2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

The Enhanced Vector of Convergence for Particle Swarm Optimization Based on Constrict Factor

Wei Zhang, *Student Member IEEE*, Yanan Gao and Chengxing Zhang

***Abstract*—The Particle Swarm Optimizer is used very widely for unimodal and multi-modal optimization problems. Recently, most of variant PSOs are combing several evolutionary strategies in order to achieve a better performance on Benchmark functions, and even for shifted, rotated, or composite functions. In this paper, a new method known as Enhanced Vector of Convergence is proposed and combined with constrict factor to improve the convergence performance of Particle Swarm Optimizer. In experimental study, other 5 variant Particle Swarm Optimizers are compared, and acceptance rate, *t-Test* are used for further evaluation. The results indicate that the Enhance Vector of Convergence can significantly improve the accurate level of Particle Swarm Optimizer.**

I. Introduction

Particle Swarm Optimization (PSO) was born in 1995, which is a new algorithm of swarm intelligence, mimicking the behavior of birds' flock [1], [2]. Although PSO is considered as one of branch of Evolutionary Algorithm (EAs), it takes a simple principle to update all particles rather than using complicated classical evolutionary operation known as selection, crossover and mutation. The attractive advantage of PSO is easy to implement, and there is much faster convergence velocity of PSO than Genetic Algorithm (GA). Modified PSOs inspire more and more investigators; the variant PSOs are utilized in various engineering optimization, for example, in solving and optimal controller design. Despite the differential equations and recursion equations cannot explain the principle of PSO, in practical application, PSO is definitely proved as an efficient tool. Swarm communication makes the population converge faster. So, when PSO was introduced, and it has produced many representative results. But current composite functions are introduced, which is more difficult for EAs to find its global optimum. The future test functions are more and more challenging. Most investigators hope that a simple and efficient PSO will be proposed, and it can successfully find the global optimum in Benchmark functions, shifted, rotated, biased and even composite functions with a very higher accurate level and cost less time, moreover it is easy to implement.

In last decades, to provide more intelligence to the variant PSOs, the combined several evolutionary strategies are utilized. Just like Adaptive Particle Optimization (APSO), it contains four studying strategies such as convergence, exploitation, exploration and jumping out [3]. By following the corresponding studying strategies, the APSO can adaptively adjust inertia weight and accelerate coefficients to find the optima. Actually, a regulation used in the APSO is a supervisor to distinguish the status of the swarm. The Frankenstein's PSO uses a topology strategy and adaptive inertia weight. The novel variant PSOs named Self-Learning Particle Swarm Optimization (SLPSO) classify the status of particles as four strategies, which are same as APSO. But SLPSO update the velocity and position of particles following on a different principle. The SLPSO does not take an advantage of any supervisor, the roulette wheel is used to assume the current status of the swarm, and then a specific update is doing based on different operation [4]. In 2013, Adaptive Particle Swarm Optimization with Multiple Adaptive Methods (APSO-MAM) finds the global optimum by controlling three parameters and adaptive inertia weight. The APSO-MAM can present a very high accurate level in these shifted, rotated functions [5], [6]. Since there is not any variant PSO can performs perfect on every Benchmark functions, the multiple learning strategies are more and more important and useful.

Based on our empirically evaluation and a large number of experiments, a novel variant PSO known as Particle Swarm Optimizer Based on Constrict Factor and Enhanced Vector of Convergence (CFPSO-EVC) is proposed. Since Particle Swarm Optimizer Based on Constrict Factor (CFPSO) can avoid balancing the global and local search by introduction of constrict factor, which is a nonlinear combination of two accelerate coefficients widely used in the traditional PSO and other variant PSOs. Compared with other anterior works, there are several distinctive mechanism are included in the CFPSO-EVC. Firstly, since inertia weight can balance global search and local search in PSO, an independent linear deceasing inertia weight, based on the iterations, is not very effective. Some

W. Zhang is with the Department of Computer Science, University of Georgia, Athens, GA 30602 USA (e-mail: waffenzw@uga.edu).

Y. Gao is with Lanzhou University of Technology, Lanzhou, 730050, China. She is now with the Department of Control Engineering and Theory, (e-mail: gynan87@gmail.com).

C. X. Zhang is with the Lanzhou University of Finance and Economics, Lanzhou 730020 China on (e-mail: zhangcx@lzcc.edu.cn).

This work was supported by 2012 Western Boundary and Areas Planning Foundation of China Ministry Education under Grant No. 12XJA910002.

variant PSOs with random inertia weight do not solve a complicated multi-modal problem either. A variant inertia weight in accordance with fitness value is proposed, and it is combined with a presented time-varying accelerate coefficients. Furthermore, a novel conception named Enhancement Vector of Convergence (EVC) is presented. The EVC can follow a certain known probability to replace the current position of particles, based on minimum fitness value. The EVC can reduce the iterations, with the same accurate level. However, the EVC is not a stable factor introduced in CFPSO-EVC. For PSOs, it is important to make sure all particles must be within the given range.

The rest of our paper is organized as below. The second section will introduce eminent and current relevant work on the variant PSOs. In third section, the details of CFPSO-EVC are described. Then, the fourth part is organized by the experimental comparison, which will be displayed in order to support the effectiveness of CFPSO-EVC. And the analysis and comparison will be made. The fifth section is going to make a conclusion.

II. Variant Particle Swarm Optimizer

A. Literature Review

The canonical PSO is a stochastic algorithm based on the swarm intelligence, and the main difference from other EAs is the particle can learn interactive information from others. In contrast, the EAs as GA or Differential Evolution (DE) are not able to learn other so-called chromosome. A potential solution represents a particle in the given range. The initialization is the generation of a group of particles, and these random particles can obtain others' information of position, moving toward current best position. PSO is likewise a flock of birds are pursuing their food.

In 1999, Shi and Eberhart investigated the variant of inertia weight [7]. A changed inertia weigh according to iterations was able to provide a balance to global and local search. The linear decreasing inertia weight was proposed. Lately, a fuzzy logic for variant inertia weight was established [8]. The principle of adjusting inertia weight is as follow:

Several years later, based on analysis of convergence behavior of PSO, the PSO with constrict factor was presented. The CFPSO utilized a constrict factor to preserve the present velocity of particles rather than classical inertia weight and accelerate coefficients. Some investigators argued that two principles for updates the velocity is very similar [9], the CFPSO has a faster convergence speed and is more accurate than simple PSO; since the CFPSO used a combination mechanism of two accelerate coefficients, without consideration of global search and local search. It is a very difficult issue that an algorithm can apply global search or local search model in a right time. The global search model update based on the entire swarm, and it is rapid, but it is vulnerable to trap in a local optimum. However, the local search model is slow, but it updates with several elements of a whole swarm, and it is not easy to get stuck in local optimum. The CFPSO uses a principle to update as below:

The FIPS was a modified PSO to contain neighborhood information for update [10]. Moreover, the FIPS examined the topology of particles, which was vital for the PSOs. In 2006, J. J. Liang and P. N. Suganthan reported CLPSO, which used a comprehensive learning mechanism that can construct current position by tournament selection [11]. Actually, it intentionally contained some worse positions, whose fitness value was much larger. It was thought as an efficient way to preserve the diversity of swarm. The CLPSO was really not vulnerable to trap in local optimum, and it was usually used as an exemplar for comparison.

Three years later, another two variant PSOs were presented. There are Frankenstein's Particle Swarm Optimizer (FPSO) and Adaptive Particle Swarm Optimizer (APSO) respectively. The FPSO embraced several advantages of previous variant PSOs [12]. Firstly, it considered the influence of topology; furthermore, the FPSO took the linear decreasing inertia weight; finally, the FPSO did not simply update all particles following a canonical regulation; it used a fully informed model which was introduced by FIPS. And, APSO used a multiple learning strategies to update. A regulation on reorganization current status of particles was designed. For anterior investigations, the inertia weight and accelerate coefficients were set to pursue a single model, just like the linear decreasing. However, in APSO, there were 4 kinds of adjust for these parameters. The APSO was able to reduce or increase each parameter, when it was necessary. The performance of APSO was more accurate, compared with other previous work on variant PSOs [3]. In TRIBES-PSO, the topology was adjusted according to the swarm behavior and the strategies of replacement were selected based on performances of the particles [13]. In 2012, SLPSO was presented as another variant PSO using multiple learning strategies. It updates the positions and velocity of particles following different 4 strategies, which can be decided by roulette. Compared with APSO, the SLPSO used four different regulations for update as: Exploitation, Jumping Out, Exploration, and Convergence. These four kinds of principle is helpful to SLPSO update the particles according to different situations. The aforementioned four principles are similar to the APSO, but the method for update is entirely different. Only APSO can adjust these three parameters, but the SLPSO utilizes four different regulations to update. The SLPSO performs better than other previous variant PSO on shifted, rotated, and shifted, rotated Benchmark functions. Moreover, SLPSO was accurate on higher dimensional experiment. Generally, the higher dimension of functions influent the performance of all PSOs seriously. In [4], SLPSO was evaluated under 30, 50 and 100 dimensions.

An intelligent augmented PSO with multiple adaptive methods (PSO-MAM) was proposed and was demonstrated to be efficient for most Benchmark functions. But, the performance of PSO-MAM heavily depended on the settings of three parameters: the two accelerate coefficients and the inertia weight. A parameter control mechanism adaptively adjusts the parameters and thus improves the robustness of PSO-MAM was used for a new method-adaptive PSO-MAM (APSO-MAM). Despite APSO-MAM used a complicated mechanism to change these three key parameters, it achieved a perfect performance including test was made on composite Benchmark functions. And, APSO-MAM is more robust than PSO-MAM.

There are a large number of variant PSOs, based on previous research work, it is obvious that the adaptive mechanism, multiple learning strategies are demonstrated to be very effective, and to improve the diversity of whole in later stage of PSO or variant PSOs is an important issue.

III. Enhanced Vector of Convergence for CFPSO

The EVC is a novel conception proposed in this paper, and firstly using for CFPSO-EVC. Based on anterior research, it is well known that adaptive inertia weight, and time-varying accelerate coefficients are considered to enhance the convergence diversity. These modifications enhance the improvement of convergence velocity, but they are not very ideal. Despite there are some investigators to test or change the value of the inertia weight and acceleration coefficients, only modify these parameters that is not an efficient way to improve the performance of PSO. Therefore, based on a lot of empirically evaluation and attempts, some new ideas on revision on current positions of particles are discussed. A serious disadvantage of PSO is vulnerable to get stuck in the local optimum, since the some particles in the early period of evolution cannot jump out of local optima [14]-[16]. But this drawback is able to be used rightly under some control. Based on tournament selection used in CLPSO [11], some particles with smaller fitness value rather than larger value are included.

Then, one dimension of particles are randomly selected from *gbest* to construct to each element of a vector. Next, the vector is added to the current position to next iteration. By the observation on Benchmark functions, especially for unimodal functions including shifted and rotated ones, the EV dramatically improve the convergence velocity; for example, the canonical PSO, based on our evaluation, does not perform qualifiedly on 30 dimensions unimodal functions, however, if EV is added, the performance is superior to some prominent variant PSOs, such as FIPS, CLPSO, POMA , FPSO. Moreover, compared with CLPSO, for the same accurate level, the canonical PSO with EV just need about 1,000 iterations, but CLPSO really needs about 100,000 iterations. For further improvement, EV is not used alone, and it should be better to combine other mechanism as adaptive inertia weight and time-varying acceleration coefficients. A pseudo-code of EV:

Step 1	***for*** each dimension *d* of *gbest*
Step 2	Vector(i):=random(0,1) $\otimes$ gbest(i);
Step 3	***end for***
Step 4	Calculate fitness value of Vector;
Step 5	Find the minimum fitness value and the relevant Vector(i);
Step 6	Perform Roulette Wheel to find a particle;
Step 7	Use Vector(i) to substitute original *position*

The *gbest* is:

$$gbest = [g_1 \quad g_2 \quad \cdots \quad g_N] \tag{1}$$

Each g_i represents an element of *gbest*, in step 2, each g_i is used to construct a vector such as $vector = [g_i \quad g_i \quad \cdots \quad g_i]$; by iterations, we can pick up a vector with a minimum fitness value in step 3 and step 4 , and the Enhance Vector of Convergence such as $EVC = [g_P \quad g_P \quad \cdots \quad g_P], (1 \le P \le N)$; finally, in step 6 and step 7, EVC is utilized to replace one particle of *pbest* in PSO.

The EVC is really efficient for improvement of convergence velocity, but including EVC is sometimes a negative factor for some functions. The influence of EV is really powerful, it may cause some particles are running out of the given range. And the passive influence is continuous functional until the allowance maximum iteration is reached. The result is that most of particles search the space which is out of given range. All particles diverge rather than convergence. As a result, it must be developed a new strategy to control the diverging phenomena. The principle proposed in CLPSO is used to avoid the negative factor of EVC, if the position of particle is out of given range, this particle cannot be used to update.

The update model is the same as CFPSO, and EVC is applied when the update of positions and velocity is finished. The best information of position will be abstracted, and EVC is constructed as a potential power to attract other particles to move toward the global optima.

IV. Experimental Study

A. Test Functions

Benchmark functions and its revised version is used widely as an experimental system for PSO in recent years [3]-[6], [11] and [14]. Test functions *f1-f11* is selected from standard Benchmark functions [17], and *f12-f16* is shifted Sphere, shifted Rosenbrock, shifted Schwefel 1.2, shifted Ackley, and rotated Sphere, respectively [4], [18]. The test functions *f1-f16* is in 30 dimensions. The functions *f17-f20* are are corresponding to *f12-f15* of standard Benchmark functions. In Table 1, the parameters setting of six variant PSOs is shown :

TABLE I
Parameter Tuning

Name	Setting Reference
CFPSO	[9]
UPSO	[19]
CPSO	[16]
FIPS	[10]
CLPSO	[11]

CFPSO_EVC	$C_1 = C_2 = 2.05, \omega_0 = 0.9$

B. Results

In this section, our new variant CFPSO-EVC is firstly tested on 20 functions, compared with other 5 peer PSOs. The dimension of function *f1-f16* is 30. All algorithms are run independent 30 times. The mean and variance solution accuracy is presented. Furthermore, all algorithms are compared by successful rate experiment. In [14], the acceptance of function *f1-f11* are introduced, the comparison is made from *f1* to *f11*. Finally, a statistical known as *t-test* can indicate the distinctive of these 6 variant PSOs based on 30 independent tests. The analysis of aforementioned 3 experiments will be discussed in the following section.

TABLE II
COMPARISON OF MEAN AND VARIANCE

Functions	*F1*	*F2*	*F3*	*F4*
CFPSO	7.04e+001 ± 1.23e-001	3.47e+002 ± 4.84e+000	1.33e+003 ± 2.38e+000	4.92e+001 ± 8.53e+000
UPSO	9.84e-056 ± 3.25e-098	1.97e-028 ± 3.09e-045	5.28e-054 ± 3.27e-084	1.28e-001 ± 6.83e-004
CPSO	1.54e-005 ± 4.63e-010	5.76e-003 ± 4,73e-005	7.51e-004 ± 5.94e-008	3.34e-001 ± 6.52e-003
FIPS	1.76e-019 ± 2.67e-025	7.54e-004 ± 3.45e-010	1.17e-004 ± 3.93e-006	3.94e-001 ± 4.57e-003
CLPSO	7.36e-029 ± 3.43e-053	7.67e-015 ± 3.45e-024	9.48e-023 ± 7.43e-030	9.49e-014 ± 6.47e-020
CFPSO-EVC	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**3.27e-108 ± 2.23e-177**
Functions	*F5*	*F6*	*F7*	*F8*
CFPSO	3.46e+004 ± 4,83e+001	1.41e+001 ± 3.83e+000	4.24e+004 ± 3.84e+000	5.43e+006 ± 3.24e+002
UPSO	8.79e+001 ± 9.84e+000	**0.00 ± 0.00**	9.89e+001 ± 9.35e-001	4.66e+010 ± 3.25e+002
CPSO	4.66e+001 ± 3.56e+000	3.97e-006 ± 4.83e-010	1.47e+001 ± 9.83e-001	**1.33e-002 ± 6.35e-005**
FIPS	2.44e+001 ± 9.19e-001	3.53e-007 ± 4.78e-011	9.46e+000 ± 8.86e-001	3.24e+004 ± 2.48e+001
CLPSO	2.24e+000 ± 8.76e-001	6.37e+000 ± 3.24e-002	8.98e+000 ± 5.56e-001	1.04e+004 ± 1.36e+001
CFPSO-EVC	**1.02e+000 ± 7.64e-001**	**0.00 ± 0.00**	**1.00e+000 ± 1.24e-001**	9.84e+002 ± 7.45e+000
Functions	*F9*	*F10*	*F11*	*F12*
CFPSO	1.12e+002 ± 2.64e-001	1.82e-006 ± 4.46e-009	1.17e+000 ± 5.31e-002	2.04e+001 ± 3.84e-001
UPSO	6.16+001 ± 5.47e-001	1.82e-006 ± 3.94e-010	**0.00 ± 0.00**	**0.00 ± 0.00**
CPSO	8.66e-007 ± 4.37e-009	5.45e-004 ± 3.84e-008	7.06e-006 ± 8.92e-010	2.14e-005 ± 7.08e-008
FIPS	1.03e+002 ± 9.73e+000	1.01e-004 ± 5.52e-007	**7.17e-004 ± 6.03e-007**	1.53e-006 ± 7.83e-011
CLPSO	**0.00 ± 0.00**	1.83e-006 ± **3.83e-010**	**0.00 ± 0.00**	4.93e+001 ± 3.98e+000
CFPSO-EVC	**0.00 ± 0.00**	**1.82e-010** ± 4.55e-010	**0.00 ± 0.00**	**0.00 ± 0.00**
Functions	*F13*	*F14*	*F15*	*F16*
CFPSO	7.31e+003 ± 4.55e+001	1.55e+001 ± 3.65e+000	3.18e-001 ±	**0.00 ± 0.00**
UPSO	7.28e+001 ± 6.43e+000	**0.00 ± 0.00**	7.39e-002 ± 8.14e-004	1.81e-008 ± 2.36e-011
CPSO	2.70e+001 ± 4.36e-001	9.14e-003 ± 8.97e-005	3.69e-002 ± 4.93e-003	4.78e-012 ± 5.67e-020
FIPS	4.36e+001 ± 3.55e+000	2.20e-001 ± 3.64e-003	4.58e-005 ± 9.74e-006	1.21e-004 ± 3.06e=005
CLPSO	7.14e+002 ± 3.98e+000	2.56e+001 ± 7.84e-001	7.83e-001 ± 3.45e-003	**3.22e-029 ± 4.56e-035**
CFPSO-EVC	**1.24e+001 ± 5.43e-001**	**0.00 ± 0.00**	**0.00 ± 0.00**	1.11e-022 ± 4.53e-030
Functions	*F17*	*F18*	*F19*	*F20*
CFPSO	1.13e+001 ± 4.49e+000	4.84e+001 ± 1.81e+001	**1.98e+000** ± 5.92e-016	3.43e-009 ± 4.41e-011
UPSO	4.39e+000 ± 3.21e+000	2.19e-001 ± 1.17e-009	2.00e+000 ± 1.48e-013	**9.15e-004** ± 1.67e-004
CPSO	4.31e+001 ± 1.93e-001	**3.91e-002** ± 2.41e-003	**1.98e+000** ± 6.38e-012	8.00e-003 ± 1.65e-002
FIPS	4.51e+001 ± 3.87e+001	4.97e+001 ± 6.93e+001	1.99e+000 ± 2.96e-012	1.00e+000 ± **4.98e-05**
CLPSO	4.31e+001 ± 1.92e-001	2.69e+001 ± 1.06e+001	**1.98e+000** ± 3.11e-013	1.00e-003 ± 4.57e-004
CFPSO-EVC	**3.46e+000 ± 2.55e-010**	2.43e+000 ± **1.15e-009**	**1.98e+000 ± 4.68e-016**	1.00e-003 ± 4.17e-004

TABLE III
COMPARISON OF ACCEPTANCE RATE
#S(ACCEPTANCE RATE>80%), #PS(50%< ACCEPTANCE RATE <80%) AND #NS (ACCEPTANCE RATE<50%)

	ITERATION	UPSO	CPSO	FIPS	CFPSO	CLPSO	CFPSO-EVC
#S, #PS, #NS	5000	0,2,9	0,1,10	0,4,7	0,1,10	0,0,11	10,0,1
#S, #PS, #NS	10000	2,2,7	1,1,9	1,3,7	0,2,9	5,3,3	10,1,0

#S, #PS, #NS	100000	10,0,1	9,1,1	2,3,6	2,1,8	10,0,1	10,1,0

TABLE IV
COMPARISON OF *T-TEST*
#+(BETTER), #~(EQUAL) AND #-(WORSE)

	ITERATION	**UPSO**	**CPSO**	**FIPS**	**CFPSO**	**CLPSO**	**CFPSO-EVC**
#+, #~, #-	10000	2,7,11	2,0,18	0,2,18	1,0,19	2,1,17	8,5,7

C. *Analysis*

Table II indicates that the testing results of comparison on mean and variance.

CFPSO performs excellent on some low dimensional functions, but fail in most of higher dimensional problems. The performance of CFPSO is influence by shifted, rotated methods.

UPSO can obtain a higher accuracy in shifted problems, for standard problems, and UPSO can also achieve a relative accurate results. For *f8*, UPSO cannot find a qualified result, compared with CLPSO and CFPSO-EVC. UPSO can solve some multimodal problem in low dimension.

CPSO can provide the most accurate result of *f8*, but this variant PSO does not perform well on most unimodal, multimodal, and even shifted, rotated functions.

The performance of FIPS is not impressive. For most test problems, the FIPS cannot present a qualified accurate level. Its performance is like to UPSO, in some low dimensional multimodal problem, and it can gain the most accuracy.

CLPSO is not skillful at solving most unimodal problem. Although CLPSO owns a slow convergence velocity for unimodal functions, it obtains the most accurate result for rotated problems.

Based on constrict factor and EVC, CFPSO-EVC owns a very fast convergence velocity and most accuracy for most functions. The performance on Rosenbrock function (*f5*) and shifted Rosenbrock function (*f13*) better than other peer variant PSOs. CFPSO-EVC can achieve a very good accurate level for shifted functions, and gain the most accurate results on the most of unimodal problems. CFPSO-EVC is not good at solving rotated functions, and some low dimensional multimodal functions. For *f17-f20*, the four multimodal functions, CFPSO-EVC obtains the highest accurate level in *f17*. In *f19*, the performance of CFPSO-EVC is equal to CPSO, CLPSO and CFPSO.

Based on acceptance rate and *t-Test*, obviously, CFPSO-EVC is superior to other 5 PSOs. Generally, CFSO-EVC is more accurate variant PSOs, and owns a faster convergence velocity.

V. CONCLUSION

This paper presents a new mechanism combined constrict factor and EVC. The principle EVC is trying to collect the best information of global optima found by the whole swarm, and then the EVC is randomly to substitute original position of particles. Generally, EVC is a potential power to persuade other particles to move toward best value. The disadvantage of EVC is that some particles are more vulnerable to jump out the given range, which is a negative factor for convergence. As a result, the particles, which is within the original range, will be used to update.

Based on our test, CFPSO-EVC is very skilled at unimodal problems, shifted unimodal and some multimodal problems. For some low dimensional multimodal problems, the CFPSO-EVC is not a right choice.

CFPSO-EVC can be combined with other algorithm to enhance the convergence velocity in order to obtain a more accurate level in most problems.

ACKNOWLEDGMENT

This paper is supported by the 2012 Western Boundary and Areas Planning Foundation of China Ministry Education under Grant No. 12XJA910002.

The authors thank Professor P. N. Suganthan to provide the source code of CLPSO for our experimental study.

REFERENCES

[1] R. C. Eberhart and J. Kennedy, “A New Optimizer Using Particle Swarm Theory,” in *Proc. 6th Int. Symp*. Micromachine Human Sci., Nagoya, Japan, pp. 39 –43, 1995.

[2] J. Kennedy and R. C. Eberhart, “Particle Swarm Optimization,” in *Proc. IEEE Int. Conf. Neural Networks*, pp. 1942 –1948, 1995.

[3] Z. Zhan, J. Zhang, Y. Li, and H. S. Chung, “Adaptive Particle Swarm Optimization,” *IEEE Trans. Syst., Man, Cybern*. B, *Cybern*., vol. 39, no. 6, pp. 1362–1381, 2009.

[4] C. H. Li, S. X. Yang and T. T. Nguyen, “Self-Learning Particle Swarm Optimizer for Global Optimization Problem,” *IEEE Trans. Syst., Man, Cybern*. B, *Cybern*., vol. 42, no. 3, pp. 627-646, 2012.

[5] M. Hu, T. Wu, and J. D. Weir, “An Intelligent Augmentation of Particle Swarm Optimization with Multiple Adaptive Methods,” *Inform. Sci* , vol. 213, pp. 68–83, 2012.

[6] M. Hu, T. Wu, and J. D. Weir, “An Adaptive Particle Swarm Optimization with Multiple Adaptive Methods,” *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 705-720, 2013.

[7] Y. Shi and R. C. Eberhart, “A Modified Particle Swarm Optimizer, ” in *Proc. IEEE Congr. Evol. Comput.*, pp. 69 –73, 1998.

[8] Y. Shi and R. C. Eberhart, “Parameter Selection in Particle Swarm Optimization,” in *Proc. 7th Conf. Evol. Programming*, New York, pp. 591 –600, 1998.

[9] M. Clerc and J. Kennedy, "The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space, " *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58 –73, 2002.

[10] R. Mendes, J. Kennedy, and J. Neves, "The Fully Informed Particle Swarm: Simpler, maybe better," *IEEE Trans. Evol. Comput.* , vol. 8, no. 3, pp. 204–210, 2004.

[11] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baska, "Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, 2006.

[12] M. A. M. de Oca, T. Stutzle, M. Birattari, and M. Dorigo, "Frankenstein's PSO: A Composite Particle Swarm Optimization Algorithm," *IEEE Trans. Evol. Comput*. , vol. 13, no. 5, pp. 1120–1132, 2009

[13] Y. Cooren, M. Clerc, and P. Siarry, "Performance Evaluation of TRIBES, An Adaptive Particle Swarm Optimization Algorithm," *Swarm Intell*, vol. 3, no. 2, pp. 149–178, 2009.

[14] Z. H. Zhan, J. Zhang, Y. Li, and Y. Shi, "Orthogonal Learning Particle Swarm Optimization," *IEEE Trans. Evol. Comput*. vol. 15, no. 6, 2011

[15] Z. X. Zhu, J. R. Zhou, J. Zhen, and Y. Shi, "DNA Sequence Compression Using Adaptive Particle Swarm Optimization-Based Memetic Algorithm," *IEEE Trans. Evol. Comput*., vol. 15, no. 5, pp. 643-657, 2011.

[16] F. van den Bergh and A. P. Engelbrecht, "A Cooperative Approach to Particle Swarm Optimization," *IEEE Trans. Evol. Comput*., vol. 8, no. 3, pp. 225-239, 2004

[17] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, 1999.

[18] J. Liang, P. N. Suganthan, and K. Deb, "Novel Composition Test Functions for Numerical Global Optimization," in *Proc. Symp. Swarm Intell.*, pp. 68–75, 2005.

[19] K. E. Parsopoulos and M. N, Vrahatis. "UPSO-Unified Particle Swarm Optimization Scheme," in Lecture Series on Computational Science, 2004, pp.868-873.

[20] A. Corana, M. Marchesi, C. Martini, "Minimizing Multimodal Functions of Continuous Variables with the Simulated Annealing' Algorithm," ACM Transaction on Mathematical Software, 1987, vol. 13, no. 3, pp. 262-280, 1987.