Hybrid Particle Swarm Optimization with Bat Algorithm

Tien-Szu Pan[1], Thi-Kien Dao[1], Trong-The Nguyen[2], and Shu-Chuan Chu[3]

[1] Department of Electronics Engineering,
National Kaohsiung University of Applied Sciences, Taiwan
[2] Faculty of Information Technology, Haiphong Private University, Vietnam
[3] School of Computer Science, Engineering and Mathematics,
Flinders University, Australia
vnthe@hpu.edu.vn

Abstract. In this paper, a communication strategy for hybrid Particle Swarm Optimization (PSO) with Bat Algorithm (BA) is proposed for solving numerical optimization problems. In this work, several worst individuals of particles in PSO will be replaced with the best individuals in BA after running some fixed iterations, and on the contrary, the poorer individuals of BA will be replaced with the finest particles of PSO. The communicating strategy provides the information flow for the particles in PSO to communicate with the bats in BA. Six benchmark functions are used to test the behavior of the convergence, the accuracy, and the speed of the approached method. The results show that the proposed scheme increases the convergence and accuracy more than BA and PSO up to 3% and 47% respectively.

Keywords: Hybrid Particle Swarm Optimization with Bat Algorithm, Particle Swarm Optimization Algorithm, Bat Algorithm Optimizations, Swarm Intelligence.

1 Introduction

Computational intelligence algorithms have been prosperously used to solve optimization problems in engineering, economic, and management fields for recently years. For example, genetic algorithms (GA) have been used prosperously in various applications, including engineering, the budgetary and the security area [1-3]. Particle swarm optimization (PSO) techniques have fortunately been used to forecast the exchange rates, to optimize related multiple interference cancellations [4-6], to construct the portfolios of stock, human perception [3, 7, 8]. Ant colony optimization (ACO) techniques have successfully been used to solve the routing problem of networks, the secure watermarking [9, 10]. Artificial bee colony (ABC) techniques have successfully been used to solve the lot-streaming flow shop scheduling problem [11]. Cat swarm optimization (CSO) [12] techniques have successfully been used to discover proper positions for information hiding [13].

Communication between two algorithms is to take the advantage of the strength points of each type of algorithms. The idea of this paper is based on communication

H. Sun et al. (eds.), *Genetic and Evolutionary Computing*,
Advances in Intelligent Systems and Computing 329, DOI: 10.1007/978-3-319-12286-1_5

strategies in parallel processing for swarm intelligent algorithms. Information between populations is only exchanged when the communication strategy is triggered. The parallel strategies simply share the computation load over several processors. The sum of the computation time for all processors can be reduced compared with the single processor works on the same optimum problem. In this paper, the concepts of parallel processing and communication strategy are applied to hybrid Particle Swarm Optimization with Bat Algorithm is presented. In this new method, the several poorer particles in PSO will be replaced with best artificial bats in Bat algorithm after running some fixed iterations and on the contrary, the poorer individuals of BA will be replaced with the better particles of PSO.

The rest of this paper is organized as follows: a brief review of PSO and BA is given in session 2; our analysis and designs for the hybrid PSO-BA is presented in session 3; a series of experimental results and the comparison between PSO, BA and hybrid PSO-BA are discussed in session 4; finally, conclusion is summarized in session 5.

2 Related Work

Particle swarm optimization (PSO) is a heuristic global optimization algorithm, based on the research of bird and fish flock in movement behavior, proposed by Kennedy, J. Eberhart, R. and Shi [14, 15]. The particles are randomly initialized and then freely fly across the multi-dimensional search space. While they are flying, its velocity and position are updated based on its own best experience and also of entire population. The updating policy will cause the particle swarm to move toward a region with a higher object value. The position of each particle is equivalent to a candidate solution of a problem. The particle moves according to an adjusted velocity, which is based on that particle's experience and the experience of its companions.

The original particle swarm optimization algorithm can be expressed as follows:

$$V_i^{t+1} = V_i^t + C_1 \times r_1(P_i^t - X_i^t) + C_2 \times r_2(G^t - X_i^t) \quad (1)$$

where V_i^t is the velocity of the *i-th* particle at the *t-th* iteration, C_1 and C_2 are factors of the speed control, r_1 and r_2 are random variables such that $0 \le r_1, r_2 \le 1$, P_i^t is the best previous position of the *i-th* particle at the *t-th* iteration, G^t is the *best* position amongst all the particles, from the first iteration to the t-$^{\text{th}}$ iteration, and X_i^t is the *i-th* particle for the *t-th* iteration.

$$X_i^{t+1} = X_i^t + V_i^{t+1}, i = 0,1,..N-1 \quad (2)$$

where N is the particle size, $-V_{max} \le V^{t+1} \le V_{max}$ (V_{max} is the maximum velocity).

A modified version of the particle swarm optimizer [15] and an adaption using the inertia weight which is a parameter for controlling the dynamics of flying of the modified particle swarm [16], have also been presented. The latter version of the modified particle swarm optimizer can be expressed as equation (3)

$$V_i^{t+1} = W^t \times V_i^t + C_1 \times r_1(P_i^t - X_i^t) + C_2 \times r_2(G^t - X_i^t) \tag{3}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1}, \quad i = 0,1,..N-1 \tag{4}$$

where W^t is the inertia weight at the t-$^{\text{th}}$ iteration.

Moreover, the Bat Algorithm (BA) is a new optimization algorithm, proposed by Xin-SheYang, based on swarm intelligence and the inspiration from observing the bats [17]. BA simulates parts of the echolocation characteristics of the micro-bat in the simplicity way. Three major characteristics of the micro-bat are employed to construct the basic structure of BA. The idealized rules in this method are listed as follows: The echolocation to detect the prey is utilized for all bats, but not all species of the bat do the same thing. However, the micro-bat, one of species of the bat is a famous example of extensively using the echolocation. Hence, the first characteristic is the echolocation behavior. The second characteristic is the frequency. The frequency is sent by the micro-bat with fixed frequency f_{min} and with a variable wavelength λ. The loudness A_0 is used to search for prey. The other characteristic of them are listed as follows:

1. Bats fly randomly with velocity v_i at position x_i. They can adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission r from 0 to1, depending on the proximity of their target;
2. There are many ways to adjust the loudness. For simplicity, the loudness is assumed to be varied from a positive large A_0 to a minimum constant value, which is denoted by Amin.

The movement of the virtual bat is simulated by equation (5) – equation (7):

$$f_i = f_{min} + (f_{max} - f_{min}) * \beta \tag{5}$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_{best}) * f_i \tag{6}$$

$$x_i^t = x_i^{t-1} + \mathrm{v_i^t} \tag{7}$$

where f is the frequency used by the bat seeking for its prey, f_{min} and f_{max}, represent the minimum and maximum value, respectively. xi denotes the location of the *i-th* bat in the solution space, vi represents the velocity of the bat, t indicates the current iteration, β is a random vector, which is drawn from a uniform distribution, and β ∈[0, 1], and x_{best} indicates the global near best solution found so far over the whole population.

In addition, the rate of the pulse emission from the bat is also taken to be one of the roles in the process. The micro-bat emits the echo and adjusts the wavelength depending on the proximity of their target. The pulse emission rate is denoted by the symbol r_i, and $r_i \in [0,1]$, where the suffix i indicates the *i-th* bat. In every iteration, a random number is generated and is compared with r_i. If the random number is greater than r_i, a local search strategy, namely, random walk, is detonated. A new solution for the bat is generated by equation (8):

$$x_{new} = x_{old} + \varepsilon A^t \tag{8}$$

where ε is a random number and $\varepsilon \in [-1,1]$, and the average loudness of all bats is represented at the current time step t. After updating the positions of the bats, the loudness A_i and the pulse emission rate r_i are also updated only whenever the global near best solution is updated and the random generated number is smaller than A_i. The update of A_i and r_i are operated by equation (9) and equation (10):

$$A_i^{t+1} = \alpha A_i^t \tag{9}$$

$$r_i^{t+1} = r_i^0[1 - e^{-\gamma t}] \tag{10}$$

where α and γ are constants. In Yang's experiments, $\alpha = \gamma = 0.9$ is used for the simplicity.

The process of BA is depicted as follows:

Step 1. Initialize the bat population, the pulse rates, the loudness, and define the pulse frequency

Step 2. Update the velocities to update the location of the bats, and decide whether detonate the random walk process.

Step 3. Rank the bats according to their fitness value, find the current best solution found so far, and then update the loudness and the emission rate.

Step 4. Check the termination condition to decide whether go back to step 2 or end the process and output the result.

3 The Proposed Hybrid PSO with BA

Hybrid optimization algorithm is structured by communication strategies between two algorithms in this paper. This idea is based on replacing the weaker individuals according to fitness evaluation of one algorithm with stronger individuals from other algorithm in parallel processing for swarm intelligent algorithms. Several groups in a parallel structure of hybrid algorithm are created by dividing the population into subpopulations. Each of the subpopulation evolves independently in regular iterations. They only exchange information between populations when the communication strategy is triggered. It results in taking advantage of the individual strengths of each type of algorithm. The replacement of weaker individuals in running algorithms will be achieved so on to get the benefit of the cooperation.

Hybrid Particle Swarm Optimization with Bat algorithm (hybrid PSO-BA) is designed based on original PSO and Bat algorithm. Each algorithm evolves by optimization independently, i.e. the PSO has its own individuals and the better solution to replace the worst artificial bats of BA. In contrast, the better artificial bats of BA are to replace the poorer individuals of PSO after running some fixed iterations. The total iteration contains R times of communication, where $R = \{R_1, 2R_1, 3R_1, ...\}$.

Let N be the number of population of hybrid PSO-BA, and N_1, N_2 be the number of population of PSO and BA respectively, where N_1 and N_2 are set to be $N/2$. If $t \cap R \neq \varphi$, k agents with the top k fitness in N_1 will be copied to N_2 to replace the same number of individuals with the worst fitness, where t denotes the current iteration count, R_1 and k are the predefined constants.

The diagram of the hybrid PSO-BA with communication strategy is shown in figure 1

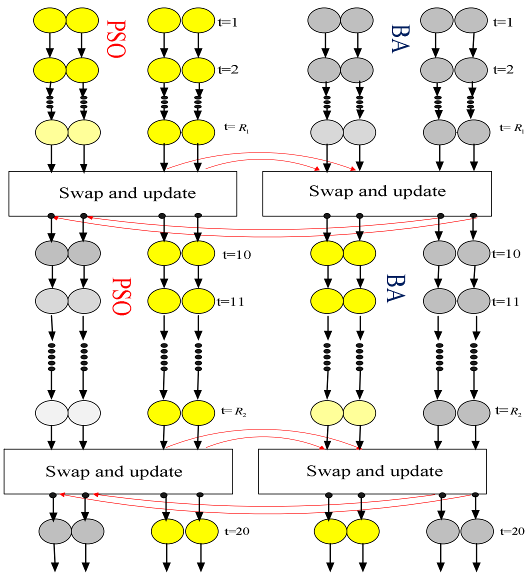

Fig. 1. The diagram of hybrid PSO-BA with a communication strategy

1. **Initialization:** Generate populations for both PSO and BA. Each population is initialized by BA or by PSO independently. Defined the iteration set R for executing the communication strategy. The N_1, N_2 are the number of particles and artificial agents in solutions S_{ij}^T and X_{ij}^T for populations of PSO and BA respectively, $i = 0, 1, \ldots, N_1 - 1$, $j = 0, 1,..D$. D is dimension of solutions and t is current iteration number. Set $t = 1$.

2. **Evaluation:** Evaluate the value of $f_1(S_{ij}^T)$, $f_2(X_{ij}^T)$ for both PSO and BA in each population. The evolvement of the populations is executed independently by both PSO and BA.

3. **Update:** Update the velocity and the positions of PSO using equation (1), and (2). Update the location and velocity of Bat in the best fitness value, which are found by the bat using equation (6), (7).

4. **Communication Strategy:** Migrate the best artificial bats among all the individuals of BA's population, copy k bats with the top k fitness in N_1 replace the poorer particles in N_2 of PSO's population and update for each population every R_1 iterations.

5. **Termination:** Repeat step 2 to step 5 until the predefined value of the function is achieved or the maximum number of iterations has been reached. Record the best value of the function $f(S^t)$ and the best particle position among all the particles S^t. Record the best value of the function $f(X^t)$ and the best location among all the bats X^t.

4 Experimental Results

This section presents simulation results and compares the hybrid PSO-BA with the primary PSO, and original BA, both in terms of solution quality, convergence capability, and the accuracy. The execution times in the number of function evaluations are also taken. Six benchmark functions are used to test the accuracy and the convergence of hybrid PSO-BA.

All the benchmark functions for experimenting are averaged over different random seeds with 10 runs. Let $S=\{s_1, s_2, ..., s_m\}$, $X=\{x_1, x_2, ..., x_m\}$ be the m-dimensional real-value vectors for PSO and BA respectively. The benchmark functions are Ackley, Griewank, Quadric, Rastrigin, Rosenbrock and Spherical. The equation numbers (11) to (16). The goal of the optimization is to minimize the outcome for all benchmarks. The population size of hybrid PSO-BA, primary PSO and original BA are set to 20 (N=20) for all the algorithms in the experiments. The detail of parameter settings of PSO can be found in [14], and setting of BA can be found in [17].

$$f_1(x) = 20 + e - 20e^{-0.2\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}} - e^{\frac{\sum_{j=1}^{n} \cos(2\pi x_j)}{n}} \tag{11}$$

$$f_2(x) = 1 + \sum_{i=1}^{N} \frac{x_i^2}{4000} + \prod_{i=1}^{N} cos \frac{x_i}{\sqrt{i}} \tag{12}$$

$$f_3(x) = \sum_{i=1}^{n} (\sum_{k=1}^{i} x_i) \tag{13}$$

$$f_4(x) = \sum_{i=1}^{N} [10 + x_i^2 - 10cos2\pi x_i \tag{14}$$

$$f_5(x) = \sum_{i=1}^{n-1} (100(x_{i-1} - x_i^2)^2 + (1 - x_i)^2 \tag{15}$$

$$f_6(x) = \sum_{i=1}^{N} x_i^2 \tag{16}$$

The initial range and the total iteration number for all test functions are listed in Table1.

Table 1. The initial range and the total iteration of test standard functions

Function		Initial range $[x_{min}, x_{max}]$	Total iteration
Ackley	$f_1(x)$	[-100,100]	200
Griewangk	$f_2(x)$	[5.12,5.12]	200
Quadric	$f_3(x)$	[-100,100]	200
Rastrigin	$f_4(x)$	[-30,30]	200
Rosenbrock	$f_5(x)$	[-100,100]	200
Spherical	$f_6(x)$	[-100,100]	200

The optimization for all of these test functions is to minimize the outcome. The parameters setting for hybrid PSO-BA with primary PSO side are the initial inertia weight $W = (0.9 - 07 * rand)$, coefficients of learning factors c_1=-2 and c_2=2 in PSO, the total population size N_1 = 10 and the dimension of the solution space M = *10*, and with original BA side are the initial loudness $A_i^0 = 0.25$, pulse rate $r_i^0 = 0.5$ the total population size N_1 = 10 and the dimension of the solution space M = *10*, frequency minimum f_{min} = *the lowest of initial range function* and frequency minimum f_{max} = *the highest of initial range function.* The proposed scheme is executed for 10 runs and each run contains 200 iterations. The final result is obtained by taking the average of the outcomes from all runs. These results also are compared with the primary PSO and original BA respectively.

Table 2 compares the quality of optimizing performance and time running for numerical problem optimization between hybrid PSO-BA and PSO. It is clearly seen that, almost these cases of testing benchmark functions for hybrid PSO-BA are better than PSO in terms of convergence and accuracy. It is special case with test function of Rosenbrock, $f_5(x)$ has the mean of value function minimum of total seeds of 10 runs is 1.02E+09 for hybrid PSO-BA performance evaluation, but, for original PSO is 2.90E+09, reaches at 48% improvement of convergence. The average performance evaluation value of six benchmark functions is 1.70E+08 for hybrid PSO-BA and 4.83E+08 for original PSO, gets at 47% improvement of accuracy. However, all benchmark functions for average time consuming of hybrid BA-BA are longer than that in original PSO, for the reasons, the hybrid algorithm must perform mutation and update operations.

Table 2. The comparison between hybrid PSO-BA and origianl PSO in terms of quality performance evaluation and speed

Function	Performance evaluation		Time running evaluation (seconds)	
	PSO	***Hybrid PSO-BA***	***PSO***	***Hybrid PSO-BA***
$f_1(x)$	1.96E+01	1.85E+01	0.079	0.134
$f_2(x)$	1.92E+00	1.81E+00	0.086	0.139
$f_3(x)$	4.46E+03	2.62E+03	0.109	0.230
$f_4(x)$	1.23E+02	1.11E+02	0.080	0.148
$f_5(x)$	2.90E+09	1.02E+09	0.081	0.159
$f_6(x)$	1.62E+04	7.13E+03	0.064	0.121
Average value	**4.83E+08**	**1.70E+08**	**0.33**	**0.49**

Figure 2 shows the experimental results of six benchmark functions in running repeatedly same iteration of 200 in random seeds of 10 runs. It clearly can be seen that the most cases of curves of hybrid PSO-BA (solid red line) are more convergence that its of PSO (doted blue line).

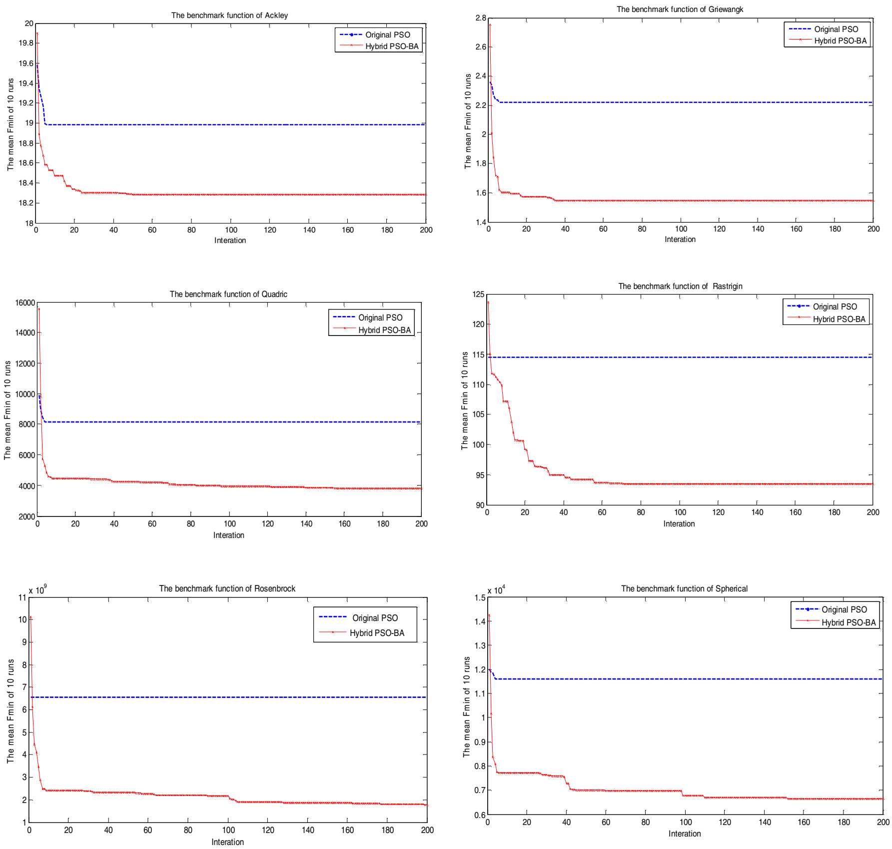

Fig. 2. The mean of function minimum curves in comparing Hybrid PSO-BA and original PSO algorithms for function of Ackley, Griewank, Quadric, Rastrigin, Rosenbrock and Spherical

Table 3 compares the quality of performance and time running for numerical problem optimization between Hybrid BA-PSO and original BA. It is clearly seen that, almost these cases of testing benchmark functions for Hybrid BA-PSO are more convergence than original BA.

Average value of all benchmark functions for hybrid BA- PSO is 2.32E+07 in performance evaluation, but this figure is 2.30E+07 for original BA, reaches at 3% improvement of accuracy. However, average times consuming of all benchmark functions for hybrid BA-PSO is longer taken than original BA. For this result, the reason is the hybrid algorithm must perform mutation and update operations.

Table 3. The comparison between hybrid BA-PSO and origianl BA in terms of quality performance evaluation and speed

Function	Performance evaluation		Time running evaluation (seconds)	
	BA	***Hybrid BA-PSO***	***BA***	***Hybrid BA-PSO***
$f_1(x)$	1.84E+01	1.65E+01	0.087	0.104
$f_2(x)$	7.37E-01	7.34E-01	0.094	0.119
$f_3(x)$	2.59E+03	2.05E+03	0.120	0.180
$f_4(x)$	4.67E+01	4.60E+01	0.087	0.098
$f_5(x)$	1.38E+08	1.38E+08	0.089	0.109
$f_6(x)$	2.65E+03	2.47E+03	0.071	0.071
Average value	**2.35E+07**	**2.30E+07**	**0.101**	**0.124**

Figure 3 shows the experimental results of six benchmark functions in running 10 seeds output with the same iteration of 200. It clearly can be seen that the most cases of curves of hybrid BA-PSO (solid red line) are more convergence that its of BA (doted blue line).

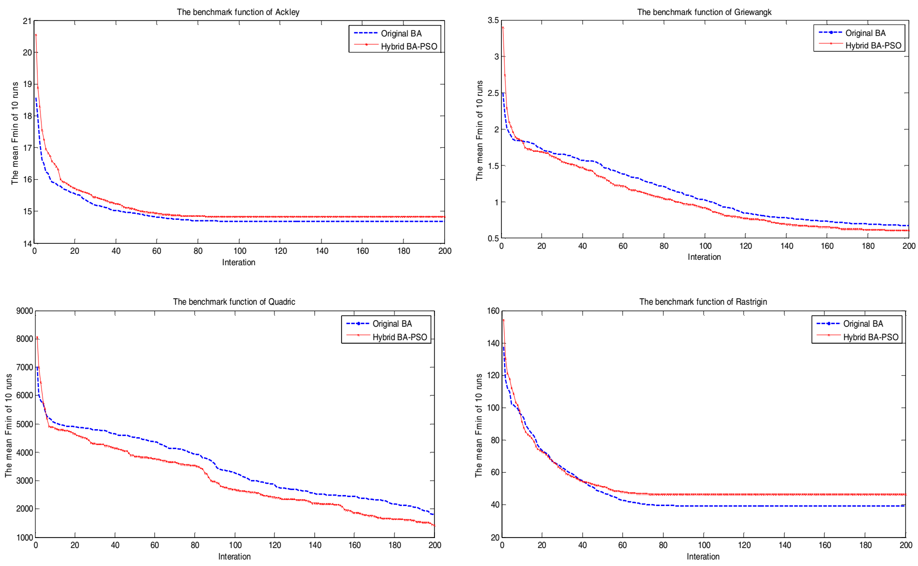

Fig. 3. The mean of function minimum curves in comparing hybrid BA-PSO and BA algorithms for function of Ackley, Griewank, Quadric, Rastrigin, Rosenbrock and Spherical

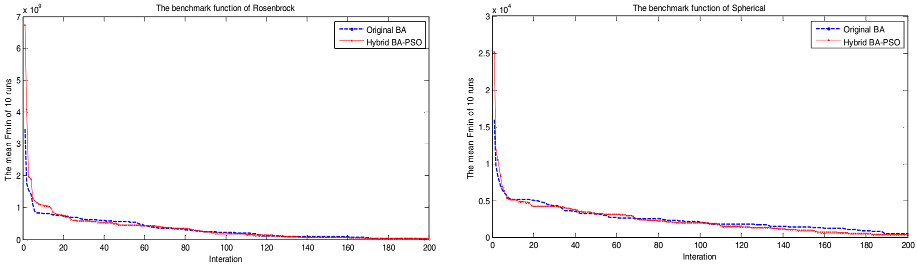

Fig. 3. (*continued*)

5 Conclusion

This paper, a novel proposed optimization scheme was presented, namely hybrid PSO-BA (hybrid Particle Swarm Optimization with Bat Algorithm). The implementation of hybrid for optimization algorithms could have important significance for taking advantages of the power of each algorithm and achieving cooperation of optimization algorithms. In the new proposed algorithm, the several worse individuals in PSO are replaced with the best artificial bats in BA algorithm after running some fixed iterations, and on the contrary, the poorer bats of BA are replaced with the better particles of PSO.

The proposed communication strategy provides the information flow for the particles to communicate in PSO with the bats in BA. The performance of hybrid PSO-BA algorithm is better than both original PSO and BA in terms of convergence and accuracy. The results the proposed algorithm on a set of various test problems show that hybrid PSO-BA increases the convergence and accuracy more than original PSO and original BA is up to 47 % is at 3% on finding the near best solution improvement.

References

1. Srinivas, M., Patnaik, L.M.: Genetic algorithms: a survey. Computer 27(6), 17–26 (1994)
2. Yang, B., Niu, X., Wang, S.: A Secure Steganography Method based on Genetic Algorithm. Journal of Information Hiding and Multimedia Signal Processing 1(1), 8 (2010)
3. Ruiz-Torrubiano, R., Suarez, A.: Hybrid Approaches and Dimensionality Reduction for Portfolio Selection with Cardinality Constraints. IEEE Computational Intelligence Magazine 5(2), 92–107 (2010)
4. Chen, S.-M., Chien, C.-Y.: Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. Expert Systems with Applications 38(12), 14439–14450 (2011)
5. Shyr, W.-J., Hsu, C.-H., Kuo, K.-H.: Optimizing Multiple Interference Cancellations of Linear Phase Array Based on Particle Swarm Optimization. Journal of Information Hiding and Multimedia Signal Processing 1(4), 292–300 (2010)

6. Chen, S.-M., Kao, P.-Y.: TAIEX forecasting based on fuzzy time series, particle swarm optimization techniques and support vector machines. Information Sciences 247(0), 62–71 (2013)
7. Jui-Fang, C., Shu-Wei, H.: The Construction of Stock_s Portfolios by Using Particle Swarm Optimization, p. 390
8. Puranik, P.B.P., Abraham, A., Palsodkar, P., Deshmukh, A.: Human Perception-based Color Image Segmentation Using Comprehensive Learning Particle Swarm Optimization. Journal of Information Hiding and Multimedia Signal Processing 2(3), 227–235 (2011)
9. Pinto, P.C., Nagele, A., Dejori, M., Runkler, T.A., Sousa, J.M.C.: Using a Local Discovery Ant Algorithm for Bayesian Network Structure Learning. IEEE Transactions on Evolutionary Computation 13(4), 767–779 (2009)
10. Chouinard, J.-Y., Loukhaoukha, K., Taieb, M.H.: Optimal Image Watermarking Algorithm Based on LWT-SVD via Multi-objective Ant Colony Optimization. Journal of Information Hiding and Multimedia Signal Processing 2(4), 303–319 (2011)
11. Pan, Q.-K., Tasgetiren, M.F., Suganthan, P.N., Chua, T.J.: A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. Inf. Sci. 181(12), 2455–2468 (2011)
12. Chu, S.-C., Tsai, P.-W.: Computational Intelligence Based on the Behavior of Cats. International Journal of Innovative Computing, Information and Control 3(1), 8 (2006)
13. Wang, Z.-H., Chang, C.-C., Li, M.-C.: Optimizing least-significant-bit substitution using cat swarm optimization strategy. Inf. Sci. 192, 98–108 (2012)
14. Kennedy, J., Eberhart, R.: Particle swarm optimization, vol. 4, pp. 1942–1948 (1995)
15. Yuhui, S., Eberhart, R.: A modified particle swarm optimizer, pp. 69–73 (1998)
16. Yuhui, S., Eberhart, R.C.: Empirical study of particle swarm optimization, vol. 3, p. 1950 (1999)
17. Yang, X.-S.: A New Metaheuristic Bat-Inspired Algorithm. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (eds.) NICSO 2010. SCI, vol. 284, pp. 65–74. Springer, Heidelberg (2010)