Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/081L45

A Novel Double Glowworm Swarm Co-Evolution Optimization Algorithm based Lévy Flights

*Yongquan Zhou**, *Yingju Wang, Sucai He and Jinzhao Wu*

Key Laboratory of Hybrid Computation and IC Design Analysis, College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006 P. R. China

Received: 21 May. 2013, Revised: 15 Sep. 2013, Accepted: 16 Sep. 2013
Published online: 1 Apr. 2014

Abstract: In this paper, a novel double glowworm swarm co-evolution optimization algorithm based Lévy flights is presented. According to the different colors of light emitted by glowworm swarm, a certain amount of glowworm swarm was divided into two groups. Lévy flights with higher randomness were introduced into one group. Then the two groups of glowworm swarm seek the optimal solution simultaneously and co-evolution for achieving the global optimization. The numerical simulation results show that double glowworm swarm co-evolution optimization algorithm based Lévy flights has greatly improved than the basic algorithm in terms of overall and convergence.

Keywords: Glowworm swarm optimization, Lévy flights, double glowworm swarm co-evolution, function optimization.

1 Introduction

Swarm intelligence algorithm is derived from the inspiration of the law of the natural or biological population. According to its principle and imitating its own rules, the algorithm was designed for solving the problems [1]. With the development of computational intelligence techniques in recent years, some new biological intelligent algorithm have been proposed, such as ant colony algorithm, particle swarm algorithm, cuckoo search, glowworm swarm optimization algorithm, and so no. The expert of bionics' research result shows that, in the nature, glowworms communicate with each other by releasing luciferin. Glowworms release luciferin when they are flying, so they can give out fluorescent light. Glowworms attract others around them by giving out fluorescent light. The higher the concentration of fluorescein, the greater the intensity of fluorescence, then glowworm can be able to attract more other glowworms.

Inspired by the behavior of natural glowworm swarm, Glowworm Swarm Optimization (GSO) algorithm which is a novel swarm intelligence algorithm was advanced by Indian scholars Krishnanand and Ghose in 2005 years [2, 3]. At present, GSO algorithm has been successfully used in the noise test, simulation of the sensor machine crowd [4], clustering analysis [5,6], numerical optimization calculation [7,8], knapsack problem [9], etc. But the basic GSO algorithm has some shortcomings, such as slow convergence, low precision and easy to fall into local optimization. These shortcomings limited the range of application of GSO algorithm greatly. Based the analysis of defects in the basic GSO algorithm, this algorithm was improved and the Lévy flights [10,11,12] was used in it, so double glowworm swarm co-evolution optimization algorithm based Lévy flights was presented.

2 Basic Glowworm Swarm Optimization Algorithm and Analysis

2.1 The basic GSO

In the basic GSO algorithm, a swarm of glowworms are randomly distributed in the search space of object functions. Accordingly, these glowworms carry a luminescent quantity called luciferin along with them and they have their own decision domain $r_d^i (0 < r_d^i \leq r_s)$. The glowworms emit light which intensity is proportional to the associated luciferin and interact with other glowworms within a variable neighborhood. The glowworms' luciferin intensity is related to the fitness of their current locations. The higher the intensity of

* Corresponding author e-mail: yongquanzhou@126.com

luciferin, the better the location of glowworm, in other words, the glowworm represents a good target value. Otherwise, the target value is poor. A glowworm i considers another glowworm j as its neighbor if j is within the neighborhood range of i and the luciferin level of j is higher than that of i. In particular, the neighborhood is defined as a local-decision domain that has a variable neighborhood range r_d^i bounded by a radial sensor range $r_s(0 < r_d^i \leq r_s)$. Each glowworm selects, using a probabilistic mechanism, a neighbor that has a luciferin value higher than its own and moves toward it. That is, glowworms are attracted by neighbors that glow brighter. In addition, the size of the neighborhood range of each glowworm is influenced by the quantity of glowworms in the neighborhood range. The neighborhood range of the glowworm is proportional to the density of its neighbors. If the neighborhood range covers low density of glowworms, the neighborhood range will be increased. On the contrary, the neighborhood range will be reduced.

In short, GSO algorithm includes four stages: the initial distribution of glowworms, Luciferin-update phase, Movement-phase, Neighborhood range update.

(1) *The initial distribution of glowworms phase*

The initial distribution of glowworms phase, in other words, it is initialization phase. Purpose is to make the glowworms randomly distribute in the search space of object functions. Accordingly, these glowworms carry the same intensity luciferin and they have the same decision domain r_0.

(2) *Luciferin-update phase*

The glowworms' luciferin intensity is related to the fitness of their current locations. The higher the intensity of luciferin, the better the location of glowworm, in other words, the glowworm represents a good target value. Otherwise, the target value is poor. In the algorithm of each iteration process, all the glowworms' position will change, and then the luciferin value also follows updates.

At time t, the location of the glowworm i is $x_i(t)$, corresponding value of the objective function at glowworm i's location at time t is $J(x_i(t))$, put the $J(x_i(t))$ into the $l_i(t)$. $l_i(t)$ Represents the luciferin level associated with glowworm i at time t. The formula as follows:

$$l_i(t) = (1-\rho)l_i(t-1) + \gamma J(x_i(t)) \quad (1)$$

where ρ is the luciferin decay constant $(0 < \rho < 1)$, γ is the luciferin enhancement constant.

(3) *Movement-phase*

At the movement phase, every glowworm selects a nei-ghbor and then moves toward it with a certain probability. As the glowworm i's neighbor need to meet two requirements: one, the glowworm within the decision domain of glowworm i; two, the luciferin value is larger than the glowworm i's. Glowworm i moves toward a neighbor j which comes from $N_i(t)$ with a certain probability, the probability is $p_{ij}(t)$. Using the formula (2) calculates it:

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum\limits_{k \in N_i(t)} l_k(t) - l_i(t)} \quad (2)$$

Glowworm i after moving, then the location is updated, the location update formula is:

$$x_i(t+1) = x_i(t) + st^* \left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \right) \quad (3)$$

where st is the step size.

(4) *Neighborhood range update phase*

With the glowworm's position updating, it neighborhood range also follow update. If the neighborhood range covers low density of glowworms, the neighborhood range will be increased. On the contrary, the neighborhood range will be reduced. The formula of neighborhood range update as follows:

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_t - |N_i(t)|)\}\} \quad (4)$$

where β is a constant parameter and n_t is a parameter used to control the number of neighbors.

The basic GSO algorithm as follows [1]:

Set number of dimensions= m;
Set number of glowworms= n;
Let S be the step size;
Let $x_i(t)$ be the location of glowworm i at time t;
Deploy_agents_randomly;
For $i = 1$ *to* n *do* $l_i(0) = l_0$;
$r_d^i(0) = r_0$;
Set maximum iteration number=iter_max;
While $(t < iter\_max)$ *do*
{
For each glowworm i do:%Luciferin-update phase;
$l_i(t) = (1-\rho)l_i(t-1) + \gamma J(x_i(t))$; %See (1)
For each glowworm i do:%Movement-phase
{
$N_i(t) = \{j : d_{ij}(t) < r_d^i(t); l_i(t) < l_j(t)\}$
For each glowworm $j \in N_i(t)$ *do;*
$p_{ij}(t) = \frac{l_j(t)-l_i(t)}{\sum_{k \in N_i(t)} l_k(t)-l_i(t)}$; %See(2)
$x_i(t+1) = x_i(t) + st^* \left(\frac{x_j(t)-x_i(t)}{\|x_j(t)-x_i(t)\|}\right)$; % See(3)
$j = select\_glowworm\ (\overrightarrow{p})$;
$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_t - |N_i(t)|)\}\}$; %See(4)
}
$t \leftarrow t+1$:
}.

2.2 Analysis of the basic GSO algorithm

In the present model of GSO algorithm, each glowworm, according to the luciferin value, decides to move toward a neighbor that has a luciferin value higher than its own. Finally, glowworms are attracted to neighbors with glow brighter. Glowworms seek the glowworm with the brightest light through moving toward it. In the present GSO algorithm, glowworms search in a certain area. If there are a lot of glowworms in the certain area, each glowworm have more neighbors which can increase the number of glowworms that the glowworm must be researched. This can result in having more time to seek the optimal solution, that is, slow convergence. If there are little glowworms in the certain area, each glowworm has little neighbors which lead to inadequate among glowworms and not timely collaboration and easy to fall into local optimization. That is low precision.

In nature, glowworms emit a luminescence through releasing luciferin. There are different kinds of glowworms. Because of this, different kinds of glowworms emit different colors of light. The yellow and green colors are usually seen. The glowworms with same color move toward each others. In the present model of GSO algorithm, this phenomenon is not considered. Based on this, the strategy that double glowworm swarm was used. In the Cuckoo Search algorithm, in order to enhance the randomness of Cuckoo searching the optimal solution, Cuckoo uses a specified flight way that with higher randomness-Lévy flights. This flight mode greatly improved the randomness of Cuckoo searching the optimal solution. In this paper, Levy flights is applied in the double glowworm swarm, so the double glowworm swarm co-evolution optimization algorithm based Lévy flights (LDGSO) was presented.

3 Lévy Flights

In the Cuckoo Search (CS) algorithm [12], the cuckoo seeks the nest by the Lévy flights. Cuckoo gains the best path of looking for the nest by Lévy flights. During the search, the flight path of the cuckoo is a combination by different length fight path that some long and some short. There have a small angle between the adjacent flight paths. The short paths appear with a higher frequency, while the appearance of longer path is relatively sparse. This is an optimal search strategy known as Lévy flights. Actually Lévy flights provided a random walking path. The random pace comes from a wide range of Lévy flights. The formal of Lévy flights as follows.

$$X_i^{(t+1)} = X_i^t + \alpha \oplus L\acute{e}vy(\lambda)\ (i = 1, 2, \cdots, n) \quad (5)$$

where X_i^t represents the location of the glowworm i at the time t. $\oplus$ represents the point to point multiplication. α is a parameter used to control the seize of steps. $L(\lambda)$ represents the random search path of Lévy. In the Cuckoo Search algorithm, the expression can be approximated described as follow:

$$L\acute{e}vy \sim u = t^{-\lambda}, (1 < \lambda \leq 3) \quad (6)$$

where λ is a constant.

4 Double Glowworm Swarm Co-evolution Optimization Algorithm Based Lévy Flights

4.1 The strategy of the improved algorithm

When using the GSO algorithm optimize the functions, according to the different colors of light, a certain amount of glowworm swarm is divided into two groups. The glowworms with yellow luminescence make up a sub-population, the glowworms with green luminescence make up another sub-population. The two groups of glowworm swarm simultaneously seek the optimal solution in the search time. One group searches the optimal solution according to the way of basic GSO; another group takes the way of Lévy flights to get the optimal solution.

When reaching a certain number of iterations, the glowworms of two populations basically converge to the around of optimal value, then the two populations of glowworms are seen as two glowworms. Each glowworm will be seen as the glowworm with the brightest light in the sub-populat-ion. Next, one glowworm moves toward another one that has a higher luciferin value. This collaborative way between two populations is helpful for glowworms out of local optimum and the speed of convergence will be improved greatly. According to this strategy the double glowworm swarm co-evolution optimization algorithm based Lévy flights (LDGSO) is designed.

4.2 The steps of the improved algorithm

The steps of double glowworm swarm co-evolution optimization algorithm based Lévy fights can be described as follows:

Step 1: Initialize the population: set dimension is m, the number of glowworms is $2n$, step size is st, and so on.

Step 2: According to the different colors of light, a certain amount of glowworm swarm is divided into two groups. The size of sub-population is n.

Step 3: Placing the two groups of $2n$ glowworms randomly in the search space of the object function.

Step 4: Using the formula (1) put the $J(x_i(t))$ into the $l_i(t)$. $l_i(t)$ represents the luciferin level associated with glowworm i at time t. $J(x_i(t))$ represents the value of the objective function at glowworm i's location at time t.

Step 5: Each glowworm selects a neighbor that has a luciferin value higher than its own to make up the $N_i(t)$.

Step 6: Each glowworm using the formula (2) selects a neighbor.

Step 7: The glowworms of one group move by Lévy flights and then using the formula (5) update the location of the glowworms.

Step 8: The glowworms of another group move by basic GSO and using the formula (3) update the location of the glowworms.

Step 9: Using the formula (4) update the value of the variable neighborhood range.

Step 10: Selecting the glowworm with the brightest light of each sub-population at time t.

Step 11: If reached the specified number of iterations and do not reached the maximum number of iterations, one sub-population move toward another. Otherwise, executing the step (4).

Step 12: If reached the maximum number of iterations, executing the step (10); otherwise, executing the step (4).

Step 13: Output the results. The end.

5 Experimental Results and Analysis

Environment for running programs of this experiment: processor: CPU E4500, main frequency: 2.19GHz, memory: 1.00GB, operating system: Microsoft Windows XP Professional, Version 2002 Mathematical software for programming: Matlab 7.0.

5.1 Test Functions

Values of algorithm parameters that are kept fixed for all the experiments in Table 1. According to the different test functions, the values of other algorithm parameters take different values.

Table 1 The values of algorithm parameters

ρ	β	γ	st	n_t	l_0	α	λ
0.4	0.08	0.6	0.03	5	5	1	-2

Test functions as follows [13]:

$$f_1(x) = 0.5 + \frac{((\sin(\sqrt{x_1^2+x_2^2}))^2 - 0.5)}{(1+0.001*(x_1^2+x_2^2))^2};$$
$$f_2(x) = \sum_{i=1}^{30} x_i^2;$$
$$f_3(x) = 100*(x_i^2 - x_2^2)^2 + (1-x_1)^2;$$
$$f_4(x) = x_i^2 - 10*(\cos(2*\pi*x_i)) + 10;$$
$$f_5(x) = \sum_{i=1}^{30} i*x_i^2;$$
$$f_6(x) = 1 + \frac{1}{4000}\sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30}\cos(\frac{x_i}{\sqrt{i}});$$
$$f_7(x) = \sum_{i=1}^{30}(0.2*x_i^2 + 0.1*x_i^2*\sin 2x_i);$$
$$f_8(x) = (x_1^2+x_2^2)^{0.25}*(\sin^2(50*(x_1^2+x_2^2)^{0.1})+1.0);$$

Table 2 Test functions

Function	Function name	Search space size	Minumum
$f_1(x)$	Schaffer F6	[-100,100]	0
$f_2(x)$	Sphere	[-100,100]	0
$f_3(x)$	Rosen Brock	[-30,30]	0
$f_4(x)$	Restringing	[-10,10]	0
$f_5(x)$	Axis parallel hyper ellipsoid	[-5.12,5.12]	0
$f_6(x)$	Grievance	[-100,100]	0
$f_7(x)$	Function 15	[-10,10]	0
$f_8(x)$	Schaffer F7	[-100,100]	0

5.2 Experimental results

Experimental results are shown in Table 3, Table 4 and range Figure 2 to Figure 9. The results of Table 3 and Table 4 are taken from 10 experiments. Figure 2 to Figure 9 respectively show the optimization results of the basic algorithm and the improved algorithm to eight test functions. In the Figures, solid lines represent the basic GSO algorithm; dotted lines represent the improved GSO algorithm.

5.3 Experimental result analysis

The data from Table 3 can be drawn that the improved GSO algorithm is superior to the basic GSO algorithm in

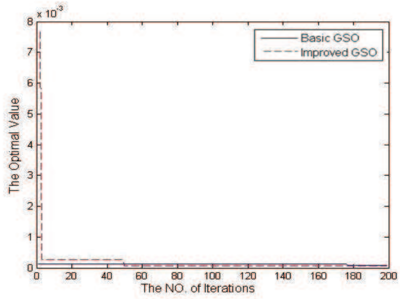

Fig. 1 The $f_1(x)$ convergence cures of the GSO and IGSO

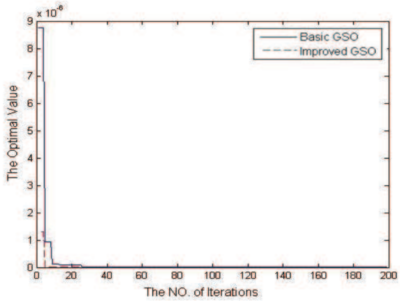

Fig. 2 The $f_2(x)$ convergence cures of the GSO and IGSO

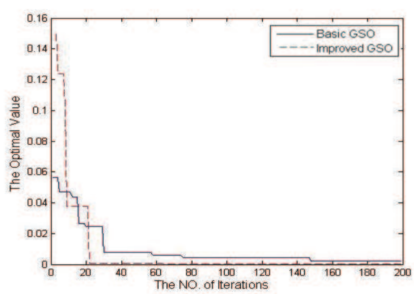

Fig. 3 The $f_3(x)$ convergence cures of the GSO and IGSO

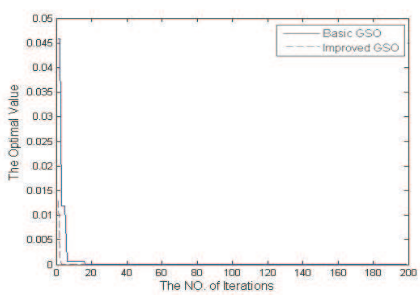

Fig. 4 The $f_4(x)$ convergence cures of the GSO and IGSO

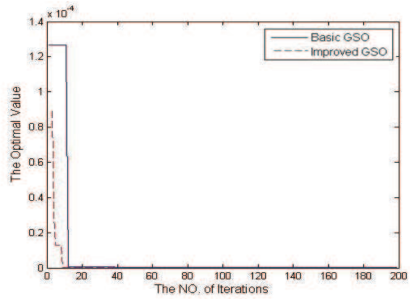

Fig. 5 The $f_5(x)$ convergence cures of the GSO and IGSO

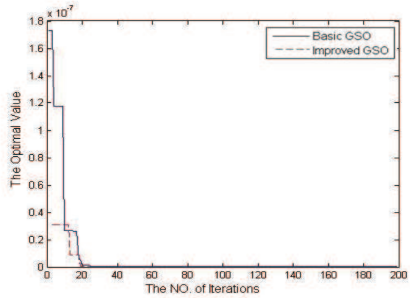

Fig. 6 The $f_6(x)$ convergence cures of the GSO and IGSO

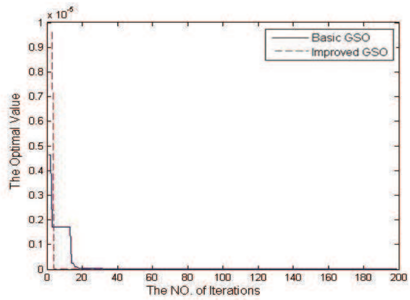

Fig. 7 The $f_7(x)$ convergence cures of the GSO and IGSO

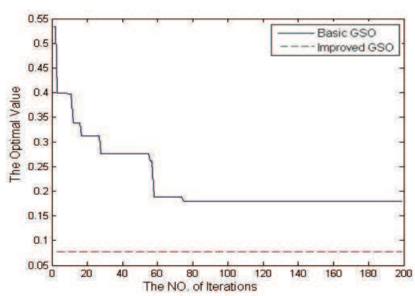

Fig. 8 The $f_8(x)$ convergence cures of the GSO and IGSO

Table 3 Experimental results under different functions

Function	Algorithm	Optimal valve	Wrost value	Average value
$f_1(x)$	GSO	5.057416284 187655e-005	6.469566980 986108e-004	1.8609e-004
	IGSO	3.019746600 796402e-005	3.927655619 168635e-004	1.3617e-004
$f_2(x)$	GSO	1.282756843 401603e-012	5.777756989 937466e-011	2.0133e-011
	IGSO	4.819881950 685840e-014	1.412424004 542435e-011	4.8326e-012
$f_3(x)$	GSO	4.584493371 919619e-004	0.004353462 426470	0.0023
	IGSO	1.081526661 234373e-004	5.663705545 683880e-004	2.6732e-004
$f_4(x)$	GSO	2.772885920 876433e-009	6.886615368 628668e-006	8.9155e-007
	IGSO	1.033324537 047520e-010	1.029835026 145065e-007	1.6376e-008
$f_5(x)$	GSO	3.206368190 292584e-011	1.320524458 972470e-008	6.9909e-009
	IGSO	2.356952193 305129e-011	3.279002228 164048e-009	1.2722e-009
$f_6(x)$	BGSO	2.142730437 526552e-014	7.612799279 854698e-012	3.0680e-012
	IGSO	9.880984919 163893e-015	7.542855229 303314e-013	2.3683e-013
$f_7(x)$	GSO	3.367640317 619063e-012	1.800146785 306065e-010	5.2947e-011
	IGSO	1.039759602 732340e-014	8.066891419 328334e-011	1.1412e-011
$f_8(x)$	GSO	0.08878537 8430453	0.181853623 576050	0.1433
	IGSO	0.04616282 3998851	0.12661467 8347334	0.0831

Table 4 The average number of iterations and the average search time

Functions		$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
The average NO. of iterations	GSO	116	124	128	137.7
	IGSO	96	59	144	95
The average search time	GSO	4.6812	5.0372	4.1716	4.9006
	IGSO	2.8235	3.0058	2.4109	2.9995
Functions		$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$
The average NO. of iterations	GSO	92	104	106	143.7
	IGSO	56	103	55	59.8
The average search time	GSO	4.8173	5.4011	4.8562	4.3883
	IGSO	2.8821	3.3623	3.0479	2.7513

the accuracy. From Table 3 we can see that the accuracy in the improved GSO algorithm is higher than in the basic algorithm. The data from Table 4 can be drawn that the improved GSO algorithm is better than the basic GSO algorithm in the average number of iterations and the average search time. So that we can draw the improved GSO algorithm is superior to the basic GSO algorithm in the speed of convergence. From Figure 2 to Figure 9: the convergence map of basic and improved GSO algorithm, which can prove the improved GSO algorithm is better than the basic GSO algorithm. The experimental results show that the double glowworm swarm co-evolution optimization algorithm based Lévy flights is superior to the basic GSO algorithm.

6 Conclusion

In this article, firstly, the model of the basic glowworm swarm optimization algorithm has been introduced and analyzed at some length. The analysis shows that the basic glowworm swarm optimization has some disadvantages. On this basis, according to the different colors of fluorescent light, a certain amount of glowworm swarm was divided into two groups. In order to increase the randomness of the whole population, Lévy flights with higher randomness were introduced into one group. Then the two groups of glowworm swarm seek the optimal solution simultaneously. It can be seen from the experimental results, the proposed algorithm has a great improvement in the convergence and in terms of overall.

Acknowledgement

This work is supported by National Science Foundation of China under Grant No. 61165015. Key Project of Guangxi Science Foundation under Grant No. 2012GXNSFDA0530
28, Key Project of Guangxi High School Science Foundation under Grant No. 20121ZD008 and the Funded by Open Research Fund Program of Key Lab of Intelligent Perception and Image Understanding of Ministry of Education of China under Grant No. IPIU01201100. Thanks for the help.

References

[1] K. N. Krishnand, D. Ghose, Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In proceeding of IEEE swarm intelligence symposium, Piscataway: IEEE press, 84-91 (2005).
[2] K. N. Krishnand, D. Ghose, Glowworm swarm optimisation for simultaneous capture of multiple local optima of multimodal functions, Swarm Intell, **3**, 87-124 (2009).
[3] K. N. Krishnand, D. Ghose, Glowworm swarm optimization: a new method for optimizing multi-modal functions, Int. J. Computational Intellingence Studies, **1**, 93-119 (2009).

[4] K. N. Krishnand, D. Ghose, Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications, Multiagent and Grid Systems, **2**, 209-222 (2006).
[5] Kuo-Tai Tseng, A glowworm algorithm for solving data clustering problems, Tatung University, (2008).
[6] Zhengxin Huang, Yongquan Zhou, Using glowworm swarm optimization algorithm for clustering analysis, Journal of Convergence Information Technology, **6**, 78-85 (2011).
[7] Yan Yang, Yangquan Zhou, Glowworm swarm optimization algorithm for solving numerical integral, Communications in Computer and Information Science, **134**, 389-394 (2011).
[8] Yan Yang, Yangquan Zhou, Qiaoqiao Gong, Hybrid artificial glowworm swarm optimization algorithm for solving system of nonlinear equations, Journal of Computational Information Systems, **6**, 3431-3438 (2010).
[9] Qiaoqiao Gong, Yongquan Zhou, Yan Yang, Artificial glowworm swarm optimization algorithm for solving 0-1 knapsack problem. Advanced Materials Research, **143-144**, 166-171 (2011).
[10] Xin-She Yang, Suash Deb, Cuckoo Search via Lévy Flights, In: Proc. of World Congress on Nature Biologically Inspired Computing (NaBIC2009), India: IEEE Publications, 210-214 (2009).
[11] Barthelemy P., Bertolotti J., Wiersma D. S., A Lévy flight for light, Nature, **453**, 495-498 (2008).
[12] Brown C., Liebovitch L.S., Glendon R., Lévy flights in Dobe Ju/hoansi foraging patterns, Human Ecol., **35**, 129-138 (2007).
[13] Kusum Deep, Jagdish Chand Bansal, Mean particle swarm optimization for function optimization. Int. J. Computational Intelligence Studies, **1**, 72-92 (2009).

Yongquan Zhou received the MS degree in Computer science from Lanzhou University in 1993, and the PhD degree in Computation Intelligence fr-om the Xiandian University in 2006. He is currently a professor in Guangxi University for Nationalities. His research interests are in the areas of computation intelligence, neural networks, and intelligence information processing.

Yingju Wang, Master, was born in 1987. Her research interests include computation intelligence, swarm intelligence algorithm and applies.

Sucai He Master, was born in 1985. His research interests include computation intelligence, swarm intelligence algorithm and applies.

Jinzhao Wu Ph.D. Professor. His received the MS degree in Computer science from Lanzhou University in 1991, and the Ph. D degree in Computation Science from the Pek-ing University in 1995. His is currently a professor in Guan-gxi University for Nationalities. His research interests are in the areas of hybrid computation and Integrated circuit validation.