Applied Mathematics & Information Sciences
An International Journal

An improved Cuckoo Search Algorithm for Solving Planar Graph Coloring Problem

Yongquan Zhou[1,2], Hongqing Zheng[2], Qifang Luo[2], Jinzhao Wu[1,2]

[1] Guangxi Key Laboratory of Hybrid Computation and Integrated Circuit Design Analysis, Nanning Guangxi 530006, P. R. China.
[2] College of Information Science and Engineering, Guangxi University for Nationalities, Nanning Guangxi 530006, P. R. China.

Received: Jul 19, 2012; Revised Oct. 27, 2012; Accepted Nov. 13, 2012
Published online: 1 Mar. 2013

Abstract: In this paper, we proposed an improved cuckoo search optimization (ICS) algorithm for solving planar graph coloring problem. The improved cuckoo search optimization algorithm is consisting of the walking one strategy, swap and inversion strategy and greedy strategy. The proposed improved cuckoo search optimization algorithm can solve the planar graph coloring problem using four-colors more efficiently and accurately. The experimental results show that we proposed improved cuckoo search optimization algorithm can get smaller average iterations and higher correction coloring rate.

Keywords: cuckoo search algorithm; improved cuckoo search optimization; planar graph coloring problem.

1. Introduction

The graph coloring problem has been proved to be a classic NP-complete problem. Until now, there is not an effective strategy to get the best solution. For solving this kind of problem, both the exact algorithms and approximate algorithms have been used including ant colony optimization algorithm in [1,4,12,13], tabu search algorithm in [2,7,11], genetic algorithm in [18], particle swarm optimization algorithm in [6,10], neural network algorithm in [14], Quantum Search Algorithm [19] etc. It can be applied to many engineering applications, such as time tabling and scheduling in [5], radio frequency assignment in [8], computer register allocation in [3], and printed circuit board testing in [9].
Recently, a novel heuristic search algorithm, called Cuckoo Search(CS) [15], has been proposed by Yang and Deb in 2009. The CS is a search swarm intelligence algorithm based on the interesting breeding behavior such as brood parasitism of certain species of cuckoos. Each nest within the swarm is represented by a vector in multi-dimensional search space; the CS algorithm also determines how to update the position of cuckoo laid egg. Each cuckoo updates it position of lay egg based on current step size via Lévy flights. It has been shown that this simple model has been applied successfully to continuous nonlinear function, engineering optimization problem [17], etc. The CS was originally developed for continuous valued spaces, but many problems are, however, defined for discrete valued spaces where the domain of the variables is finite. By solving planar graph coloring problem using CS, we proposed a discrete quaternary version of cuckoo search.

The remainder of this paper is organized as follows: Section 2 briefly overviews the procedure of the planar graph coloring. Section 3 describes the CS algorithm and an improved Cuckoo Search algorithm (ICS). The experiment result in Section 4, and discusses the proposed ICS for solving planar graph coloring problem. This paper concludes in Section 5.

2. The Procedure of the Planar Graph Coloring

What is the minimum number of colors that can be used to color the regions in a planar map with neighboring regions having different colors? This has been a problem of interest for over a century. As early as 100 years ago there were many scholars who had been attracted to carry on researches on this problem, and many mathematicians

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

had proven that any planar graph could be colored by four kinds of colors, which is called the four-color problem (i.e. four-color conjecture). The four-color problem was originally posed as a conjecture in 1850s. It was finally proved by the American mathematicians Appel and Haken in 1976. Coloring regions (whether these are states, countries, counties) in a map with a minimum number of colors such that neighboring regions (those sharing a common boundary) are colored differently has been proved to be a classic NP-complete problem. In this section, a brief overview of the planar graph coloring is addressed. The procedure of the planar graph coloring is described as follows:

Step 1: Transferring the map to a graph

It is not particularly difficult to show that the map can be colored with four-colors, that is, each region of the map can be assigned one of four given colors such that neighboring regions are colored differently. So, with each map, there is associated a graph G, called the dual of the map, whose vertices are the regions of the map and such that two vertices of G are adjacent if the corresponding regions are neighboring regions.

Step 2: Creating the adjacency matrix of graph

As we know, a graph G can be defined by two sets, namely its vertex set $V(G)$ and edge set $E(G)$ as described in Eqs. (1) and (2), respectively.

$$V(G) = \{v_1, v_2, v_3, \cdots, v_n\} \tag{1}$$

$$E(G) = \{e_1, e_2, e_3, \cdots, e_m\} \tag{2}$$

where n is the number of nodes and m is the number of edges. A graph can also be described by an adjacency matrix using Eq. (3).

$$a_{ij} = \begin{cases} 1, \text{ if } v_i v_j \in E(G); \\ 0, \text{ otherwise.} \end{cases} \tag{3}$$

Step 3: Specifying color number to the vertex

Coloring program of the adjacency matrix A of planar graph with n nodes $V = v_1, v_2, \cdots, v_n$ can be indicated as the coloring sequence $R = r_1, r_2, \cdots, r_n$, where $r_x \in R(1 \le x \le n)$ and $r_x \in \{0, 1, 2, 3\}$. In order to determine whether the coloring sequence R satisfies the conditions of the coloring program, we define the fitness function $f(R)$ and the conflict matrix Co, according to adjacency matrix as stated in Eqs. (4) and (5).

$$f(R) = \sum_{x=1}^{n} \sum_{y=1}^{n} conflict_{xy} \tag{4}$$

$$conflict_x y = \begin{cases} a_{xy} \text{ if } r_x = r_y \text{ and } x \neq y \\ 0 \text{ otherwise} \end{cases} \tag{5}$$

where $r_x, r_y \in R(1 \le x \le n, 1 \le y \le n)$, $a_{xy} \in A$ and $conflict_{xy}$ represents the coloring conflict between node v_x and node v_y, $f(R)$ represents the aggregate of node coloring in coloring sequence R.

Step 4: Adjust coloring number according to adjacency matrix

Obviously, the fitness function $f(R) \ge 0$ and $f(R)$ is an even number. In order to consider in such a way that no two adjacent vertices (i.e. regions) are of the same color, it corresponds to a reasonable program when the fitness function $f(R) = 0$. For $f(R) = 0$, the vertex must adjust the color number.

3. Improved Cuckoo Search

In order to describe the Cuckoo Search algorithm more clearly, let us briefly review the interesting breed behaviour of certain cuckoo species. Then, we will outline the basic ideas and steps of the proposed algorithm. Base Cuckoo Search algorithm, In this section, an improved Cuckoo Search algorithm for solving planar graph coloring problem is proposed.

3.1. Cuckoo Breeding Behaviour [15, 18]

In nature, cuckoo is fascinating birds, not only because of the beautiful sounds they can make, but also because of their aggressive reproduction strategy. Some species such as the ani and Guira cuckoos lay their eggs in communal nests, though they may remove others' eggs to increase the hatching probability of their own eggs. Quite a number of species engage the obligate brood parasitism by laying their eggs in the nests of other host birds (often other species). There are three basic types of brood parasitism: intraspecific brood parasitism, cooperative breeding, and nest takeover. Some host birds can engage direct conflict with the intruding cuckoos. If a host bird discovers the eggs are not their owns, they will either throw these alien eggs away or simply abandon its nest and build a new nest elsewhere. Some cuckoo species such as the New World brood-parasitic Tapera have evolved in such a way that female parasitic cuckoos are often very specialized in the mimicry in colour and pattern of the eggs of a few chosen host species. This reduces the probability of their eggs being abandoned and thus increases their reproductivity.

In addition, the timing of egg-laying of some species is also amazing. Parasitic cuckoos often choose a nest where the host bird just laid its own eggs. In general, the cuckoo eggs hatch slightly earlier than their host eggs. Once the first cuckoo chick is hatched, the first instinct action it will take is to evict the host eggs by blindly propelling the eggs out of the nest, which increases the cuckoo chicks share of food provided by its host bird. Studies also show that a cuckoo chick can also mimic the call of host chicks to gain access to more feeding opportunity.

3.2. Lévy Flight

On the other hand, various studies have shown that flight behaviour of many animals and insects has demonstrated the typical characteristics of Lévy flights [20]. A recent study by Reynolds and Frye shows that fruit flies or Drosophila melanogaster, explore their landscape using a series of straight flight paths punctuated by a sudden 900 turn, leading to a Lévy-flight-style intermittent scale free search pattern. Studies on human behaviour such as the Ju/'hoansi hunter-gatherer foraging patterns also show the typical feature of Lévy flights. Even light can be related to Lévy flights [20]. Subsequently, such behaviour has been applied to optimization and optimal search, and preliminary results show its promising capability.

Furthermore, various studies have shown that ?ight behaviour of many animals and insects has demonstrated the typical characteristics of Lévy lights. A recent study by Reynolds and Frye shows ?ies or Drosophila melanogaster, explore their that fruit landscape using a series of straight ?ight paths punctuated by a sudden 90° turn, leading to a Lévy-flight-style intermittent scale free search pattern [19]. Studies on human behaviour such as the Ju/'hoansi hunter-gatherer foraging patterns also show the typical feature of Lévy flights [20]. The conclusion that light is related to Lévy flights is proposed by P.Barthelemy, etc (2008) [21]. The study by Mercadier etc. shows that the Lévy flights of photons in hot atomic vapours (2009) [22]. Subsequently, such behaviour has been applied to optimization and optimal search, and preliminary results show its promising capability.

3.3. Basic CS

Cuckoo Search (CS) is an efficient, robust and simple optimization algorithm for solving many continuous optimization problems. Aiming at the particularity of the discrete space optimization problem such as the graph coloring, we adapt the quaternary CS to our problem and introduce the walking one strategy, swap and inversion strategy and the greedy transform algorithm.

CS is a heuristic search algorithm which has been proposed recently by Yang and Deb [15]. The algorithm is inspired by the reproduction strategy of cuckoos. At the most basic level, cuckoos lay their eggs in the nests of other host birds, which may be of different species. The host bird may discover that the eggs are not its own and either destroy the egg or abandon the nest all together. This has resulted in the evolution of cuckoo eggs which mimic the eggs of local host birds. To apply this as an optimization tool, Yang and Deb [16] used three ideal rules:

– Each cuckoo lays one egg, which represents a set of solution co-ordinates, at a time and dumps it in a random nest;
– A fraction of the nests containing the best eggs, or solutions, will carry over to the next generation;
– The number of nests is fixed and there is a probability that a host can discover an alien egg. If this happens, the host can either discard the egg or the nest and this result in building a new nest in a new location.

Based on these three rules, the basic steps of the Cuckoo Search (CS) can be summarized as the pseudo code as follow [16]:

---

begin
  Objective function $f(x), x = (x_1, x_2, \cdots, x_d)^T$;
  Generate initial population of
  n host nests $x_i (i = 1, 2, \ldots, n)$.
while $(t < MaxGeneration)$ or $(stop\ criterion)$
  Get a cuckoo randomly by Lévy flights;
  evaluate its quality/fitness F_i;
  Choose a nest among n (say, j) randomly.
if $(F_i > F_j)$
  replace j by the new solution;
end
  A fraction (p_a) of worse nests;
  are abandoned and new ones are built;
  Keep the best solutions
  (or nests with quality solutions);
  Rank the solutions and find the current best.
end while
  Postprocess results and visualization.
end

---

when generating new solution $x^{(t+1)}$ for, say cuckoo i, a Lévy flight is performed.

$$x_i^{(t+1)} = x_i^{(t)} + \partial \oplus L\acute{e}vy(\beta) \quad (6)$$

where ∂ is the step size which should be related to the scales of the problem of interests. In most cases, we can use $\partial = 1$. The above equation is essentially the stochastic equation for random walk. In general, a random walk is a Markov chain whose next status/location only depends on the current location (the first term in the above equation) and the transition probability (the second term). The product means entrywise multiplications. This entrywise product is similar to those used in PSO, but here therandom walk via Lévy flight is more efficient in exploring the search space as its step length is much longer in the long run.

From a quick look, it seems that there is some similarity between CS and hill-climbing in combination with some large scale randomization. But there are some significant differences. Firstly, CS is a population-based algorithm, in a way similar to GA and PSO,GSO (glowworm search optimization), but it uses some sort of elitism and/or selection similar to that used in harmony search. Secondly, the randomization is more efficient as the step length is heavy-tailed, and any large step is possible. Thirdly, the number of parameters to be tuned is less than GA and PSO, GSO, and thus it is potentially more generic to adapt to a wider class of optimization problems. In addition, each nest can represent a set of solutions, CS can thus be extended to the type of meta-population algorithm.

The product $\oplus$ means entry-wise walk while multiplications. Lévy flights essentially provide a random walk while their random steps are drawn from a Lévy Distribution for large steps

$$L\acute{e}vy \sim \mu = t^{-1-\beta}(0 \leq \beta \leq 2) \tag{7}$$

This has an infinite variance with an infinite mean. Here the consecutive jumps/steps of a cuckoo essentially form a random walk process which obeys a power-law step-length distribution with a heavy tail. In addition, a fraction pa of the worst nests can be abandoned so that new nests can be built at new locations by random walks and mixing. The mixing of the eggs/solutions can be performed by random permutation according to the similarity/difference to the host eggs.

Obviously, the generation of step size s samples is not trivial using Lévy flights. A simple scheme discussed in detail by Yang can be summarized as

$$x_i^{(t+1)} = x_i^{(t)} + \partial \oplus L\acute{e}vy(\beta)$$

$$L\acute{e}vy(\beta) \sim 0.01 \frac{\mu}{|\nu|^{1/\beta}}(x_i^{(t)} - x_j^{(t)}) \tag{8}$$

where μ and ν are drawn from normal distributions. That is

$$\mu \sim N(0, \sigma_\mu^2), \quad \nu \sim N(0, \sigma_\nu^2) \tag{9}$$

With $\sigma_\mu = \{\frac{\Gamma(1+\beta)sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}}\}^{1/\beta}$, $\sigma_\nu = 1$. Here where Γ is the standard Gamma function.

3.4. The New Planar Graph Coloring Model

A new planar graph coloring, named ICS, based on the cuckoo search. The ICS model is inspired by the literature [10], which employs the walking one strategy, swap and inversion strategy and greedy transform algorithm, is proposed in this paper.

3.4.1. Walking One Strategy

Cui et al. (2008) developed a quaternary-valued PSO method by defining the particles positions and velocities. The walking one strategy is a probability function based on quaternary-valued PSO; it depends on the node and adjacent nodes of the number of the conflicts. The number of the conflict nodes could get from Eq. (10) based on Eq. (5). Then the number of the conflict nodes will be converted to collision factor through the sigmoid function (i.e. Eq. (11)). If the collision factor is large, then it will be assigned a higher probability to change its color number. Otherwise, it will

0	1	2	**3**	0	1	**2**

Parent

0	1	2	**2**	0	1	**3**

Offspring

Figure 1 An example of swap node

0	1	2	3	**0**	**1**	**2**

Parent

0	1	2	3	**2**	**1**	**0**

Offspring

Figure 2 An example of inverse node

be assigned a lower probability. (i.e. Eqs. (12) and (13)). The walking one strategy is shown as follows:

$$Cr_j = \Sigma_{k=1}^{n} conflict_{jk} \tag{10}$$

$$Cf_j = \frac{1}{1+e^{-Cr_j+2}} \tag{11}$$

$$M(V_j) = \begin{cases} 1, \text{ if } Cf_j > rand() \& S(V_j) > rand() \\ 0, \text{ otherwise} \end{cases} \tag{12}$$

$$P_j = mod((P_j + M(V_j)), 4) \tag{13}$$

where $Cr_j (1 \leq j \leq n)$ is the number of conflict nodes with jth node, and Cf_j is the collision factor of jth node in the range [0, 1], n is the number of nodes, $rand()$ is a uniformly distributed random number in the range [0, 1], $S(V_j)$ is the sigmoid function given by $S(\nu) = 1/(1 + e^{-v})$.

3.4.2. Swap and Inversion Strategy

See fig. 1 and fig. 2 please.

3.4.3. Greedy transform algorithm

In graph coloring, the maximum conflict node is the most troublesome and needs to be processed first. In this strategy, we will find the maximum conflict node. The strategy is described in algorithm 1 as follow:

Algorithm 1 (Greedy transform algorithm)

Calculate the conflict matrix cr;
Sort the cr matrix in descending order;

for $i, (1 \leq i \leq n)$ **do** % is the number of node
for color $c(0 \leq c \leq 3)$ **do**
Calculate the new conflict matrix;
(i.e.,fni)
if $fni = 0$ **then**
Exit for;
else if $fni < foi$ **then**
Accept the color;
endif
endfor
endfor

3.4.4. ICS Algorithm

The whole running procedure of the ICS is described in algorithm 2 as follow:

Algorithm 2 (ICS algorithm)

Initialize all nest's position
while the stop condition is not satisfied **do**
for nest **do**
Calculate the step size of nest i according to levy flight.
Calculate the Cr_i of nest according to Eq.(10).
Calculate the Cf_i of nest according to Eq.(11).
Calculate the $M(V_i)$ of nest i according to Eq.(12).
Calculate the P_i of nest i according to Eq.(13).
Greedy transform algorithm.
Evaluate its quality/fitness F_i.
Swap node.
Evaluate its quality/fitness F_i.
Inverse node.
Evaluate its quality/fitness F_i;
Keep the best solutions (or nests with quality solutions);
Rank the solutions and find the current best;
endfor
endwhile

4. Experimental Results

In this section, the performance of the ICS algorithm is extensively investigated by a large number of experimental studies. All computational experiments are conducted with Matlab R2010a, and run on Celeron (R) Dual-core CPU T3100, 1.90GHZ with 2GB memory capacity. The essential parameters of ICS model for the planar graph coloring

Table 1. Comparison of the experimental results

Node	Algorithm	Maximal iterations	Minimal iterations	Average iterations	Correct coloring rate
7	**ICS**	**0**	**0**	**0**	**100**
	MPSO	7	0	5	100
	PSO	8	0	6	100
10	**ICS**	**3**	**0**	**0.88**	**100**
	MPSO	47	5	26	100
	PSO	66	6	32	100
20	**ICS**	**20**	**7**	**13.5**	**100**
	MPSO	4637	36	1569	59.7
	PSO	5895	116	2418	42.3
30	**ICS**	**67**	**27**	**42.4**	**100**
	MPSO	10000	1786	5439	16.4
	PSO	10000	3426	6432	10.9
54	**ICS**	**446**	**192**	**317.6**	**100**
	MPSO	-	-	-	-
	PSO	-	-	-	-
100	**ICS**	**5761**	**3214**	**4352.7**	**100**
	MPSO	-	-	-	-
	PSO	-	-	-	-

are set as follows. We simulated 100 runs (nodes 30) and 10 runs (nodes¿30). Let the maximal number of iterations be 10000, the number of nests be 25.

In order to validate the results of the ICS algorithms presented in this paper to compare the performance of the ICS algorithm with improved MPSO algorithm and the original PSO algorithm (the number of particles as 200). By randomly generating a given scaled planar graph and calculating 100/10 times to ICS, the experimental results we obtained are shown in Table 1. Here,-represents no records in the literature. It can be seen that ICS has a faster convergent velocity and a better global search capability to solve the same problem. In order to demonstrate the performance of ICS algorithm for solving the planar graph coloring problem further, we take the coloring problem of the map of China as an example. Fig.3 is the map of China, including 32 provinces, municipalities and autonomous regions, of which incidence matrix is a symmetric matrix of 3232. We, respectively, use PSO, the improved MPSO and ICS to perform the experiment and simulation. The results demonstrate that the ICS method is able to present a coloring scheme to this problem in a feasible time. Figure 3 to Figure 4 is also a specific description of one solution for solving its 4-coloring problem using ICS algorithm.

The best individual color coding 2 3 1 0 2 0 3 0 1 2 0 2 1 0 2 1 0 2 0 3 1 1 3 0 2 1 0 0 1 2 0 2.

5. Conclusions

In terms of the discrete space optimization problem such as the graph coloring, in this paper, we proposed a modified CS algorithm. A greedy transform algorithm is added to a cuckoo search algorithm for improving CS algorithms

Figure 3 The map of China administrative region Numbers

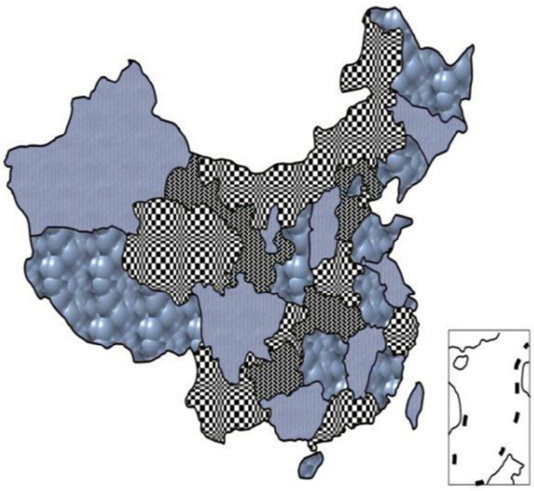

Figure 4 The effect coloring

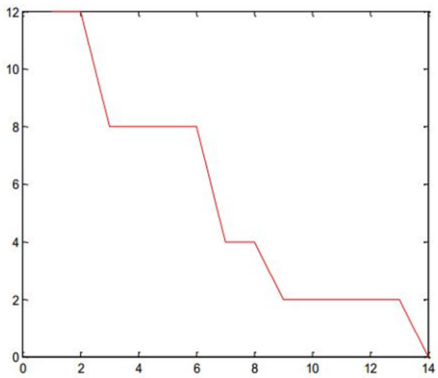

Figure 5 The convergence curve of coloring map

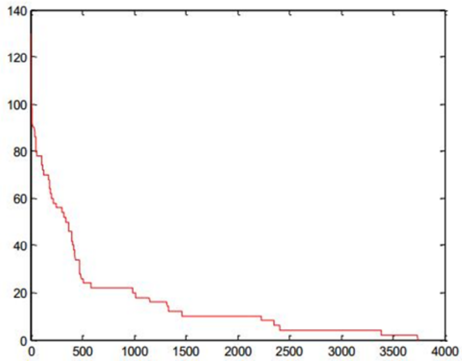

Figure 6 The convergence curve of coloring 100 regions

performance. The experimental results show that this algorithm is considerably effective to the graph coloring problem with moderate size, and the ICS is more efficient and accurate than the modified PSO algorithm proposed by Cui et al. (2008), however, with the increasing scale of the problem, the nests iterations need increase and they are more and more difficult to jump out of the approximate best solution. Generally speaking, the larger scale of the nest swarms needs more evolutionary time so the more quantities of the best solutions are required, and the computational complexity will subsequently increase greatly. Therefore, it is necessary to continually find a more effective coloring algorithm to the larger scale coloring problem. This will be the direction in the future research.

Acknowledgements

This work is supported by National Science Foundation of China under Grant No. 61165015. Key Project of Guangxi Science Foundation under Grant No. 2012GXNSFDA053028, 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.

References

[1] Bui, T. N., Nguyen, T. H., Patel, C. M., & Phan, K.-A. T. *An ant-based algorithm for coloring graphs.* Discrete Applied Mathematics,2008, 156: 190-200.

[2] Blochliger, I., Zufferey, N. *A graph coloring heuristic using partial solutions and a reactive tabu scheme.* Computers & Operations Research, 2008,35:960-975.

[3] Chaitin, G. J., Auslander, M. A., Chandra, A. K., Cocke, J., Hopkins, M. E., & Markstein, P. W. *Register allocation via coloring.* Computer Languages, 1981,6:47-57.

[4] Dowsland, K. A., & Thompson, J. M. *An improved ant colony optimization heuristic for graph colouring*. Discrete Applied Mathematics, 2008,156:313-324.
[5] de Werra, D. *An introduction to timetabling.* European Journal of Operational Research, 1985,19:151-162.
[6] Guangzhao Cui, Limin Qin, Sha Liu, Yanfeng Wang, Xuncai Zhang, Xianghong Cao. *Modified PSO algorithmfor solving planar graph coloring problem.* Progress in Natural Science, 2008, 18:353-357.
[7] Galinier, P., Hertz, A. A survey of local search methods for graph coloring. Computers & Operations Research, 2006, 33:2547-2562.
[8] Gamst, A. *Some lower bounds for a class of frequency assignment problems.* IEEE Transactions on Vehicular Technology, 1986,35:8-14.
[9] Garey, M. R., Johnson, D. S., & So, H. C. *Application of graph coloring to printed circuit testing.* IEEE Transactions on Circuits Systems, 1976, CAS-23, 591-599.
[10] Ling-Yuan Hsu, Shi-Jinn Horng, Pingzhi Fan, Muhammad Khurram Khan, Yuh-Rau Wang, Ray-Shine Run, Jui-Lin Lai, Rong-Jian Chen. *MTPSO algorithm for solving planar graph coloring problem.* Expert systems with Applications. 2011,38: 5525-5531.
[11] Osman, I. H. *A tabu search procedure based on a random Roulette diversification for the weighted maximal planar graph problem.* Computers & Operations Research, 2006, 33: 2526-2546.
[12] Salari, E. Eshghi, K. *An ACO algorithm for graph coloring problem.* Int. J. Contemp. Math. Sciences, 2008, 6: 293-304 .
[13] Sang Hyuck, A., Seung Gwan, L., & Tae Choong, C. *Modified ant colony system for coloring graphs.* In Proceedings of the 2003 joint conference of the fourth international conference on Information, communications and signal processing, 2003, 3:1849-1853.
[14] Talavan, P. M., Yanez, J.. *The graph coloring problem: A neuronal network approach.* European Journal of Operational Research, 2008, 191:100-111.
[15] Xinshe Yang Suash Deb. *Cuckoo Search via Lévy flight.* In Proc. of World Congress on Nature & Biologically Inspired Computing, December 2009, India. pp.210-214 .
[16] Yang X S, Deb S. *Engineering optimization by cuckoo search.* Int. J. Math Modeling & Num Optimization, 2010, 4:330-343.
[17] Yanez, J., & Ramirez, J. *The robust coloring problem.* European Journal of Operational Research, 2003, 148: 546-558.
[18] Payne R. B., Sorenson M. D., and Klitz K., *The Cuckoos,* Oxford University Press, 2005.
[19] Ahmed Younes. *Towards More Reliable Fixed Phase Quantum Search Algorithm.* Applied Mathematics & Information Sciences, 2013, 1, 93-98.
[20] Reynolds A. M. and Frye M. *A. Free-?ight odortracking in Drosophila is consistent with an optimal intermittent scale-free search,* PLoS One, 2007,2, e354.
[21] Pierre Barthelemy, Jacopo Bertolotti, Diederik S. Wiersma. *A Lévy Flight for Light,* Nature, 2008,453:495-498.
[22] N.Mercadier,W. Guerin,M. Chevrollier, R. Kaiser. *Lévy flights of photons in hot atomic vapours,* Nature Physics 2009,5: 602-605.
[23] Brown C., Liebovitch L. S., Glendon R. *Lévy flights in Dobe Ju/'hoansi foraging patterns,* Human Ecol., 2007, 35, 129-138.

Yongquan Zhou, Ph.D. Professor. His received the MS degree in Computer science from Lanzhou University in 1993, and the Ph. D degree in Computation Intelligence from the Xiandian University in 2006. He is currently a professor in Guangxi University for Nationalities, and Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis. His research interests are in the areas of computation intelligence, neural networks, and intelligence information processing.

Hongqing Zheng, MS. His research interests are in the areas of swarm intelligence theory and method, neural networks, and information systems.

Qifang Luo, Associate Professor. Her received the MS degree in computer science from Guangxi University in 2005, Her research interests are in the areas of computation intelligence, and intelligence information systems.

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 Peking University in 1995. His is currently a professor in Guangxi University for Nationalities. His research interests are in the areas of hybrid computation and Integrated circuit validation.