Baldwin Effect based Particle Swarm Optimizer for Multimodal Optimization

Ji Qiang Zhai
College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin, 150040, P. R. China
School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, P. R. China
Email: zhaijiqiangnfus@163.com

Ke Qi Wang
College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin, 150040, P. R. China

***Abstract*—Particle Swarm Optimization (PSO) is an effective optimal technique. However, it often suffers from being trapped into local optima when solving complex multimodal optimizing problems due to its inefficient exploiting of feasible solution space. This paper proposes a Baldwin effect based learning particle swarm optimizer (BELPSO) to improve the performance of PSO when solving complex multimodal optimizing problems. This Baldwin effect based learning strategy utilizes the historical beneficial information to increase the potential search range and retains diversity of the particle population to discourage premature. On the other hand, the exemplars provided by Baldwin effect based learning strategy can flatten out the fitness landscape closing to optima and hence guide the search path towards optimal region. Experimental simulations show that BELPSO has a wider search range of feasible solution space than PSO. Furthermore, the performance comparison between BELPSO and amount of population based algorithms on sixteen well-known test problems shows that BELPSO has better performance in quality of solution.**

***Index Terms*—Particle Swarm Optimization; Baldwin effect; Swarm intelligence; Population based algorithm; Computational intelligence**

I. INTRODUCTION

Optimization has been an active area of research for several decades. Since many practical problems which arise in almost every field of science, engineering and business can be formulated to multi-modal optimization problems, many algorithms and approaches are presented to solving these complex optimization problems.

The particle swarm optimizer (PSO), proposed by Kennedy and Eberhart [1, 2] in 1995, has gained growing interesting and has been widely applied to deal with numerous engineering applications [3, 4]. However, the performance of PSO greatly depends on its parameters and it often suffers from being trapped in local optima. Thus, a large amount of variants of PSO have been proposed for improving its performance. Shi and Eberhart first proposed a linearly decreasing inertia weight during the process of search [5], and designed fuzzy methods to nonlinearly change the inertia weight [6]. In Ref. [7], a self adaptive approach of changing each particle's inertia weight is proposed. Clerc and Kennedy [8] introduced a constriction factor in PSO to guarantee the convergence and improve the convergent speed.

Improving the performance of PSO by combining PSO with other search techniques has been an active research direction. The selection operator of evolutionary algorithms has been used in PSO to preserve the best particles and thus to ensure the convergence [9]. Besides, the mutation operator has also been used for retain the swarm diversity so that to avoid tripping into a local optimum [10]. Bergh and Engelbrecht proposed a cooperative approach by searching one dimension separately by particles and combine the results together [11]. CLPSO [12] introduces a comprehensive learning strategy into PSO algorithms, whereby all other particles' historical best information is used to update a particle's velocity.

Baldwin effect is a nature phenomena where individuals will survive longer through learning from others to fit the environment better and thus to improve the entire evolution process [13]. An advantage of Baldwinian learning is that it can flatten out the fitness landscape around the optimal regions, and hence help find the global optimum even in a dynamic environment [14, 15]. The first work of exploring the Baldwin effect can be traced back to 1980s when Hinton and Nowlan [16] proposed a hybrid algorithm combining a genetic algorithm and a Baldwinian learning strategy for developing simple neural networks. Baldwinian learning has been gained increasing interesting, and hence numbers of further investigations and models based on Baldwin effect were presented [17-20].

This paper aims to alleviate the premature of PSO algorithm and to further improve the performance in quality of solution on complex multi-modal problems, based on Baldwin effect, a novel Baldwin effect based learning particle swarm optimizer (BELPSO) is presented for improving the performance when solving complex multi-modal problems. This learning strategy utilizes the historical beneficial information to increase the search

doi:10.4304/jcp.7.9.2114-2119

range and retains diversity of the particles to discourage premature. On the other hand, the exemplars provided by Baldwin effect based learning can flatten out the fitness landscape approaching optimum and hence guide the search path towards optimal region. Experimental results show that BELPSO has good performance in solving most of the test problems and is an effective algorithm for complex multi-modal problem optimization.

II. Baldwin Effect Based Learning Particle Swarm Optimizer

A. Baldwin Effect Based Learning Strategy for Velocity Updating

This new learning strategy not only flattens out the optimal region but also keeps the diversity of the particle population. The following velocity updating equation is used in BELPSO

$$V_i^d \leftarrow w(k) * V_i^d + c * rand_i^d * (pbest\_baldwin_i^d - X_i^d) \quad (1)$$

where c is acceleration constants. $rand_i^d$ is random number selected from the range [0, 1]. $w(k)$ is the inertia weight at k th generation. $pbest\_baldwin_i^d$ is the dth dimension of the ith particle $pbest\_baldwin_i$, $pbest\_baldwin_i$ is obtained as follows:

$$pbest\_baldwin_i = \begin{cases} pbest_i \quad if\ suc(i) <= LG \\ pbest_i + s * \sum_{<j,l> \in randgen(ps)} \begin{pmatrix} \sup(pbest_j, pbest_l) \\ -\mathrm{infe}(pbest_j, pbest_l) \end{pmatrix} \\ \quad if\ suc(i) > LG \end{cases} \quad (2)$$

where $pbest_i = (pbest_i^1, pbest_i^2, ..., pbest_i^D)$ denotes i th particle's $pbest$, ps is the size of particle population, $randgen(ps)$ means generating a set consisting of a comparison shows that BELPSO performs better than CSA and DEA on most of the test functions with D=10.random number of unique tuples from $1, 2, ..., ps$, i.e. (<1,2>,<3,4>,<5,6>) from $1, 2, 3, ..., 6$, $\sup(x, y)$ and $\mathrm{infe}(x, y)$ are the superior and inferior of x and y, respectively. Take minimizing problem for example, $\sup(x, y)$ and $\mathrm{infe}(x, y)$ are the minimum and maximum of x and y respectively. $s \in [0,1]$ is the Baldwin learning strength. $suc(i)$ defines the successive generation without improvement of i th particle's $pbest$, LG expresses the learning gap which controls the local search ability by minimizing the time wasted in poor search direction to some extent. When $suc(i) > LG$, the particle executes a Baldwin learning to alter the search space and thereby provides the good exemplars towards the optimal regions.

B. The Proposed BELPSO

The novel algorithm is implemented as Fig. 1. In BELPSO, not only the particle's own $pbest$ but also all particles' $pbest$ s and some of neighbors can potentially be the learning exemplars, while only particle's own $pbest$ and $gbest$ be the exemplars in simple PSO. Besides, there is only one exemplar $pbest\_baldwin$ to be learned in every generation in BELPSO, in stead of the two exemplars $pbest$ and $gbest$ in simple PSO.

Step 1 BELPSO initialization. For each particle i in the population, randomly generate the X_i and V_i, evaluate $f(X_i)$ to initialize the $pbest_i$, set k=1.

Step 2 Repeat until the termination criterion is satisfied. If the termination criterion is satisfied, then stop iteration and output the best solution pS such that $f(pS) \le f(pbest_i)$; else set k=k+1 and go to Step 2.1

Step 2.1 If $i \ge ps$, then set $i = 0$ and go to Step 2; else set $i = i + 1$ and go to Step 2.2.

Step 2.2 If $suc(i) > LG$, then go to Step 2.3; else go to Step 2.4.

Step 2.3 Baldwin effect based learning. Update $pbest\_baldwin_i$ as statement of Eq. (2), then go to Step 2.5.

Step 2.4 Update $pbest\_baldwin_i = pbest_i$.

Step 2.5 Firstly, update V_i^d as Eq. (1), and then apply the following equation to control the flying step of particle i.

$$V_i^d \leftarrow \min(V_{\max}^d, \max(-V_{\max}^d, V_i^d))$$

where $V_{\max}^d$ is a positive constant value specified by the user, in our study, it is set to twenty percent of the maximum search range of each dimension.

Step 2.6 Update X_i^d.

Step 2.7 Evaluate $f(X_i)$ if X_i is in the feasible searching range; else go to Step 2.1.

Step 2.8 If $f(X_i) \le f(pbest_i)$, then update $pbest_i = X_i$ and set $suc(i)$ =0; else set $suc(i)$ = $suc(i)$ +1, and then go to Step 2.1.

Figure 1. Flowchart of the BELPSO algorithm

III. Simulation Experiments

In this section, we use experiments to evaluate the performance of BELPSO by solving sixteen function optimization problems [21-24].

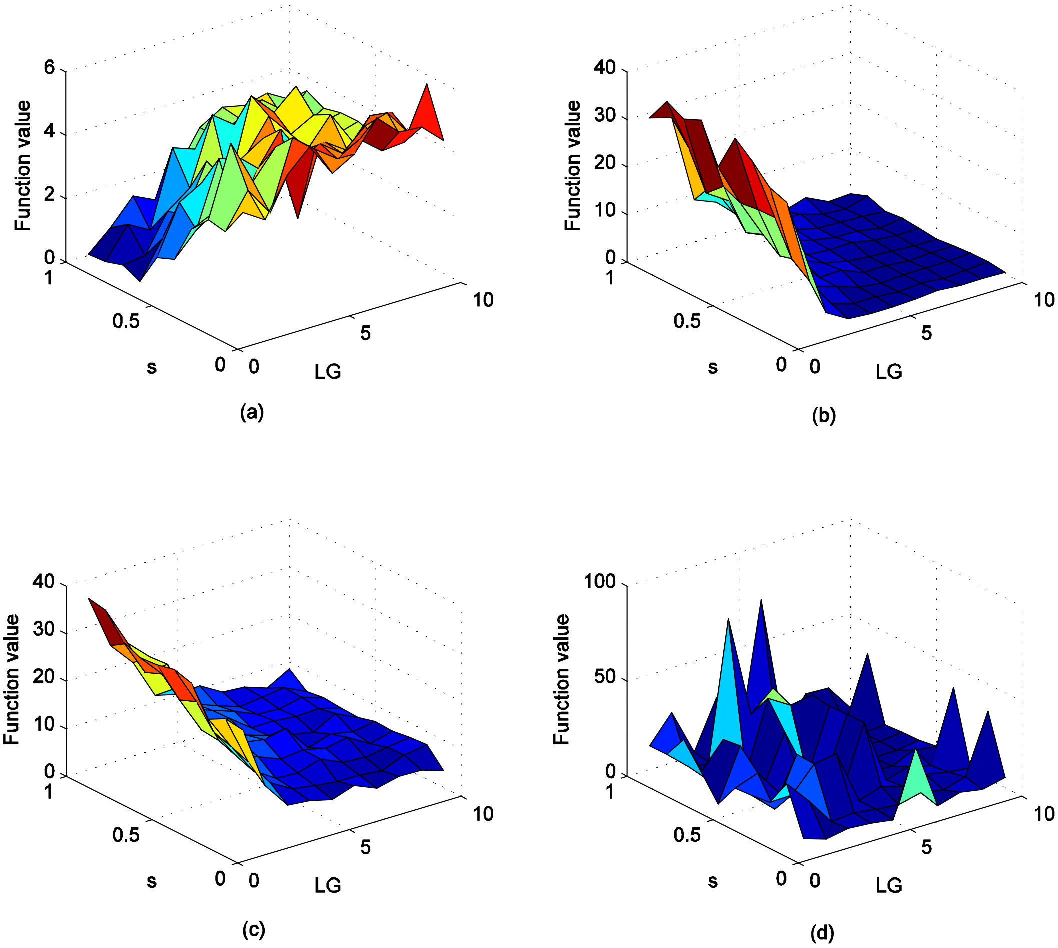

Figure 2. Sensitivity in relation to learning parameters of *LG* and *s*, (a), (b), (c), (d) are BELPSO's function values to *LG* and *s* in optimizing f_2, f_6, f_{12}, and f_{15}, respectively

A. Sensitivity in Relation to Parameters

We investigate the effects of the main parameters about Baldwin effect based learning of BELPSO by applying it to solve the unimodal function, unroated multi-modal function, rotated multi-modal function and composition function with various learning gap *LG* and learning strength s.

The experimental results of BELPSO in optimizing f_2, f_6, f_{12}, and f_{15} with learning gap *LG* increased from 1 to 10 in steps of 1 and the learning strength *s* from 0.1 to 1 in steps of 0.1 are shown in Fig. 2. The values of other parameters are as follows: the problem dimension D is 10, the population size is set at 10 and the maximum FEs is set at 30000. The inertia weight at k th generation $w(k)$ is as follows:

$$w(k) = w_0 * \frac{(w_0 - w_1) * k}{\max\_gen} \qquad (3)$$

where k denotes the generation, $\max\_gen$ is the maximum generations, it is set at 3000 in our study, w_0 and w_1 are specified to 0.9 and 0.4, respectively, which is the same as [12].

Fig. 2 shows the statistical average values obtained from 30 independent runs. From Fig. 2 we observe that learning gap and learning strength can influence the performance of BELPSO. For f_2, we obtained a faster convergence velocity and better results when LG is set at 1 and *s* is set at 1. For f_6, f_{12}, and f_{15}, too small values of LG make the algorithm trap into local optima, better results were obtained when LG is 6~10 and *s* is 0.1~ 0.6. The results demonstrate that too much learning or too less learning may discourage the convergence speed when deal with complex multi-modal problems, which complies with Baldwin effect [16]. Hence, in our study, the learning gap and learning strength are set at 7 and 0.5 respectively for all test functions.

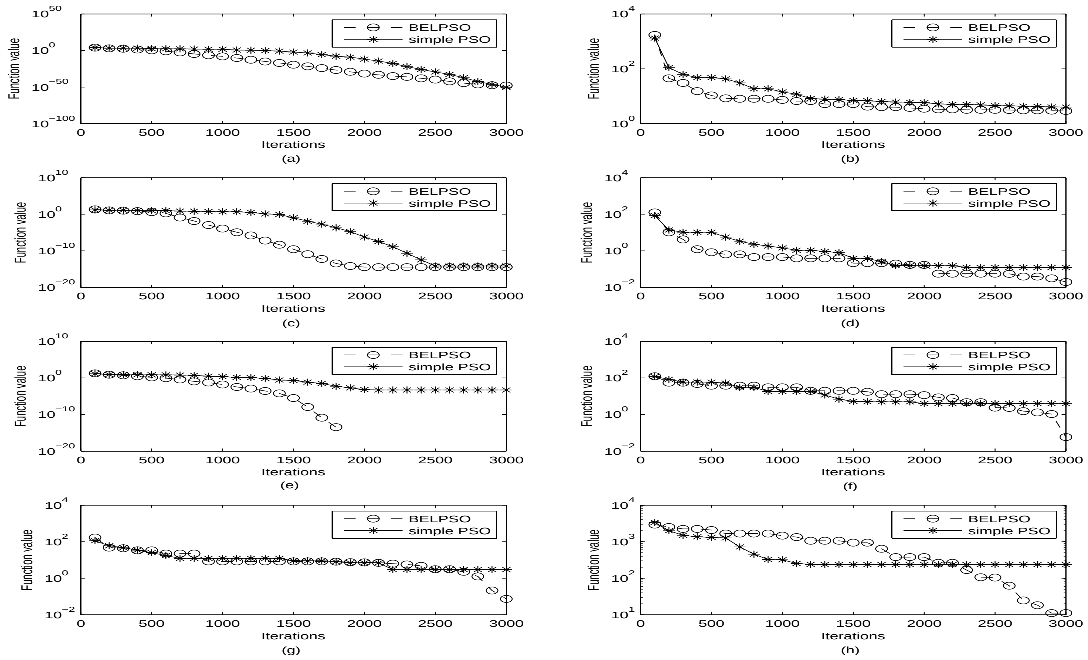

Figure 3. The convergence graph of BELPSO and simple PSO to iterations on all test functions with D=10. (a), (b), (c), (d), (e), (f), (g) and (h) are results of the two algorithms in optimizing functions 1, 2, 3, 4, 5, 6, 7 and 8, respectively

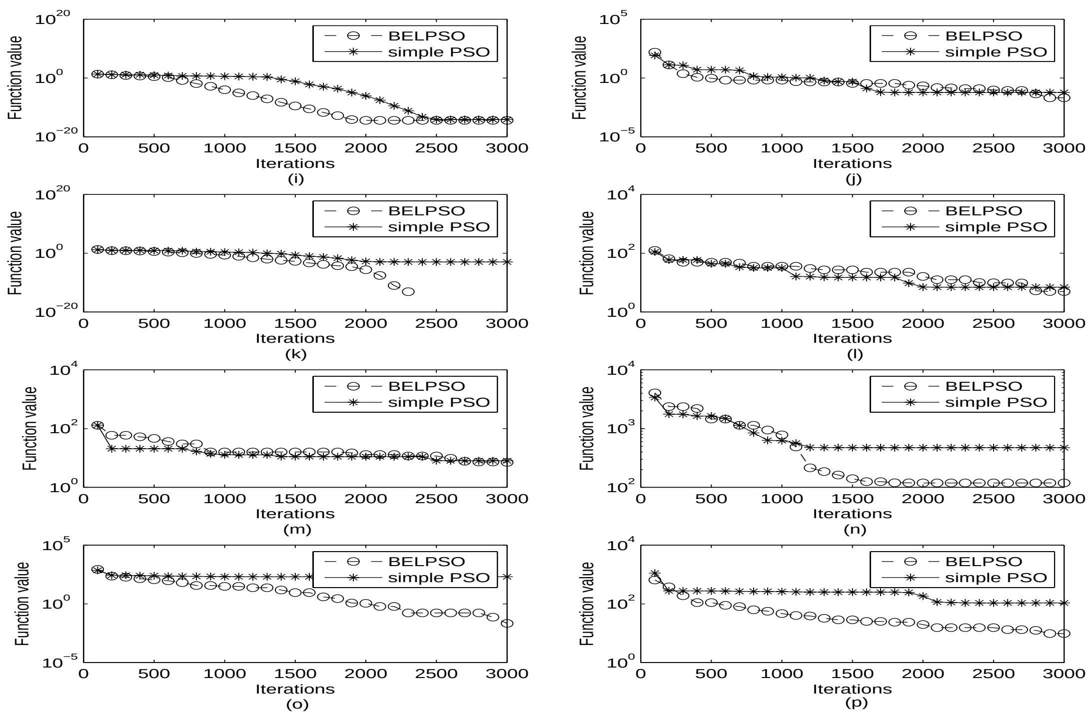

Figure 3. (Continued) The convergence graph of BELPSO and simple PSO to iterations on all test functions with D=10. (i), (j), (k), (l), (m), (n), (o) and (p) are results of the two algorithms in optimizing functions 9, 10, 11, 12, 13, 14, 15 and 16, respectively

B. The Statistical Results on the Test Functions

Via the analysis of the sensitivity of the BELPSO, we set the main parameters as follows: The learning gap is 7, the learning strength is 0.5, and w is the same as above. When solving the 10-D functions, the population size is set at 10 and the maximum FEs is set at 30000. When solving the 30-D functions, the population size is set at 30 and the maximum FEs is set at 2000000. All experiments were run 30 times.

Fig. 3 illustrates the comparison of BELPSO and simple PSO's convergence characteristics on all of test functions, where the results are the median value of 20 independent runs with D=10. The parameters of BELPSO and simple PSO are same as above. Fig. 3 shows that BELPSO outperform the simple PSO on almost all of the test problems. Since Baldwin effect based learning can provide wide potential search spaces to maintain population diversify and thus to alleviate the premature, the better results are obtained when apply this Baldwin effect based learning strategy to PSO on complex multi-modal problems, especially on the composition problems. In addition, with the ability of smooth out the optimal region of Baldwin effect based learning, the quality of the results of BELPSO is higher than that of simple PSO.

C. Performance Comparisons between BELPSO with Some of Heuristic Population based Algorithms

In this section, we compare the performance of BELPSO with CSA [25] and DEA [26]. The reported results of CSA and DEA are obtained from [27] for direct comparison.

The comparisons of BELPSO with CSA and DEA on 10-D functions are shown in Table Ⅰ. Table Ⅰ indicates that BELPSO surpasses CSA and DEA on functions 1, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13 and 14 (twelve out of sixteen functions), especially improve the results on functions 1, 3, 5 and 9 significantly. This comparison shows that BELPSO performs better than CSA and DEA on most of the test functions with D=10.

Table I.
MEAN AND STANDARD DEVIATION VALUES OBTAINED BY CSA, DEA AND BELPSO ON ALL TEST FUNCTIONS WITH D=10

	Group A	Group A	Group B	Group B
	f_1	f_2	f_3	f_4
CSA	3.54e+000 ± 1.53e+000	1.69e+000 ± 0.63e+000	1.83e+000 ± 0.36e+000	0.91e+000 ± 0.10e+000
DEA	9.55e-013 ± 1.32e-012	**1.02e-002 ± 8.60e-003**	4.80e-017 ± 3.58e-007	4.30e-001 ± 7.19e-002
BELPSO	**1.03e-041 ± 2.40e-041**	3.53e+000 ± 6.29-001	**3.55e-015 ± 4.32e-31**	**2.70e-002 ± 1.04e-002**
	Group B	Group B	Group B	Group B
	f_5	f_6	f_7	f_8
CSA	1.17e+000 ± 0.13e+000	1.92e+000 ± 0.55e+000	1.90e+000 ± 0.61e+000	7.66e+000 ± 3.82e+000
DEA	7.31e-004 ± 2.87e-004	2.20e+001 ± 4.60e+000	1.37e+001 ± 2.08e+000	**1.28e-002 ± 4.45e-002**
BELPSO	**0 ± 0**	**2.38e-001 ± 4.04e-001**	**1.70e-001 ± 4.10e-001**	1.59e+002 ± 1.44e+002
	Group C	Group C	Group C	Group C
	f_9	f_{10}	f_{11}	f_{12}
CSA	2.45e+000 ± 0.36e+000	0.90e+000 ± 0.11e+000	2.71e+000 ± 0.57e+000	1.61e+001 ± 3.66e+001
DEA	5.98e-007 ± 3.66e-007	5.37e-001 ± 9.71e-002	1.10e-002 ± 4.50e-003	3.62e+001 ± 5.49e+001
BELPSO	**4.74e-015 ± 1.83e-015**	**6.84e-002 ± 4.45e-002**	**3.55e-005 ± 4.21e-005**	**5.77e+000 ± 1.88e+000**
	Group C	Group C	Group D	Group D
	f_{13}	f_{14}	f_{15}	f_{16}
CSA	1.05e+001 ± 2.99e+000	4.81e+002 ± 1.43e+002	6.05e+000 ± 3.51e+000	**3.20e+000 ± 9.64e+000**
DEA	2.10e+001 ± 4.24e+001	9.00e+002 ± 4.47e+002	**1.73e-012 ± 2.55e-012**	1.20e+001 ± 2.99e+001
BELPSO	**7.25e+000 ± 7.51e-001**	**1.98e+002 ± 1.42+002**	4.56e+000 ± 7.89e+000	1.39e+001 ± 1.11e+001

Ⅳ. CONCLUSIONS

By incorporating a novel Baldwin effect based learning strategy into particle swarm optimizer, a novel algorithm termed BELPSO, is presented for solving complex multimodal problems. BELPSO was executed to solve sixteen test problems. The performance comparisons of the BELPSO with other variants of PSO, and several population based algorithms including CSA and DEA indicated that BELPSO perform better on most of the multi-modal test functions.

REFERENCES

[1] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39-43.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.

[3] S. Meshoul and T. Al-Owaisheq, "QPSO-MD: A Quantum Behaved Particle Swarm Optimization for Consensus Pattern Identification," in *Computational Intelligence and Intelligent Systems*. vol. 51, 2009, pp. 369-378.

[4] P. M. Pradhan, V. Baghel, G. Panda, and M. Bernard, "Energy Efficient Layout for a Wireless Sensor Network using Multi-Objective Particle Swarm Optimization," in *Advance Computing Conference, 2009. IEEE International*, 2009, pp. 65-70.

[5] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings.* New York, NY, USA, 1998, pp. 69-73.

[6] S. Yuhui and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII. 7th International Conference, EP98. Proceedings, 25-27 March 1998*, Berlin, Germany, 1998, pp. 591-600.

[7] B. Liu, L. Wang, Y. H. Jin, F. Tang, and D. X. Huang, "Improved particle swarm optimization combined with chaos," Chaos Solitons & Fractals, vol. 25, pp. 1261-1271, 2005.

[8] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," IEEE Transactions on Evolutionary Computation, vol. 6, pp. 58-73, 2002.

[9] P. J. Angeline, "Using selection to improve particle swarm optimization," in *Proceedings IEEE World Congress on Computational Intelligence.*, 1998, pp. 84-89.

[10] T. K. R. Morten Lovbjerg , Thiemo Krink, “Hybrid Particle Swarm Optimiser with Breeding and Subpopulations”, in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2001, pp. 469-476.

[11] F. van den Bergh and A. P. Engelbrecht, "A Cooperative approach to particle swarm optimization," IEEE Transactions on Evolutionary Computation, vol. 8, pp. 225-39, 2004.

[12] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," IEEE Transactions On Evolutionary Computation, vol. 10, pp. 281-295, 2006.

[13] J. M. Baldwin, "A New Factor in Evolution " The American Naturalist, vol. 30, No. 354 pp. 441-451, 1896.

[14] D. Whitley, V. S. Gordon, and K. Mathias, "Lamarckian evolution, the Baldwin effect and function optimization," in *Parallel Problem Solving from Nature - PPSN III International Conference on Evolutionary Computation The Third Conference on Parallel Problem Solving from Nature*, Berlin, Germany, 1994, pp. 6-15.

[15] R. W. Anderson, "Learning and evolution: A quantitative genetics approach," Journal of Theoretical Biology, vol. 175, pp. 89-101, 1995.

[16] G. E. Hinton and S. J. Nowlan, "How learning can guide evolution," in *Adaptive individuals in evolving populations: models and algorithms*, 1996, pp. 447-454.

[17] R. K. Belew, “When Both Individuals and Populations Search: Adding Simple Learning to the Genetic Algorithm”, in *Proceedings of the 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers Inc., 1989, pp. 34-41.

[18] J. A. Bullinaria, "Exploring the Baldwin effect in evolving adaptable control systems," in *Connectionist Models of Learning, Development and Evolution*, 2000, pp. 231-242.

[19] I. Harvey, "Puzzle of the persistent question marks. A case study of genetic drift," Australian Electronics Engineering, vol. 27, pp. 15-15, 1994.

[20] J. Huang and H. Wechsler, "Visual routines for eye location using learning and evolution," IEEE Transactions on Evolutionary Computation, vol. 4, pp. 73-82, 2000.

[21] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," IEEE Transactions on Evolutionary Computation, vol. 3, pp. 82-102, 1999.

[22] C.-Y. Lee and X. Yao, "Evolutionary Programming Using Mutations Based on the Levy Probability Distribution," IEEE Transactions on Evolutionary Computation, vol. 8, pp. 1-13, 2004.

[23] Z. Tu and Y. Lu, "A robust stochastic genetic algorithm (StGA) for global numerical optimization," IEEE Transactions on Evolutionary Computation, vol. 8, pp. 456-470, 2004.

[24] S. C. Esquivel and C. A. C. Coello, "On the use of particle swarm optimization with multimodal functions," in *Evolutionary Computation*, 2003, pp. 1130-1136.

[25] L. N. de Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," IEEE Transactions on Evolutionary Computation, vol. 6, pp. 239-251, 2002.

[26] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," Journal of Global Optimization, vol. 11, pp. 341-359, 1997.

[27] Z. Lining, G. Maoguo, J. Licheng, and Y. Jie, "Improved Clonal Selection Algorithm based on Baldwinian learning," in *IEEE World Congress on Computational Intelligence*, 2008, pp. 519-526.