# An ACO Algorithm for Graph Coloring Problem

E. Salari, K. Eshghi
Department of Industrial Engineering
Sharif University of Technology, Tehran, Iran
e_salari@ie.sharif.edu, eshghi@sharif.edu

***Abstract-*** **Ant Colony Optimization (ACO) is a well-known metaheuristic in which a colony of artificial ants cooperate in exploring good solutions to a combinatorial optimization problem. In this paper, an ACO algorithm is presented for the graph coloring problem. This ACO algorithm conforms to Max-Min Ant System structure and exploits a local search heuristic to improve its performance. Experimental results on DIMACS test instances show improvements over existing ACO algorithms of the graph coloring problem.**

## I. Introduction

One of the most well-studied combinatorial optimization problem is the graph coloring problem. Given an undirected graph G=(V,E), the problem is to find a coloring of vertices with minimum number of colors such that no pair of adjacent vertices have the same color. Graph coloring problem is expected to have a wide variety of applications such as scheduling [1,2], frequency assignment in mobile networks [3], timetabling [4], crew assignment [5], etc.

Two classes of algorithms are available to solve this problem: exact and approximate algorithms. Since it has been proved that the graph coloring problem belongs to the class of NP-hard problems [6], exact algorithms [7,8,9] are confined to solve small size instances and as the problem size increases, the use of this class of algorithms quickly becomes infeasible, the only possibility is to resort to approximate algorithms in order to obtain near-optimal solutions at relatively low computational costs. Approximate algorithms for graph coloring problem can be classified into three main classes: constructive heuristics [10], local search heuristics [11], and metaheuristics. However, the majority of the approximate algorithms are due to metaheuristics implementations. Several metaheuristics are applied to graph coloring problem such as Simulated Annealing, Tabu Search, Genetic Algorithm, Ant Colony Optimization, etc. The first attempt to apply Simulated Annealing metaheuristic to graph coloring problem is due to Johnson et al [12]. Two Tabu Search algorithms are suggested by Hao and Dorne [13] and Hertz and Werra [14]. Fluerent and Ferland [15] likewise Costa et al [16] proposed two different Genetic Algorithm metaheuristics which are of the state-of-the-art algorithms for the graph coloring problem. Hao and Galinier [17] hybrid algorithm is yet another promising concept for the graph coloring problem, it uses Tabu Search as well as Genetic Algorithm to color a graph. Ant Colony Optimization [18] implementations in the context of the graph coloring problem are those of Costa and Hertz [10] which embed two graph coloring constructive heuristics RLF [1] and DSATUR [19]. Experimental results of their proposed algorithm, ANTCOL, were promising but far behind state-of-the-art algorithms. In this paper, a modification of ANTCOL, a Max-Min ant system [20] algorithm for Graph Coloring problem (MMGC), is proposed to improve ANTCOL performance. Computational results on DIMACS test instances [21] demonstrate that MMGC outperforms ANTCOL.

The remainder of this paper is organized as follows. In section II, graph coloring definition and notation is presented. we then in section III, review algorithm ANTCOL. ANCOL drawbacks as well as possible extensions are discussed in section IV. ANTCOL modification, MMGC, is proposed in section V and computational results are reported in section VI. Finally, the paper concludes in section VII.

## II. Problem Definition

Let $G=(V,E)$ be an undirected graph where V is the set of vertices and E is the set of edges. An independent set is a subset of vertices in which no pair of adjacent vertices exist. A q-coloring of G is a mapping $c:V\rightarrow\{1,2,3,...,q\}$ that assigns colors to vertices. The coloring is feasible if no two adjacent vertices have 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 q for which a feasible q-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. In addition to assignment representation above, the graph coloring problem can be formulated as a set-partitioning problem; a feasible q-coloring is a partitioning of set V into q independent sets, i.e.

$$V=\bigcup_{i=1}^{q}C_i\ \forall i,j,i\neq j:C_i\cap C_j=\phi \qquad (1)$$

Independent sets are also called color classes. By analogy, the objective is to partition the set V into minimum number of color classes.

## III. Antcol Algorithm

ANTCOL proposed by Costa and Hertz [10], is a metaheuristic to near-optimally solve the graph coloring problem. In ANTCOL a colony of artificial ants iteratively color a specific graph, at each iteration, initially, ants produce feasible colorings considering pheromone trails and heuristic information, and afterwards pheromone trails are updated according to the quality of colorings. The quality of colorings are measured using the following evaluation function.



1-4244-0020-1/05/$20.00 ã2005 IEEE

$$f(s) = \frac{1}{q(s)} \tag{2}$$

Where $q(s)$ denotes the number of colors applied in coloring s. Pheromone trails are related to pairs of nonadjacent vertices, therefore each pair of nonadjacent vertices $\{v_i, v_j\}$ have an associated pheromone trail $\tau_{ij}$ that represents the colony experience of colorings in which the two mentioned vertices have the same color, i.e. belong to the same color class. Artificial ants produce feasible colorings of the graph using modified versions of RLF and DSATUR, called ANTRLF and ANTDSATUR, respectively. Since ANTRLF outperforms ANTDSATUR, only the first one is described.

In ANTRLF, analogous to RLF, there exist several stages, at stage k the artificial ant constructs color class $C_k$, stage k also consists of several steps, at each step the artificial ant determines which uncolored vertex to be added to the color class $C_k$. Let W be the set of uncolored vertices that can be added to $C_k$, and B be the set of uncolored vertices which are not allowed to be added to $C_k$. In order to choose uncolored vertex $v_i$, ANTRLF uses three different heuristic information as follows.

$$\eta_{ik} = \deg_B(v_i) \tag{3}$$

$$\eta_{ik} = \deg_{B \cup W}(v_i) \tag{4}$$

$$\eta_{ik} = |W| - \deg_W(v_i) \tag{5}$$

Each heuristic information definition leads to a strategy in ANTRLF. However, at the beginning of stage k, there are no vertices in B, so the following two strategies are applied to add the first vertex to $C_k$.

- Randomly selecting an uncolored vertex from W.
- Selecting vertex $v_i$ with maximum $\deg_W(v_i)$.

Thus 6 different strategies are obtainable in ANTRLF via combining heuristic information strategies and the two above ones. ANTRLF experiments determined the selection of (3) and "Randomly selecting an uncolored vertex from W" as the best strategy.

Pheromone trails are initially set to 1 and at the end of each iteration, they become updated considering the following rule.

$$\tau_{ij} = (1-\rho)\tau_{ij} + \sum_{s \in S_{ij}} \frac{1}{q(s)} \tag{6}$$

Where $q(s)$ represents the number of colors applied to coloring s and $\rho$ denotes the pheromone evaporation rate. $S_{ij}$ is the subset of colorings in which the two nonadjacent vertices $v_i$ and $v_j$ belong to the same color class. In order to choose an uncolored vertex $v_i$ to be added to the color class $C_k$, pheromone trail $\tau_{ik}$ is defined as follows.

$$\tau_{ik} = \frac{\sum_{j \in C_k} \tau_{ij}}{|C_k|} \tag{7}$$

$\tau_{ik}$ contains all the pheromone trails between vertex $v_i$ and so far added vertices in color class $C_k$, in other words, it represents the colony experience of settling vertex $v_i$ with other vertices of $C_k$ in the same color class. Consequently, at each step of stage k, the probabilistic decision rule determines which uncolored vertex $v_i \in W$ to be added to the color class $C_k$ as follows.

$$P_{ik} = \begin{cases} \dfrac{\tau_{ik}^{\alpha} \eta_{ik}^{\beta}}{\sum_{j \in W} \tau_{jk}^{\alpha} \eta_{jk}^{\beta}} & v_i \in W \\ 0 & v_i \notin W \end{cases} \tag{8}$$

Where $p_{ik}$ is the probability of selecting vertex $v_i$. Stage k continues while W remains nonempty.

Adjusted parameters for ANTRLF are $\alpha = 2$, $\beta = 4$ and $\rho = 0.5$. The colony size is set to 100 ants and termination condition is defined as the number of iterations exceeds 50. ANTRLF applied to 4 samples of random graphs, experimental results were satisfactory but considerably outperformed by state-of-the-art algorithms.

## IV. Motivations

There are some drawbacks in ANTCOL which diminish its performance. In the following, these drawbacks are outlined.

Although ANTCOL exploits an ingenious pheromone trail definition, its pheromone updating rule (6) seems quite inefficient; regarding the fact that at each iteration all ants deposit pheromone, ANTCOL rapidly converges towards mediocre colorings. Generally speaking, all ACO algorithms which conform to Ant System structure suffer from stagnation. Subsequent ACO structures try to resolve this problem.

ANTCOL evaluation function (2) is not a suitable measure of colorings due to existence of a wide range of colorings with the same number of colors [2]. For example let $s^1$ and $s^2$ be two colorings of the specific graph with the following characteristics, they have equal number of color classes but the first one has a color class in which there exists only one vertex, whereas the second one has color classes in which vertices are evenly distributed, intuitively, $s^1$ seems more promising than $s^2$, but ANTCOL is incapable of distinguishing between them. Consequently, this leads to ineffective exploitation of explored colorings.

Most of the ACO algorithms utilize a local search heuristic to improve the obtained solutions in each iteration, Nevertheless, ANTCOL lacks an efficient local search which can contribute to better exploration of the solution space.

Finally, ANTCOL uses the probabilistic decision rule (8) to select an uncolored vertex to be added to the color class under construction. However, at the beginning of each stage, it uses a "randomly choosing" strategy that takes no advantages of neither pheromone trails nor heuristic information, although the color class under construction is noticeably affected by the first vertex chosen. Consequently, the first vertex should be selected more deliberately. Applying appropriate heuristic

information in selecting the first vertex may result in better performance of ANTCOL.
MMGC is a modification of ANTCOL with the aim of obviating the above drawbacks.

## V. MMGC ALGORITHM

Our proposed ACO algorithm, MMGC, conforms to Max-Min ant system structure. At each iteration a colony of artificial ants color a specific graph, and afterwards only the iteration-best ant deposits pheromone according to the quality of its coloring. In the following subsections, MMGC details are described.

### A. Coloring the Graph

Each artificial ant colors the vertices of the graph using a modified version of ANTRLF. By analogy, Its coloring consists of several stages. At each stage, a color class is built up. Each stage contains several steps. At each step, the artificial ant determines the uncolored vertex which to be added to the color class under construction using the probabilistic decision rule (8). Pheromone trails are related to pairs of nonadjacent vertices, the same as ANTCOL. Heuristic information (3) assists the artificial ant to select the proper uncolored vertex at each step. However, since at the beginning of stage k, color class $C_k$ is empty, a new probabilistic decision rule is defined to add the first vertex. This rule is just a function of the following heuristic information.

$$\eta_{ik} = \deg_W(v_i) \tag{9}$$

Hence the associated probabilistic decision rule is as follows.

$$P_{ik} = \begin{cases} \dfrac{\eta_{ik}^{\beta'}}{\sum_{j \in W} \eta_{jk}^{\beta'}} & v_i \in W \\ 0 & v_i \notin W \end{cases} \tag{10}$$

Where $p_{ik}$ denotes the probability of selecting vertex $v_i$ as the first vertex of $C_k$.

### B. Evaluating the Colorings

Johnson et al in [12] proposed a minimization evaluation function for graph coloring problem, as follows.

$$f(s) = -\sum_{i=1}^{q} |C_i|^2 + \sum_{i=1}^{q} 2|C_i||E(C_i)| \tag{11}$$

In which $E(C_i)$ represents the set of conflicts in the color class $C_i$. This evaluation function consists of two parts, as one tries to minimize it through an exploration, the first part biases the exploration towards colorings with fewer and bigger color classes and the second part towards colorings with fewer conflicts. They also showed that Local minimums of evaluation function (11) were pertinent to feasible colorings. Regarding a solution space comprises of only feasible colorings, the second part is eliminated and evaluation function (11) becomes:

$$f(s) = -\sum_{i=1}^{q} |C_i|^2 \tag{12}$$

Evaluation function (12) can widely demonstrate the differences between feasible colorings. However, smaller values of (12) does not necessarily result in fewer color classes, it biases towards unevenly distributed colorings rather than optimal ones, therefore it can not solely guide the exploration towards near-optimal colorings. In order to obviate the mentioned disadvantage, MMGC exploits number of color classes as well as the following maximization evaluation function to evaluate different colorings.

$$f(s) = \sum_{i=1}^{q} |C_i|^2 \tag{13}$$

At each iteration, ants colorings are prioritized according to non-decreasing order of number of color classes and ties are broken according to non-increasing order of evaluation function (13). Iteration-best ant is the one with prior coloring.

### C. Pheromone Trail Initialization and Limits

Considering Max-Min ant system structure, Pheromone trails are initialized to $|V|^2/\rho$, also, MMGC limits pheromone trails to interval $[\tau_{\min}, \tau_{\max}]$, where $\tau_{\min}$ and $\tau_{\max}$ are calculated according to the following equations.

$$\tau_{\max} = \frac{f(s^{gb})}{\rho} \tag{14}$$

$$\tau_{\min} = \frac{\tau_{\max}}{a} \tag{15}$$

$$a = \frac{(avg-1)\sqrt[n]{P_{best}}}{1-\sqrt[n]{P_{best}}} \tag{16}$$

In which, $s^{gb}$ denotes the global-best solution, and at the end of each iteration, pheromone trail limits are updated according to the so-far obtained $s^{gb}$. Since, on the average, there exist $|V|/2$ uncolored vertices to be selected at each step, parameter $avg$ equals to $|V|/2$ in (16) and $P_{best}$ represents the probability of constructing the convergence coloring.

### D. Applying Kempe Chain Local Search

At each iteration, MMGC improves Iteration-best coloring via Kempe chain local search heuristic [22]. The local search starts with an initial solution corresponding to iteration-best coloring and explores through Kempe chain neighborhood structure using a first improvement strategy. It uses the evaluation function (13) and holds the feasibility of colorings during exploration.

### E. Pheromone Trails Updating Rule

In MMGC, only iteration-best ant deposits pheromone considering the following pheromone updating rule.

$$\tau_{ij} = (1-\rho)\tau_{ij} + \sum_{i=1}^{q} |C_i|^2 \quad \forall i, j : v_i, v_j \in C_k \; k : 1,2,\ldots,q \tag{17}$$

In other words, regarding the iteration-best coloring, each pheromone trail, associated to a pair of nonadjacent vertices of the same color class, receives pheromone proportional to evaluation function (13).

## VI. Computational Results

Costa and Hertz applied ANTCOL to Fluerent and Ferland's randomly generated graphs and reported their results. We applied MMGC to DIMACS test instances and in order to obtain a fair comparison, we also reran ANTCOL on these test instances. Both algorithms were coded in Delphi 7 and run on a Pentium 4 PC with 1.8GHz CPU and 256MB RAM. Our experimental report consists of two parts; first, MMGC and ANTCOL running behaviors on DSJC $G_{250,0.5}$ random graph are presented, and then, the results of comparing MMGC to ANTCOL on other test instances are reported.

### A. MMGC and ANTCOL Running Behaviors

We ran MMGC as well as ANTCOL on DSJC $G_{250,0.5}$. Adjusted parameters for MMGC were as follows, $\alpha = 2$, $\beta = 4$, $\beta' = 3$, $\rho = 0.04$ and $P_{best} = 0.05$. The colony size was set to 20 ants and the termination condition was defined as the number of iterations exceeds 300. ANTCOL parameters were the same as Costa and Hertz's adjustments. Running results lead to the following points.

MMGC takes longer time to explore the solution space whereas ANTCOL rapidly converges towards mediocre colorings. So exploration phase in MMGC is much more longer than ANTCOL. The following plot depicts the MMGC and ANTCOL convergence behavior.

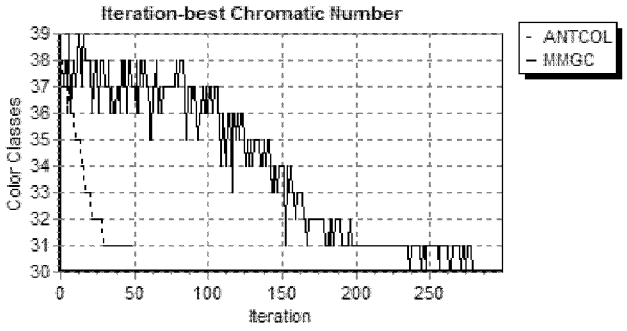


Fig. 1. Convergence Behavior of ANTCOL and MMGC

Besides, $\lambda$-Branching Factor can clearly show the pheromone distribution during MMGC and ANTCOL iterations. The following plot also verifies the long MMGC exploration phase.

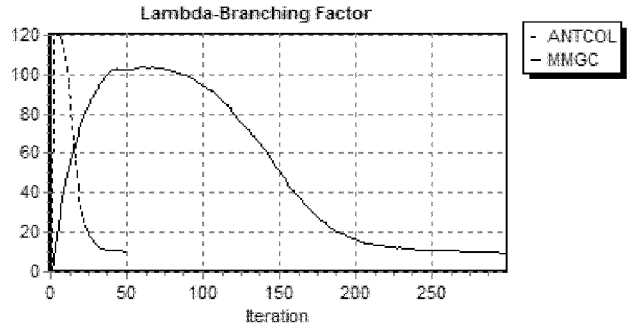


Fig. 2. $\lambda$-Branching Factor in ANTCOL and MMGC Iterations ($\lambda = 0.05$)

In MMGC, Kempe chain local search heuristic is applied to improve iteration-best colorings. This local search initially causes dramatic changes to colorings, but afterwards it becomes less effective. Kempe chain may even reduce the number of color classes at the beginning of trials. The plot below, illustrates the iteration-best evaluation function (13) value before and after applying Kempe chain local search heuristic.

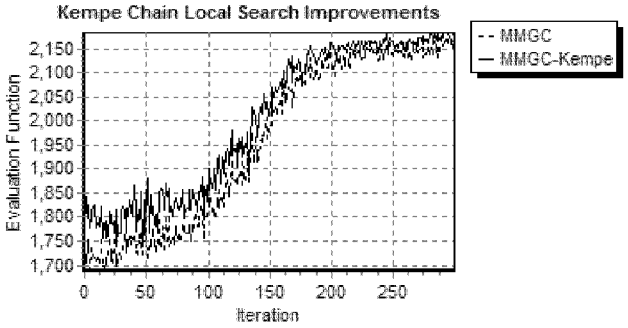


Fig. 3. Improving the Evaluation Function using Kempe Chain Local Search

### B. MMGC and ANTCOL Comparison

MMGC has been compared to ANTCOL using DSJC random graphs, Leighton, Queen and Class Scheduling graphs. We have also compared MMGC and ANTCOL with state-of-the-art algorithms. Results are presented in table I.

Since MMGC, the same as ANTCOL, utilizes ANTRLF as the solution construction procedure, by properly setting No. Iterations and No. Ants, it will nearly need the same computational time as ANTCOL does. Nonetheless, in some test instances MMGC relatively takes further running time due to applying Kempe chain local search heuristic.

MMGC outperforms ANTCOL with at least one less color in the majority of the DSJC random graphs. However, MMGC results do not approach best known results on some DSJC test instances particularly larger ones. Regarding Leighton graphs, for Le_450_15c, both algorithms found the optimal coloring and MMGC outperforms ANTCOL on Le_450_25c. MMGC comparatively colors Queen 15_15 with one less color, and in School1_nsh, both algorithms reached the best known result.

## VII. Conclusion

In this paper, a modification of an existing ACO algorithm for graph coloring problem is proposed. This modification conforms to Max-Min ant system structure and exploits a local search heuristic to improve its performance. It also uses a more promising evaluation function to distinguish between explored colorings. Experimental results show improvements over ANTCOL, however, it does not reach the best known results on some test instances. Further research attempts are required to utilize a more effective local search heuristic and to enhance the ants coloring phase through encouraging them to also produce infeasible colorings. It seems that producing only feasible colorings, during construction phase, leads to slender exploration. We are going to let ants produce partial infeasible colorings during construction phase and by applying

| Graph | Best-Known | MMGC | | | | ANTCOL | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | k | No. Iterations / No. Ants | No. Success / No. Trials | CPU time (Sec.) | k | No. Iterations / No. Ants | No. Success / No. Trials | CPU time (Sec.) |
| DSJC $G_{125,0.1}$ | 5 | **5** | 150 / 70 | 4/5 | 155 | 6 | 50 / 100 | 5/5 | 173 |
| DSJC $G_{125,0.5}$ | 17 | 18 | 150 / 70 | 5/5 | 186 | 18 | 50 / 100 | 4/5 | 125 |
| DSJC $G_{125,0.9}$ | 44 | 44 | 150 / 70 | 4/5 | 133 | 44 | 50 / 100 | 4/5 | 97 |
| DSJC $G_{250,0.1}$ | 8 | 9 | 300 / 20 | 5/5 | 650 | 9 | 50 / 100 | 5/5 | 713 |
| DSJC $G_{250,0.5}$ | 28 | **30** | 300 / 20 | 5/5 | 736 | 31 | 50 / 100 | 3/5 | 548 |
| DSJC $G_{250,0.9}$ | 72 | **74** | 300 / 20 | 4/5 | 713 | 75 | 50 / 100 | 3/5 | 421 |
| DSJC $G_{500,0.1}$ | 12 | 15 | 500 / 20 | 5/5 | 3942 | 15 | 50 / 100 | 5/5 | 3125 |
| DSJC $G_{500,0.5}$ | 48 | **53** | 500 / 20 | 4/5 | 4131 | 55 | 50 / 100 | 3/5 | 2638 |
| DSJC $G_{500,0.9}$ | 126 | **135** | 500 / 20 | 4/5 | 3063 | 136 | 50 / 100 | 4/5 | 2043 |
| Le_450_15c | 15 | 15 | 200 / 20 | 5/5 | 1809 | 15 | 50 / 100 | 5/5 | 2254 |
| Le_450_25c | 25 | **27** | 500 / 20 | 5/5 | 4002 | 29 | 50 / 300 | 5/5 | 4918 |
| Queen 15_15 | 16 | **17** | 275 / 20 | 3/5 | 845 | 18 | 50 / 100 | 4/5 | 624 |
| School1_nsh | 14 | 14 | 200 / 20 | 5/5 | 841 | 14 | 50 / 100 | 5/5 | 1316 |

TABLE I: MMGC comparison with ANTCOL and best known results of state-of-the-art algorithms. k denotes the global-best chromatic number found by the algorithms. 'No. Iterations/ No. Ants' represents termination condition and colony size, respectively. 'No. Success/No. Trials' demonstrates the number of trials out of 5 runs in which reported result is achieved.

a local search heuristic transform them into feasible ones and improve them.

## REFERENCES


[1] Leighton F.T., 1979, "A Graph Coloring Algorithm for Large Scheduling Problems", Journal of Research of the National Bureau of Standards, 85, 489-506.

[2] Dowsland K.A. and Thompson J.M., 2005, "Ant Colony Optimization for the Examination Scheduling Problem", Journal of the Operational Research Society, 56, 426-438(13).

[3] Gamst A., 1999, "Some Lower Bounds for a Class of Frequency Assignment Problem", IEEE Transactions of Vehicular Technology, 35(1), 8-14.

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

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

[6] Garey M.R. and Johnson, D.S., 1979, "Computer Science and Intractability", Freeman, San Francisco.

[7] Mehrotra A. and Trick M., 1996, "A column Generation Approach for Graph Coloring Problem", INFORMS Journal on Computing. 8(4), 344-354.

[8] Swell E.C., 1996, "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, 359-373.

[9] Caramia C. and Dell'Olmo P., 2002, "Vertex Coloring by Multistage Branch and Bound", in D.S. Johnson, A. Mehrotra, M. Trick editors, Proceedings of the Computational Symposium on Graph Coloring and its Generalization, Ithaca, New York, USA.

[10] Costa D. and Hertz A., 1997, "Ants Can Color Graphs", Journal of the Operational Research Society 48, 295-305.

[11] Chiarandini M., Dumitriscu I. and Stutzle T., 2003, "Local Search for the Graph Coloring Problem: A Computational Study", unpublished.

[12] Johnson D.S., Aragon C.R., McGeoch L.A. and Schevon C., 1991, "Optimization by Simulated Annealing: An Experimental Evaluation: Part II, Graph Coloring and Number Partitioning", Journal of Operations Research, 39(3), 378-406.

[13] Dorne R. and Hao J.K., 1999, "Tabu Search for Graph Coloring, t-colorings and Set t-colorings", In S. Voss, S. Martello, I.H. Osman and C. Roucairol editors, Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization, 77-92, Kluwer Academic Publishers, Boston, MA, USA.

[14] Hertz A. and de Werra D., 1987, "Using Tabu Search for Graph Coloring", Journal of Computing (39), 345-351.

[15] Fluerent C. and Ferland J., 1996, "Genetic and hybrid Algorithms for Graph Coloring Problem", Annals of Operations Research, 63:437-464.

[16] Costa D., Hertz A. and Dubuis O., 1995, "Embedding of a Sequential Procedure within an Evolutionary Algorithm", Journal of Heuristics (1), 105-128.

[17] Galinier P. and Hao J., 1999, "Hybrid Evolutionary Algorithms for Graph Coloring", Journal of Combinatorial Optimization, 3(4): 379-197.

[18] Dorigo M. and Stutzle T , 2004, "Ant Colony Optimization", MIT Press.

[19] Br'elaz D., 1979, "New Methods to Color the Vertices of a Graph", Communications of the ACM, 22(4), 251-256.

[20] Stutzle T. and Hoos H.H., 2000, "Max-Min Ant System", Journal of Future Generation Computer Systems, 16(8), 889-914.

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

[22] Morgenstern C., Shapiro H., 1990, "Coloration Neighborhood Structure for General Graph Coloring", in Proc. of the first annual ACM-SIAM Symposium on Discrete Algorithms, 226-235.