Parallelized Bat Algorithm with a Communication Strategy

Cheng-Fu Tsai, Thi-Kien Dao, Wei-Jie Yang, Trong-The Nguyen, and Tien-Szu Pan

Department of Electronics Engineering,
National Kaohsiung University of Applied Sciences, Taiwan
jvnkien@gmail.com

Abstract. The trend in parallel processing is an essential requirement for optimum computations in modern equipment. In this paper, a communication strategy for the parallelized Bat Algorithm optimization is proposed for solving numerical optimization problems. The population bats are split into several independent groups based on the original structure of the Bat Algorithm (BA), and the proposed communication strategy provides the information flow for the bats to communicate in different groups. Four benchmark functions are used to test the behavior of convergence, the accuracy, and the speed of the proposed method. According to the experimental result, the proposed communicational strategy increases the accuracy of the BA on finding the near best solution.

Keywords: Bat algorithm; swarm intelligence; numerical optimization.

1 Introduction

Swarm intelligence has been paid more attention from researchers, who work in the related field. Many swarm intelligence based algorithms have been developed and been successfully used to solve optimization problems in the engineering, the financial, and the management fields for recently years. For instance particle swarm optimization (PSO) techniques have successfully been used to forecast the exchange rates, the optimizing, [1-3], to construct the portfolios of stock, human perception [4-6], ant colony optimization (ACO) techniques have successfully been used to solve the routing problem of networks, the secure watermarking [7, 8], artificial bee colony (ABC) techniques have successfully been used to solve the lot-streaming flow shop scheduling problem [9], cat swarm optimization (CSO) [10] techniques have successfully been used to discover proper positions for information hiding [11].

Based on the algorithms in swarm intelligence, the idea of parallelizing the artificial agents by dividing them into independent subpopulations is introduced into the existing methods such as ant colony system with communication strategies [12], parallel particle swarm optimization algorithm with communication strategies [13], parallel cat swarm optimization [14], Island-model genetic algorithm [15], and parallel genetic algorithm [16]. The parallelized subpopulation of artificial agents increases the accuracy and extends the global search capacity than the original structure.

A. Moonis et al. (Eds.): IEA/AIE 2014, Part I, LNAI 8481, pp. 87–95, 2014.

The parallelization 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 concept of parallel processing is applied to Bat algorithm and a communication strategy for parallel BA is proposed.

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

2 Metaheuristic Bat-Inspired Algorithm

In 2010, Xin-SheYang proposed a new optimization algorithm, namely, Bat Algorithm or original Bat Algorithm (oBA), based on swarm intelligence and the inspiration from observing the bats [17] . Original BA simulates parts of the echolocation characteristics of the micro-bat in the simplicity way. It is potentially more powerful than particle swarm optimization and genetic algorithms as well as Harmony Search. The primary reason is that BA uses a good combination of major advantages of these algorithms in some way. Moreover, PSO and harmony search are the special cases of the Bat Algorithm under appropriate simplifications. Three major characteristics of the micro-bat are employed to construct the basic structure of BA. The used approximate and the idealized rules in Xin-SheYang's method are listed as follows:

All bats utilize the echolocation to detect their prey, 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 that the micro-bat sends a fixed frequency *fmin* with a variable wavelength λ and the loudness A_0 to search for prey.

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 ∈ [0, 1], 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.

In Yang's method, the movement of the virtual bat is simulated by equation (1) – equation (3):

$$f_i = f_{min} + (f_{max} - f_{min}) * \beta \tag{1}$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_{best}) * f_i \tag{2}$$

$$x_i^t = x_i^{t-1} + \mathrm{v}_\mathrm{i}^\mathrm{t} \tag{3}$$

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. x_i denotes the location of the i^{th} bat in the solution space, v_i 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 (4):

$$x_{new} = x_{old} + \varepsilon A^t \tag{4}$$

where ε is a random number and $\varepsilon \in [-1, 1]$, and at represents the average loudness of all bats at the current time step. After updating the positions of the bats, the loudness A_i and the pulse emission rate r_i are also updated only when 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 (5) and equation (6):

$$A_i^{t+1} = \alpha A_i^t \tag{5}$$

$$r_i^{t+1} = r_i^0[1 - e^{-\gamma t}] \tag{6}$$

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

The process of oBA 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 near 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 Parallelized Bat Algorithm with a Communication Strategy

Several groups in a parallel structure are created from dividing the population into subpopulations to construct the parallel processing as having been presented in some previous methods, such as parallel Cat swarm optimization [14], parallel Particle swarm optimization algorithm with communication strategies [13], parallel Genetic algorithm [16], Island-model genetic algorithm [15], and Ant colony system with communication strategies [12]. Each of the subpopulations evolves independently in regular iterations. They only exchange information between subpopulations when the communication strategy is triggered. It results in the reducing of the population size for each subpopulation and the benefit of cooperation is achieved.

The parallelized BA is designed based on original BA optimization. The swarm of bats in BA is divided into G subgroups. Each subgroup evolves by BA optimization independently, i.e. the subgroup has its own bats and near best solution. The bats in one subgroup don't know the existence of other subgroups in the solution space. The total iteration contains R times of communication,where $R = \{R_1, 2R_1, 3R_1, ...\}$. Let g_p be the subgroup, where $g \in G$ and p is the index of the subgroup. If $t \cap R \neq \varphi$, k agents with the top k fitness in g_p will be copied to $g_{(p+1) \bmod G}$ to replace the same number of agents with the worst fitness, where t denotes the current iteration count, $p = 1,2,3,..., G$ and k is a predefined constant. The diagram of the parallelized BA with communication strategy is shown in figure 1.

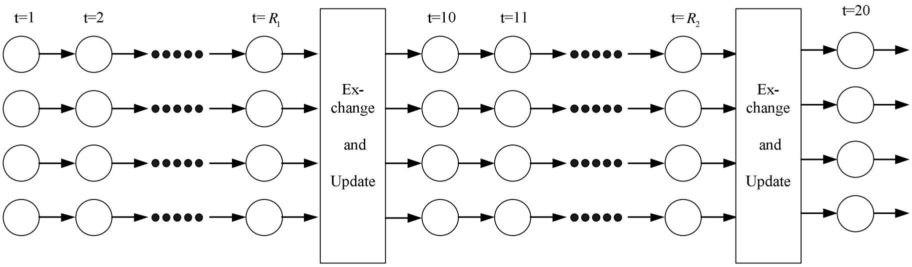

Fig. 1. The diagram of parallel BA with a communication strategy

1. **Initialization:** Generate bat population and divide them into G subgroups. Each subgroup is initialized by BA independently. Defined the iteration set R for executing the communication strategy. The N_j bats X_{ij}^T for the j^{th} group, i = 0, 1, …, N_j- 1, j = 0, 1, …, S -1, where S is the number of groups, N_j is the subpopulation size for the j^{th} group and t is the iteration number. Set t = 1.
2. **Evaluation:** Evaluate the value of $f(X_{ij}^T)$ for every bat in each group.
3. **Update:** Update the velocity and bat positions using Eqs. (1), (2) and (3).
4. **Communication Strategy:** Migrate the best bat among all the bats G^t to each group, mutate G^t to replace the poorer bats in each group and update G_j^t with G^t for each group 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 $f(G^t)$ and the best bat position among all the bats G^t.

4 Experimental Results

This section presents simulation results and compares the parallel BA with the original BA, both in terms of solution quality and in the number of function evaluations taken. Four benchmark functions are used to test the accuracy and the convergence of parallel BA. All the benchmark functions for the experiments are averaged over different random seeds with 25 runs. Let $X = \{x_1, x_2, ..., x_n\}$ be an n-dimensional realvalue vector, the benchmark functions are listed in equation (4) to equation (6). The goal of the optimization is to minimize the outcome for all benchmarks. The population size is set to 30 for all the algorithms in the experiments. The detail of parameter settings of BA can be found in [17].

$$f_1(x) = \sum_{i=1}^{N}[10 + x_i^2 - 10cos2\pi x_i \quad (7)$$

$$f_2(x) = 1 + \sum_{i=1}^{N}\frac{x_i^2}{4000} + \prod_{i=1}^{N} cos\frac{x_i}{\sqrt{i}} \quad (8)$$

$$f_3(x) = 20 + e - 20e^{-0.2\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}} - e^{\frac{\sum_{j=1}^{n}\cos(2\pi x_i)}{n}} \quad (9)$$

$$f_4(x) = \sum_{i=1}^{N} x_i^2 \quad (10)$$

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

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

Function	Initial range $[x_{min}, x_{max}]$	Total iteration
$f_1(x)$	[-5.12,5.12]	4000
$f_2(x)$	[-100,100]	4000
$f_3(x)$	[-50,50]	4000
$f_4(x)$	[-100,100]	4000

The parameters setting for both parallel BA and original BA: are the initial loudness $A_i^0 = 0.25$, pulse rate $r_i^0 = 0.5$ the total population size n = 30 and the dimension of the solution space $M = 30$, frequency minimum f_{min} = *the lowest of initial range function* and frequency minimum f_{max} = *the highest of initial range function.* Each function contains the full iterations of 4000 is repeated by different random seeds with 25 runs. The final result is obtained by taking the average of the outcomes from all runs. The results are compared with the original BA.

Comparison Optimizing Performance Algorithms: Table II compares the quality of performance and time running for numerical problem optimization between parallel Bat algorithm and original Bat algorithm. It is clearly seen that, almost these cases of testing benchmark functions for parallel BA are faster than original BA in convergence. It is special case with test function $f_1(x)$, the Rastrigin has the mean of value function minimum of total 25 seed runs is 207.0142 with average time running equal 6.8426 seconds for parallel Bat algorithm evaluation. However, for original Bat algorithm this value of function minimum of total 25 seed runs is 230.2515 with time running equal 6.2785seconds in same executing computer. The average of four benchmark functions evaluation of minimum function 25 seed runs is 4.88E+04with average time consuming 24.6783 for original BA and 4.55E+04 with average time consuming 64.767 for parallel BA respectively.

Table 2. The comparison between oBA and cBA in terms of quality performance evaluation and speed

Function	Performance evaluation		Time running evaluation (seconds)	
	Original BA	***Parallel BA***	***Original BA***	***Parallel BA***
$f_1(x)$	230.2515	207.0142	6.2785	6.8426
$f_2(x)$	4.45E+00	3.94E+00	7.0421	45.0864
$f_3(x)$	19.9624	19.9531	6.2729	7.2432
$f_4(x)$	4.86E+04	4.53E+04	5.0848	5.5948
Average value	**4.88E+04**	**4.55E+04**	**24.6783**	**64.767**

Figure number from 2 to 5 show the experimental results of four benchmark functions in 25 seed runs output with the same iteration of 4000.

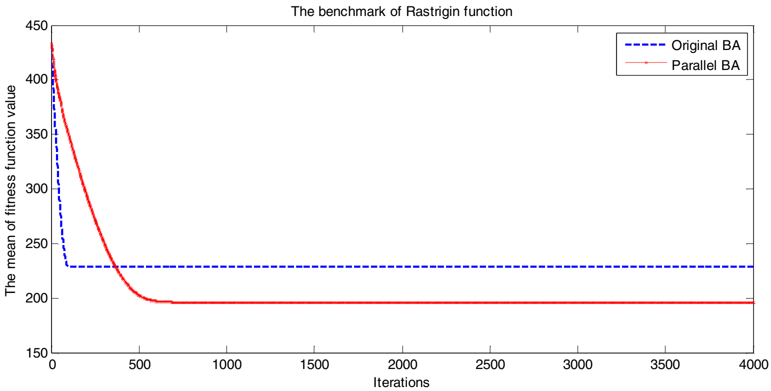

Fig. 2. The experimental results of Rastrigin function

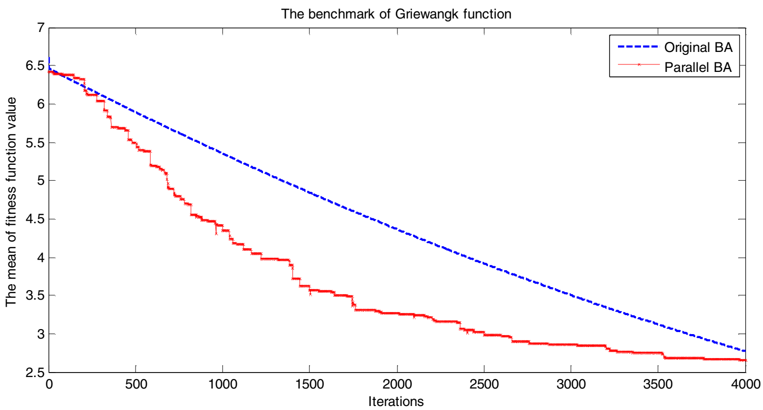

Fig. 3. The experimental results of Griewank function

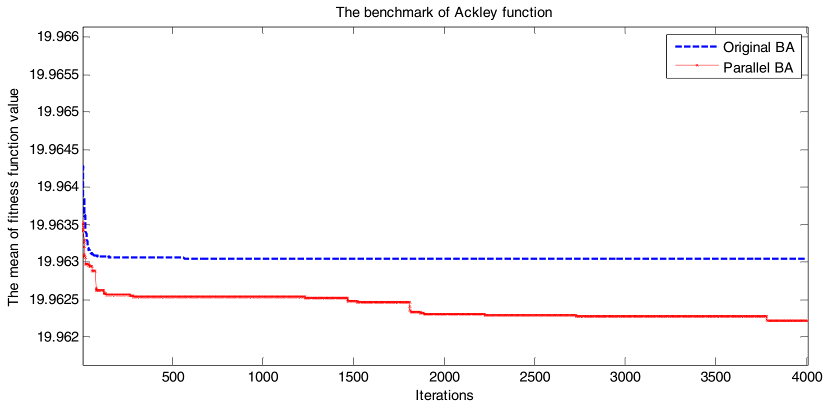

Fig. 4. The experimental results of Ackley function

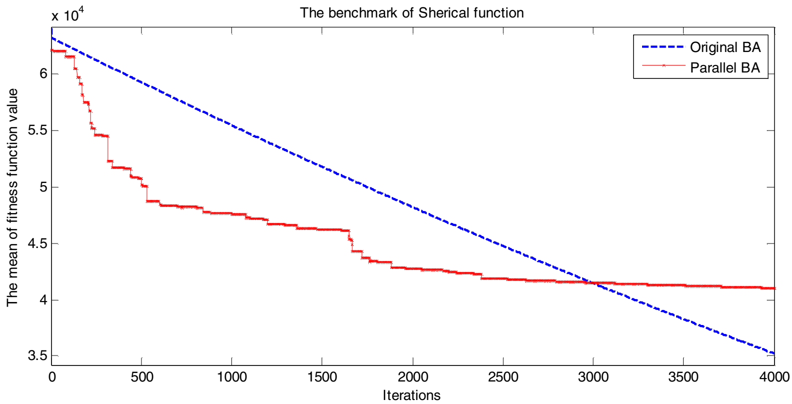

Fig. 5. The experimental results of Spherical function

According to the experimental result, parallel BA improves the convergence and accuracy of finding the near best solution about 8.3% than original BA.

5 Conclusion

In this paper, the parallelized Bat Algorithm (BA) optimization with a communication strategy is proposed for solving numerical optimization problems. The population bats are split into several independent groups based on the original structure of the BA, and the proposed communication strategy provides the information flow for the bats to communicate in different groups. In new proposed algorithm, the individual swarm poorer in among of subgroup of Bat algorithm is replaced with new individual swarm better from neighbor subgroups after each R_i iteration running. This feature is important for application problems characterized by parallel processing devices. The results of proposed algorithm on a set of various benchmark problems show that the proposed communicational strategy can be more convergence and increases the accuracy of the BA on finding the near best solution about 8.3%.

References

1. 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)
2. Hsu, C.-H., Shyr, W.-J., 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)
3. Chen, S.-M., Kao, P.-Y.: TAIEX forecasting based on fuzzy time series, particle swarm optimization techniques and support vector machines. Information Sciences 247, 62–71 (2013)
4. Jui-Fang, C., Shu-Wei, H.: The Construction of Stock_s Portfolios by Using Particle Swarm Optimization. In: Second International Conference on Innovative Computing, Information and Control, ICICIC 2007 (2007)
5. Parag Puranik, P.B., 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)
6. 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)
7. Pinto, P.C., et al.: Using a Local Discovery Ant Algorithm for Bayesian Network Structure Learning. IEEE Transactions on Evolutionary Computation 13(4), 767–779 (2009)
8. Khaled Loukhaoukha, J.-Y.C., 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)
9. Pan, Q.-K., et al.: A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. Inf. Sci. 181(12), 2455–2468 (2011)
10. Chu, S.-C., Tsai, P.-W.: Computational Intelligence Based on the Behavior of Cats. International Journal of Innovative Computing, Information and Control 3(1)(3), 8 (2006)
11. 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)
12. Chu, S.-C., Roddick, J.F., Pan, J.-S.: Ant colony system with communication strategies. Information Sciences 167(1-4), 63–76 (2004)
13. Chang, J.F., Chu, S.C., Roddick, J.F., Pan, J.S.: A parallel particle swarm optimization algorithm with communication strategies. Journal of Information Science and Engineering 21(4), 9 (2005)
14. Pei-Wei, T., et al.: Parallel Cat Swarm Optimization. In: 2008 International Conference on Machine Learning and Cybernetics (2008)

15. Whitley, D., Rana, S., Heckendorn, R.B.: The Island Model Genetic Algorithm: On Separability, Population Size and Convergence. Journal of Computing and Information Technology 1305/1997, 6 (1998)
16. Abramson, D.A., Abela, J.: A parallel genetic algorithm for solving the school timetabling problem. In: Proc. of Appeared in 15 Australian Computer Science Conference, Hobart, Australia, p. 10 (1991)
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)