned [13]. Compared with LDI-PSO, which also uses the inertia factor as a control parameter and has a linear reduction schedule for the inertia factor, IP-PSO achieves lower objective function values. Looking at LP-PSO, on the other hand, small α_k (*i.e.*, small c'_1 and large c'_2) values are assigned from around the 350th search step onward, in response to the relatively small objective function values obtained in the initial search around that step number. The solution ceases to improve for a while thereafter, but large α_k (*i.e.*, large c'_1 and small c'_2) values were assigned from step 1150 to around step 1700; as a result, the trajectory escapes the local optimum solution, and the solution continues to improve from around step 1400 onwards. After around the 1700th search step, small α_k values are assigned once more; as a result of this search centralization, the objective function continues to drop until the maximum (*i.e.*, final) search step. For AP-PSO, the solution continually improves as activity frequently fluctuates. The above results obtained with each proposed PSO technique show that the observed diverse shifts in control parameters depend on the search conditions. LP-PSO and AP-PSO achieved superior results to LDI-PSO and IP-PSO, with final objective function values of, respectively, 24.5 and 25.0 versus 135.3 and 80.6.

We next evaluated the proposed PSO techniques in which multiple parameters are exchanged simultaneously. The four techniques compared were (1) linearly decreasing inertia-factor and learning-factor PSO (LDIL-PSO), in which the inertia factor γ and

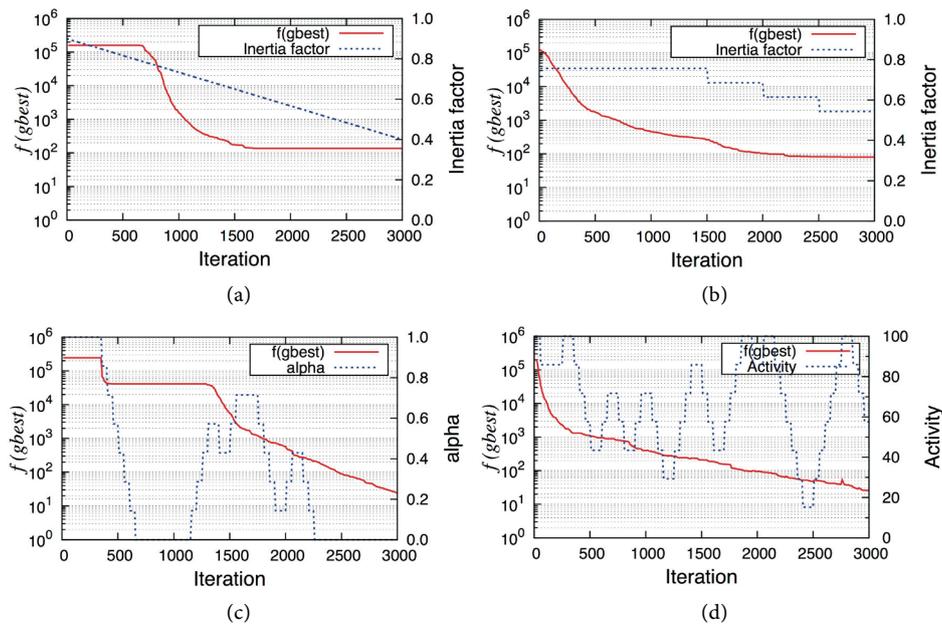


Figure 4. Performance evaluation of IP-PSO, LP-PSO, and AP-PSO: Time series of $f(gbest)$ and control parameter value $(\gamma, \alpha$ or $Act_0)$. (a) LDI-PSO; (b) IP-PSO; (c) LP-PSO; (d) AP-PSO.

the allocation parameter for learning factors α both linearly decrease with increasing step number t ; (2) ILP-PSO, in which (γ_k, α_k) pairs are simultaneously and dynamically exchanged between swarms; (3) linearly decreasing inertia-factor and activity PSO (LDIA-PSO), in which the inertia factor γ and target activity Act_0 both linearly decrease with increasing step number t ; and (4) IAP-PSO, in which $(\gamma_k, Act_{0,k})$ pairs are simultaneously and dynamically exchanged between swarms. **Table 2** shows the major simulation conditions for each technique.

Figure 5 shows time series plots for the objective function $f(gbest)$ obtained via each technique. The time series shown for LDIL-PSO and LDIA-PSO are the data for the best of eight search attempts, assuming 6400 individuals \times 1 swarm. The time series shown for ILP-PSO and IAP-PSO are the data for the best swarm within a representative search attempt (eight search attempts in total), assuming 800 individuals \times 8 swarms. We can see that the simultaneous adjustment of multiple control parameters improves search performance compared with **Figure 4**. This is true if we compare those techniques in which control parameters are linearly reduced (LDI-PSO vs. LDIL-PSO, LDIA-PSO) with one another, as well as if we compare those techniques in which control parameters are dynamically exchanged (LP-PSO and AP-PSO vs. ILP-PSO and IAP-PSO). This is because the simultaneous adjustment of multiple control parameters enables each swarm to approach the equilibrium state corresponding to the parameter values at the time more rapidly. ILP-PSO and IAP-PSO yielded mean final objective function values of 2.6 and 20.6, respectively. Performance enhancement was particularly pronounced in ILP-PSO, which achieved the best objective function value of all techniques.

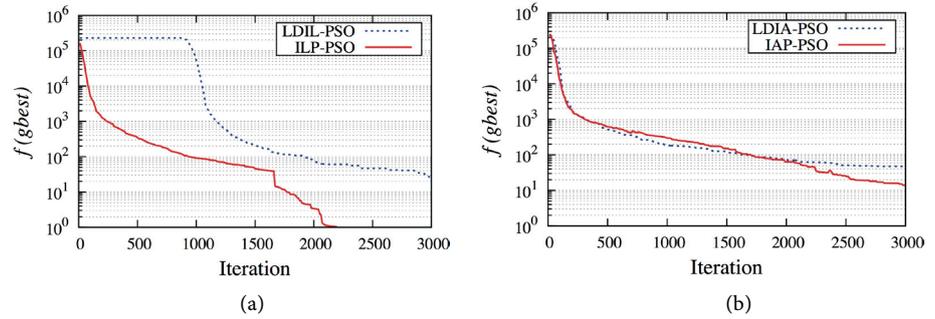


Figure 5. Performance evaluation of ILP-PSO and IAP-PSO: Time series of $f(gbest)$. (a) LDIL-PSO vs. ILP-PSO; (b) LDIA-PSO vs. IAP-PSO.

Table 2. Summary of simulation conditions for performance evaluation of the proposed PSOs which exchange multiple control parameters.

	LDIL-PSO	ILP-PSO	LDIA-PSO	IAP-PSO
Dimension N_d			100	
Total number of individuals N_p		6400		
Number of swarms N_s	1	8	1	8
Inertia factor $\gamma = [0.4, 0.9]$	Linearly decreasing	Dynamic exchange	Linearly decreasing	Dynamic exchange
Allocation factor $\alpha = [0, 1]$	Linearly decreasing	Dynamic exchange	0.5	0.5
Target activity $Act_0 = [1, 50]$	-	-	Linearly decreasing	Dynamic exchange
Maximum number of steps t_{max}		3000		

5. Conclusion

We proposed five types of adaptive parallel PSO algorithms that employed the dynamic exchange of control parameters between multiple swarms-IP-PSO, LP-PSO, AP-PSO, ILP-PSO and IAP-PSO while focusing on the PSO control parameters of inertia factor, learning factors and swarm activity. The proposed algorithms were adopted to adaptively regulate control parameters at each step of the search in an experiment consisting of a minimum search problem for a multimodal function. The results show the systems transition appropriately between global and local solution search phases, meaning that efficient searches that do not stall at local optimum solutions are possible. Superior objective function values were obtained by ILP-PSO in particular: this method achieves adaptive regulation through the simultaneous exchange of the inertia factor and learning factors. Additional numerical experiments and assessments of the performance characteristics with a larger set of testing pool such as CEC benchmark are important topics for future research.

References

[1] Kennedy, J. and Eberhart, R.C. (1995) Particle Swarm Optimization. *Proceeding of IEEE*

- International Conference on Neural Networks*, **4**, 1942-1948.
<http://dx.doi.org/10.1109/ICNN.1995.488968>
- [2] Janson, S. and Middendorf, M. (2005) A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **35**, 1272-1282. <http://dx.doi.org/10.1109/TSMCB.2005.850530>
- [3] Yen, G.G. and Daneshyari, M. (2006) Diversity-Based Information Exchange among Multiple Swarms in Particle Swarm Optimization. *Proceeding of IEEE Congress on Evolutionary Computation (CEC 2006)*, Vancouver, 16-21 July 2006, 6150-6157.
<http://dx.doi.org/10.1109/cec.2006.1688511>
- [4] Eberhart, R.C. and Shi, Y. (2000) Comparing Inertia Weights and Constrictions Factors in Particle Swarm Optimization. *Proceeding of the Congress on Evolutionary Computation (CEC 2000)*, San Diego, 16-19 July 2000, 84-88. <http://dx.doi.org/10.1109/cec.2000.870279>
- [5] Clerc, M. and Kennedy, J. (2002) The Particle Swarm: Explosion, Stability, and Convergence in a Multi-Dimensional Complex Space. *IEEE Transactions on Evolutionary Computation*, **6**, 58-73. <http://dx.doi.org/10.1109/4235.985692>
- [6] Parsopoulos, K.E. and Vrahatis, M.N. (2002) Recent Approaches to Global Optimization Problems through Particle Swarm Optimization. *Natural Computing*, **1**, 235-306.
<http://dx.doi.org/10.1023/A:1016568309421>
- [7] Cooren, Y., Clerc, M. and Siarry, P. (2009) Performance Evaluation of TRIBES, an Adaptive Particle Swarm Optimization Algorithm. *Swarm Intelligence*, **3**, 149-178.
<http://dx.doi.org/10.1007/s11721-009-0026-8>
- [8] Jana, N.D. and Sil, J. (2014) Particle Swarm Optimization with Lévy Flight and Adaptive Polynomial Mutation in Gbest Particle. *Recent Advances in Intelligent Informatics*, **235**, 275-282. http://dx.doi.org/10.1007/978-3-319-01778-5_28
- [9] Mikki, S.M. and Kishk, A.A. (2006) Quantum Particle Swarm Optimization for Electromagnetics. *IEEE Transactions on Antennas and Propagation*, **54**, 2764-2775.
<http://dx.doi.org/10.1109/TAP.2006.882165>
- [10] Hukushima, K. and Nemoto, K. (1996) Exchange Monte Carlo Method and Application to Spin Glass Simulations. *Journal of the Physical Society of Japan*, **65**, 1604-1608.
<http://dx.doi.org/10.1143/JPSJ.65.1604>
- [11] Shi, Y. and Eberhart, R.C. (1998) Parameter Selection in Particle Swarm Optimization, Evolutionary Programming VII. *Lecture Notes in Computer Science*, **1447**, 591-600.
<http://dx.doi.org/10.1007/BFb0040810>
- [12] Li, H.-R. and Gao, Y.-L. (2009) Particle Swarm Optimization Algorithm with Exponent Decreasing Inertia Weight and Stochastic Mutation. *Proceeding of 2009 2nd International Conference on Information and Computing Science*, **1**, 66-69.
<http://dx.doi.org/10.1109/ICIC.2009.24>
- [13] Xin, J., Chen, G. and Hai, Y. (2009) A Particle Swarm Optimizer with Multi-Stage Linearly-Decreasing Inertia Weight. *Proceeding of the 2009 International Joint Conference on Computational Sciences, and Optimization*, **1**, 505-508.
<http://dx.doi.org/10.1109/CSO.2009.420>
- [14] Ratnaweera, A., Halgamuge, S.K. and Watson, H.C. (2004) Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients. *IEEE Transactions on Evolutionary Computation*, **8**, 240-255. <http://dx.doi.org/10.1109/TEVC.2004.826071>
- [15] Yasuda, K., et al. (2008) Particle Swarm Optimization: A Numerical Stability Analysis and Parameter Adjustment Based on Swarm Activity. *IEEE Transactions on Electrical and Electronic Engineering*, **3**, 642-659. <http://dx.doi.org/10.1002/tee.20326>



Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

Or contact ajor@scirp.org