Int. J. Contemp. Math. Sciences, Vol. 6, 2011, no. 1, 41 - 49

Proposing a New Algorithm Based on Bees Behavior for Solving Graph Coloring

Majid Faraji

Computer Engineering Department
Science and Research Branch, Islamic Azad University, Tehran, Iran
majid.faraji61@yahoo.com

H. Haj Seyyed Javadi

Department of Mathematics and Computer Science
Shahed University, Tehran, Iran
h.s.javadi@shahed.ac.ir

Abstract

In this paper, we propose a new algorithm for graph coloring base on Bees behavior in nature (BEECOL). Finding a better conclusion in comparison to an existing Ant Colony Optimization (ACO) algorithm is the main goal of this approach. Experimental results on DIMACS test instances show improvements over an existing ACO Algorithm for the graph coloring problem.

Keywords: Graph coloring, ant colony optimization, graph coloring base on Bees behavior

I. Introduction

The graph coloring problem (COL) is a well-known problem in Combinatorial Optimization. Graph coloring problem is expected to have a wide variety of applications such as scheduling [5, 9], frequency assignment in cellular networks [6], timetabling [4], crew assignment [10], etc.

The first algorithms for graph coloring were developed in the 1960s [2, 3,14]. To solving Graph Coloring two classes of algorithms are available: exact and approximate. Exact algorithms are used for solving small size of instances and approximate algorithms obtain near-optimal solutions at relatively low computational costs. Most of the approximate algorithms for solving graph coloring are due to metaheuristics implementations such as Simulated Annealing, Tabu Search, Genetic Algorithm, Ant Colony Optimization, etc[11].

It was proposed by Salari And Eshghi as a Max-Min ant system algorithm for Graph Coloring (MMGC) [11]. In this paper, we propose a new algorithm for graph coloring base on Bees behavior (BEECOL). Computational results on DIMACS test instances [12] demonstrate that BEECOL outperforms MMGC.
The behavior of Bees will be studied in section II, and then we start to define the problem that is on section III, and describe how the coloring function in section IV is, and BEECOL algorithm is studied next in section V, in section VI we have focused on numerical conclusion, after that we start to comparison in section VII and finally the conclusion will be discussed on section VIII.

II. Bees behavior

In the nature, there are interactive relations in the species of animals specially insects, which contribute to the collective intelligence of the social group. One of the examples of such interactive behavior is the waggle dance of bees during the food procuring. By performing this dance, successful foragers share the information about the direction and distance to patches of flower and the amount of nectar within this flower with their hive mates. So this is a successful mechanism which foragers can recruit other bees in their colony to productive locations to collect various resources. Bee colony can quickly and precisely adjust its searching pattern in time and space according to changing nectar sources. The information exchange among individual insects is the most important part of the collective knowledge. Communication among bees about the quality of food sources is being achieved in the dancing area by performing waggle dance.
The previous studies on dancing behavior of bees show that while performing the waggle dance, the direction of bees indicates the direction of the food source in relation to the Sun, the intensity of the waggles indicates how far away it is and the duration of the dance indicates the amount of nectar on related food source. Waggle dancing bees that have been in the hive for an extended time, adjust the angles of their dances to accommodate the changing direction of the sun. Therefore bees that follow the waggle run of the dance are still correctly led to the food source even though its angle relative to the sun has changed. So collective intelligence of bees, based on the synergistic information, exchange during waggle dance[1,8,13].

III. Problem Definition

Let $G = (V, E)$ be an undirected graph where V is the set of vertices and E is the set of edges and N is number of vertices. An independent set is a subset of vertices in which no pair of adjacent vertices exists. A k-coloring of G is a mapping $c : V \rightarrow \{1, 2, 3, \ldots, k\}$ that assigns colors to vertices. The coloring is feasible if no two adjacent vertices has the same color, i.e. $\forall \{u,v\} \in E : c(u) \neq c(v)$, otherwise conflicts happen. A coloring with at least one conflict is called an infeasible coloring. An optimal coloring of G is a feasible coloring with smallest number of colors. This minimum number of colors k for which a feasible k-coloring exists is called the chromatic number of G and is denoted by $\chi(G)$. Given a graph G, the graph coloring problem is to find an optimal coloring[11].

IV. Coloring Function

If $deg(V_a)=m$ and the adjacent nodes of V_a are: $\{V_1,V_2,\ldots,V_m\}$ and the set of adjacent nodes' colors are: $C=\{c(V_1),c(V_2),\ldots,c(V_m)\}$ then:

```
function Coloring Vertex (V_a)
    for j = 1 to N
        if j ∉ C then
          Begin
             c(V_a) = j;
             exit;
          end
```

Which $c(V_a)$ is the color of V_a . The smallest color which does not exist in C, is considered by this function for V_a.

V.BEECOL Algorithm

Two different kinds of bees are used in this algorithm. Scout Bees and Employed Bees.

(a) *Employed Bees.* Bees which have found better results in compare with their counterparts are called employed bees. They are the only bees which dance in the dancing area. So the number of employed bees is exactly equal to the capacity of the dancing area. Besides proposing their path in the dancing area, they try to find better path near their own path too.

(b) *Scout Bees.* Bees which search for new paths in search space, using only their own deduction, are called scout bees.

First of all, *sbee* number of scout bees starting to make primary solutions. Then the quality of scout bees' solutions will be evaluated and *ebee* number of scout bees who have offered better solutions will convert to employed bees.

The factor *ebee* is the number of employed bees and also is equal to dance area capacity. Employed bees start to waggle dance and share their information with the other employed bees. Each bee memorizes the other bees' information.

The offered algorithms' pseudo code is:

1. Sending scout bees for making primary solutions.

2. Evaluating the fitness of scout bees' solutions and selecting ebee number of scout bees who have presented better solutions and changing them to employed bees.

While (stopping criterion not met)

3. Waggle dancing of employed bees and sharing information and memorizing information of all bees.
4. Neighborhood Search.
5. Updating information.

1. Making primary solutions:

The number of scout bees (*sbee*) is equal to the number of graph's nodes (*N*). First each of the scout bees is located in a graph's node randomly. Function Coloring Vertex colors that node. The next node is selected by probability function (1) and is colored by Coloring Vertex function. This procedure is repeated while all nodes have been colored. At the end of this phase *N* (number of graph nodes) solutions have been made.

$$P_{V_i,S} = \begin{cases} 0 & V_i \in W_S \\ \frac{deg\,(V_i)}{\sum_{i=1,V_i \notin W_S}^{N} deg\,(V_i)} & V_i \notin W_S \end{cases} \quad (1)$$

$P_{V_{i,S}}$ is the probability of selecting node V_i by scout bee S and W_S is the set of colored nodes by the scout bee S and N is the number of graph's nodes.

2. Evaluating the fitness of solution:

To evaluate the quality of each solution we consider a fitness value for each one, which is calculated as bellows:

$$fit(S) = \frac{1}{k(S)}$$

k(S) is the number of colors which have been used to coloring by bee S. *fit(S)* is the fitness value of the solution of bee S. *ebee* numbers of scout bees who have presented better solutions are being chosen and changing to employed bees. The number of employed bees (*ebee*) is equal to the dance area capacity.

3. Waggle dance and sharing information:

The employed bees start waggle dance and share their information to the other bees. Each employed bees memorizes all given information. Due to keeping same information of solutions by every bee, the information is the same, so just one memory is considered for whole colony. This memory consists of *N* room and each room is saves information of a node. Value of each room is:

$$\tau_{V_i} = \sum_{S=1}^{ebee} \frac{\varphi j_{V_i,S}}{c_s(V_i)}$$

$j_{V_i,S}$ Means that V_i is the j^{th} colored node of presented solution of bee S
$c_S(V_i)$ is the color of node V_i in presented solution of bee S

ebee is the number of employed bees
φ is 0.5
τ_{V_i} is the value of memory for node V_i

4. Neighborhood Search:

Def1. If $deg(V_a)=m$ and the adjacent nodes of V_a are: $\{V_1, V_2, \ldots, V_m\}$ then :

$$Hdeg(V_a)=deg(V_1)+deg(V_2)+\ldots+deg(V_m)$$

Def2. If a set of nodes of solution that presented by bee *S* which have a same color such as *T* is: $\{V_1, V_2, \ldots, V_{D_T}\}$ then:

$$Q_{S_T} = \sum_{i=1}^{D_T} Hdeg(V_i)$$

D_T is the number of nodes which have been colored by color *T*.

Def3. If a set of nodes of solution that presented by bee *S* which have a same color such as *T* is: $\{V_1, V_2, \ldots, V_{D_T}\}$ then:

$$\lambda_{S_T} = \sum_{i=1}^{D_T} \tau_{V_i}$$

D_T is the number of nodes which have been colored by color *T*.

Def4. If *N* is the number of graph's nodes and *k(S)* is the number of colors which is been used by bee *S* to color graph then

$$F_S = \frac{\left(\frac{N}{k(S)}\right)}{3}$$

Def5. If the solution of bee *S* has *k(S)* colors, then:

$$X_{S_i} = \begin{cases} Q_{S_i} - Q_{S_{i+1}} & 0 < i < k(S) \\ Q_{S_i} - Q_{S_1} & i = k(S) \end{cases}$$

In this phase each employed bee is searching the neighborhood of his solution. First for all colors which is used in the solution of bee *S*, we establish X_{S_i} ($i \in \{1,2, \ldots, k\}$).

If $d = max(X_{S_i})$ then *i* color will be selected.

Phase 1: if $i>1$, $D_{i-1}>F_S$, $i-1 \notin b_S$ then all nodes which have the color *i-1* in the solution of bee *S* will be colored by function Coloring Vertex and *i-1* will be added to set b_S. Set b_S is specified the colors which have been used for searching the neighborhood by bee *S*.

Phase 2: if $i<k(S)$, $D_{i+1}>F_S$, $i+1 \notin b_S$ then all nodes in the solution of bee *S* which have color *i+1* will be colored by function Coloring Vertex and *i+1* will be added to set b_S.

Phase 3: if $i \notin b_S$ then all nodes in the solution of bee *S* which have color *i* will be colored by function Coloring Vertex and *i* will be added to set b_S.

Phase 4: next color is calculated by following probability function.

$$P_{S_i} = \begin{cases} 0 & i \in b_S \\ \dfrac{(Q_{S_i})^{\alpha}(\lambda_{S_i})^{\beta}}{\sum_{j=1, j \notin b_S}^{k(S)} (Q_{S_j})^{\alpha}(\lambda_{S_j})^{\beta}} & i \notin b_S \end{cases}$$

P_{S_i} is the probability of selecting color *i* in the solution of bee *S*
k(S) is the number of used color for coloring graph
b_S is the used color for searching the neighborhood by bee *S*
α and β are two positive integer
Q comes from **Def2.** And λ comes from **Def3.**

After acquiring *i*, return to Phase 1 and start to repeat phase 1 to 4 till all nodes of graph color and neighborhood solution of main solution will made.

5. Updating information:

In this phase there is a comparison between fitness of neighborhood solution and fitness of main solution and if the fitness of neighborhood solution is better than or equal to fitness of main solution, values of main solution will be eliminated from colony and neighborhood solution will be considered as a main solution of bee.

VI. Numerical conclusion

Results of executing BEECOL algorithm on DIMACS sample graphs is shown in Table1. The algorithm is written in C# code and executed on PC Pentium 4 with 3.00 GHz processor and 1 GB RAM. Values of α and β is: α=5, β=3.

Table 1. Numerical conclusion BEECOL

Graph	Best *k*	*k*	*ebee*	No. Iterations	Time(Sec.)
r125.1	5	5	8	10	3
r125.1c	46	46	8	10	3
r125.5	36	36	20	400	13
r250.1	8	8	10	10	3
r250.1c	64	65	12	40	14
r1000.1	20	21	10	30	29
dsjc125.5	17	18	8	800	7
dsjc125.9	44	44	10	600	22
dsjc250.1	8	9	10	110	2
dsjc250.5	28	30	10	2900	153
dsjc500.1	12	14	10	150	9
dsjc500.5	48	53	4	3300	484
dsjr500.1	12	12	20	250	10
queen15_15	16	17	16	600	15
school1	14	14	8	1100	54
school1_nsh	14	14	8	650	24
le450_25a	25	25	10	10	3
le450_25b	25	25	10	10	3
le450_25c	25	27	17	12500	1200

First column declare graph's name, second column define the best known upper bound and the third column consist of a least color which offered algorithm has used to color graph. Forth column define the capacity of dance area or the number of employed bees (*ebee*). Fifth column show the number of repetition of algorithm and the sixth column is the executing time of algorithm in second.

VII. Comparison

In table2, acquired results of BEECOL algorithm has been compared with acquired results of MMGC algorithm[11]. As it is shown from table BEECOL algorithm make proper connection between speed and accuracy of graph coloring. The highlighted values in table specify improvement of BEECOL algorithm to MMGC algorithm.

Table 2. Comparison between BEECOL and MMGC

Graph	Best *k*	BEECOL		MMGC	
		k	Time(Sec.)	*k*	Time(Sec.)
dsjc125.5	17	18	**7**	18	186
dsjc125.9	44	44	**22**	44	133
dsjc250.1	8	9	**2**	9	650
dsjc250.5	28	30	**153**	30	736
dsjc500.1	12	**14**	**9**	15	3942
dsjc500.5	48	53	**484**	53	4131
queen15_15	16	17	**15**	17	845
school1_nsh	14	14	**24**	14	841
le450_25c	25	27	**1200**	27	4002

VIII. Conclusion

In this paper, an algorithm for solving problem of graph coloring based on bees' behavior in nature has been presented. Results of execution algorithm on sample graphs have shown us that this algorithm has the capability of establishing a proper connection between accuracy and speed of coloring the graph.

References

[1] F. Barth, Insects and flowers: The biology of a partnership, Princeton University Press, Princeton, New Jersey, 1982.

[2] J. Brown, Chromatic scheduling and the chromatic number problem, Management Science, 19(1972), 456-463.

[3] N. Christofides, An algorithm for the chromatic number of a graph, Computer Journal, 21(1971), 38-39.

[4] D. de Werra, An Introduction to Timetabling, European Journal of Operational Research, 19(1985), 151-162.

[5] K. A. Dowsland and J. M. Thompson, Ant Colony Optimization for the Examination Scheduling Problem, Journal of the Operational Research Society, 56(2005), 426-438.

[6] A. Gamst, Some Lower Bounds for a Class of Frequency Assignment Problem, IEEE Transactions of Vehicular Technology, 35(1999), 8-14.

[7] Graph Coloring : http://mat.gsia.cmu.edu/COLOR02, June 2005.

[8] R. L. Jeanne, The evolution of the organization of work in social insects, Monit. Zool. Ital. 20(1986) 267–287.

[9] F. T. Leighton, A Graph Coloring Algorithm for Large Scheduling Problems, Journal of Research of the National Bureau of Standards, 84(1979), 489-506.

[10] A. Lim and F. Wang, Metaheuristics for Robust Graph Coloring Problem, Proc. of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004), 2004.

[11] E. Salari and K. Eshghi, An ACO Algorithm for the Graph Coloring Problem. International Journal of Contemporary Mathematical Sciences, 3(2008), 293-304.

[12] E.C. Swell, An Improved Algorithm for Exact Graph Coloring, In D.S. Johnson and M. Trick editors, DIMACS Series in Computer Mathematics and Theoretical Computer Science AMS, 26(1996), 359-373.

[13] K. Von Frisch, The dance language and orientation of bees, Harvard University Press, Cambridge, Massachusetts, 1967.

[14] D. Welsh and M. Powell, An upper bound for the chromatic number of a graph and its application to timetabling problems, Computer Journal, 10(1967), 85-86.

Received: July, 2010