ICICS-PCM 2003
15-18 December 2003
Singapore

Modified Ant Colony System for Coloring Graphs

SangHyuck Ahn, SeungGwan Lee, TaeChoong Chung

Dept. of Computer Engineering School of Electronics and Information
KyungHee University
YongInSi GyongGido, 449-701, Korea.
Tel : +82 31 201 2952 Email : fisherking@iislab.kyunghee.ac.kr, tcchung@khu.ac.kr

Abstract

Ant Colony System (ACS) Algorithm is new meta-heuristic for hard combinational optimization problem. It is a population-based approach that uses exploitation of positive feedback as well as greedy search. Recently, various methods and solutions are proposed to solve optimal solution of graph coloring problem that assign different colors for adjacency node (vi, vj). This paper introduces ANTCOL algorithm to solve solution by Ant Colony System algorithm that is not known well as solution of existent graph coloring problem. After introducing ACS algorithm and Assignment Type Problem, it will be shown the way how to apply ACS to solve ATP. We propose ANT_XRLF method which uses XRLF to solve ANTCOL. Graph coloring result and execution time of our method are compared with existent generating functions (ANT Random, ANT_LF, ANT_SL, ANT_DSATUR, ANT_RLF method) Also we compare with existent generating functions with the method ANT_XRLF (ANT_XRLF_R) where re-search is added

Key Word: Ant Colony System (ACS), Graph Coloring, Assignment Type Problem (ATP)

1. Introduction

Ant algorithm introduced by Colorni, Dorigo and Maniezzo was first proposed for tackling the well known Travelling Salesman Problem (TSP). Two very similar algorithms were then proposed by the same authors for tackling the Quadratic Assignment Problem [3] and the Job Shop Problem [4]. Improving AS and formalizing it to general concept result in Ant Colony System algorithm that imitates action of ant to solve a large combinatorial optimization problem [5].

The problems we are interested in are those which can be formulated as Assignment Type Problems (ATPs). Given n items and m resources, the problem is to determine an assignment of the items to the resources optimizing a given objective function and satisfying a set of additional side constraints. The difficulty of these problems is related to the nature of the objective function and the additional constraints [6].

This paper introduces ANTCOL Algorithm that is to solve graph coloring problem using Ant Colony System algorithm. We propose ANT_XRLF method which uses XRLF to solve ANTCOL. Graph coloring result and execution time of our method are compared with existent generating functions (ANT_Random, ANT_LF, ANT_SL, ANT_DSATUR, ANT_RLF method) Also we compare with existent generating functions with the method ANT_XRLF (ANT_XRLF_R) where re-search is added.

2. Related Work

2.1 Graph Coloring Problem

The graph coloring problem can be formulated in the following way. A q-coloring of a graph G=(V, E) with vertex set V = {v1, ... , vn} and edge set E is a mapping c: V -> {1, 2, ... , q} such that c(vi) $\neq$ c(vj) whenever E contains an edge [vi, vj] linking the vertices vi and vj. The minimal number of colors q for which a q-coloring exists is called the chromatic number of G and is denoted χ (G). An optimal coloring is one which uses exactly χ (G) colors.

A mathematical formulation in terms of ATP is given below. The items are the vertices of V and the resources are the colors. Since it is always possible to color any graph G= (V, E) in n = |V| colors, number of items equal that of resources [7].

2.2 Ant Colony System

The Ant System - Ants work as follow: First, When Ants arrive at a decision point in which they have to decide to turn left or right, ants select randomly next path and deposit pheromone on the ground, since they have no clue about which is the best choice. After a short transitory period the difference in the amount of pheromone on the two paths are sufficiently large so as to influence the decision of new ants coming into the system. From now on, new ants will prefer in probability to choose the path with more pheromone.

As a consequence, Ants can smell pheromone and choose, in probability, paths marked by strong pheromone concentrations. Although ant system is useful for discovering good or optimal solutions for small TSP, the time required to find such results is unbearable for large size TSP problems. Therefore, ACS has been developed to improve its performance. The Ant Colony System - The ACS algorithm has been introduced by Dorigo and Gambardella to improve the performance of AS, that was able to find good solutions within a reasonable time only for small problem. Informally, ACS works as follows: m ants are initially positioned on n cities chosen according to some initialization rule (e.g., randomly). Each ant builds a tour (i.e., a feasible solution to the TSP) by repeatedly applying a stochastic greedy rule (the state transition rule).

0-7803-8185-8/03/$17.00 © 2003 IEEE

While constructing its tour, an ant also modifies the amount of pheromone on the visited edges by applying the local updating rule. Once all ants have terminated their tour, the amount of pheromone on edges is modified again (by applying the global updating rule). In the following we will discuss the state transition rule, the local updating rule, and the global updating rule.

2. 2. 1 The State Transition Rule

Let k be an agent whose task is to make a tour: visit all the cities and return to the starting one. Associated to k there is the list $J_k(r)$ of cities still to be visited, where r is the current city. An agent k situated in city r moves to city s using the follow rule, called pseudo-random proportional action choice rule(or state transition rule):

$$s = \begin{cases} \arg\max_{u \in J_k(r)} \{[\tau(r,u)] \cdot [\eta(r,u)]^{\beta}\} & if\ q \le q_0 \\ S & otherwise \end{cases} \quad (1)$$

Where τ(r, u) is the amount of pheromone trail on edge. η(r,u) is a heuristic function which is the inverse of the distance between cities r and u, β is a parameter which weighs the relative importance of pheromone trail ants, q is a value chosen randomly with uniform probability in [0,1], $q_0(0 \le q_0 \le 1)$ is a parameter, and S is a random variable selected according to the distribution given by Eq.(2) which gives the probability with which an agent in city r choose the city s to move to

$$p_k(r,s) = \begin{cases} \dfrac{[\tau(r,s)] \cdot [\eta(r,s)]^{\beta}}{\sum_{z \in J_k(r)} [\tau(r,u)] \cdot [\eta(r,u)]^{\beta}} & if\ S \in J_k(r) \\ 0 & otherwise \end{cases} \quad (2)$$

2. 2. 2 The Local Updating Rule

While building a solution of the TSP, ants visit edges and change their amount of pheromone trail by applying the following local updating rule:

$$\tau(r,s) \leftarrow (1-\rho) \cdot \tau(r,s) + \rho \cdot \Delta\tau(r,s) \quad (3)$$

Where ρ(0<ρ<1) is the pheromone decay parameter.

$\Delta\tau(r,s) = (n*L_{nn})^{-1}$ is the initial pheromone level, where L_{nn} is the tour length produced by the nearest neighbor heuristic and n is the number of nodes.

2. 2. 3 The Global Updating Rule

Global Updating is performed after all ants have completed their tours. The pheromone amount is updated by applying the follow global updating rule:

$$\tau(r,s) \leftarrow (1-\alpha) \cdot \tau(r,s) + \alpha \cdot \Delta\tau(r,s) \quad (4)$$

$$where\ \Delta\tau(r,s) = \begin{cases} (Lgb)^{-1}, if (r,s) \in global\_best\_tour \\ 0 \quad ,otherwise \end{cases}$$

0<α<1 is the pheromone decay parameter, and L_{gb} is the length of the globally best tour from the beginning of the trail.

2. 2. 4 ACS Algorithms for ATP

In an ordinary constructive method, these decisions are made in a myopic way by completing the current partial solution at best. In other words, they deal at each step with an item and a resource the desirability of which is maximum. At each stage k of the construction process (1 ≤ k ≤ n), an ant chooses an unassigned item i with probability pit (k, i) and a resource j, feasible for item i, with probability pre (k, i, j). Given a partial solution s[k-1] in which items o(1), ... , o(k-1) have already been assigned to resources and given a not yet assigned item i and a feasible resources j ∈ Ji for i, probabilities pit(k, i) and pre(k, i, j) are defined as follows[7].

$$p_{it}(k,\ i) = \frac{\tau_1(s[k-1],\ i) \cdot \eta_1(s[k-1],\ i)^{\beta_1}}{\sum_{o \notin \{o(1),\ \ldots,\ o(k-1)\}} \tau_1(s[k-1],\ o) \cdot \eta_1(s[k-1],\ o)^{\beta_1}}$$

$$p_{re}(k,\ i,\ j) = \frac{\tau_2(s[k-1],\ i,\ j) \cdot \eta_2(s[k-1],\ i,\ j)^{\beta_2}}{\sum_{r \in J_i} \tau_2(s[k-1],\ i,\ r) \cdot \eta_2(s[k-1],\ i,\ r)^{\beta_2}},$$

where τ1(s[k-1], i) and τ2(s[k-1], i, j) measure respectively the trail of item i and of resource j for i; η1(s[k-1], i) and η2(s[k-1], i, j) measure respectively the desirability of item i and of resource j for i.

2.3 ANTCOL

Consider the graph coloring problem presented. The objective is to find a q-coloring of G with q as small as possible. The role of each ant is to color the graph in a constructive way. The experience accumulated during the previous constructions is memorized in an n *n matrix M which is updated periodically. Given two non adjacent vertices vr and vs, the value Mrs in M is proportional to the quality of the colorings obtained by giving the same color to vr and vs. Hence, Mrs corresponds to a trail existing between vertices vr and vs. Initially, all values in M are set equal to 1. The trail stored in M evaporates progressively as time goes, the coefficient of evaporation being denoted by (1- ρ).

Let s1... snants be the colourings obtained at the end of a cycle and let Srs ⊆{s1... snants} denote the subset of solutions in which vr and vs have the same color. Finally, let qa be the number of colors used in sa (1 ≤ a ≤ nants).

The values *M*rs in M are updated as follows [7].

$$M_{rs} := \rho \cdot M_{rs} + \sum_{S_a \in S_{rs}} \frac{1}{q_a} \quad (5)$$

Given a partial solution s [k-1] = $\{v_1 \ldots v_n\}$, an uncolored vertex v and a feasible color c for v, the trail factor τ_2(s[k-1], v, c) is computed as follows

$$\tau_2(s[k-1],\ v,\ c) := \begin{cases} 1 & if\ V_c\ is\ empty \\ \dfrac{\sum_{x \in V_c} M_{xv}}{|V_c|} & otherwise \end{cases} \quad (6)$$

A sketch of ANTCOL is presented in <Table 1>

<Table 1> ANTCOL

$Mrs=1 \;\; \forall[v_r, v_s] \notin E$
/* initialization of the trail between pairs of non adjacent vertices */
$f^{\circ} = \infty$
/* number of colors in the best coloring reached so far */
For ncycles=1 to $ncycles_{max}$ *do*
$\Delta Mrs=0; \;\; \forall[v_r, v_s] \notin E;$
For a=1 to nants do
color the graph by means of a constructive method - (A)
let $s = (V_1, V_2, \ldots, V_q)$ *be the coloring obtained*
If $q < f^{\circ}$ *then* $s^{\circ} = (V_1, \ldots, V_q); f^{\circ} = q;$
$\Delta Mrs = \Delta Mrs + 1/q \; [v_r, v_s] \notin E; \{v_r, v_s\} \leq V_j, j=1,\ldots,q$
$Mrs = \rho \; Mrs + \Delta Mrs \;\; \forall[v_r, v_s] \notin E;$
Research and local Search (proposal) - (B)

Constructive method that is used on 'color the graph by means of a constructive method - (A)' part at ANTCOL algorithms are ANT_Random, ANT_LF, ANT_SL, ANT_DSATUR, ANT_RLF, that apply Random, Largest First (LF) [8], Smallest Last (SL) [9], DSATUR [10] Recursive Largest First (RLF) [11] to ACS. Costa and Hertz (1997)[8] compared execution time with coloring result that use several constructive method. And he shows that coloring result of ANT_RLF is best. Analyze result, when use ANT_Random, execution time was shortest, but coloring result is worst. He shows ANT_RLF's run-time is longer than other construction method when the graph size increase. Because node number that search for graph coloring increases and complexity of constructive method increases requiring much times to update secretion value τ_2(s[k-1], v, c) at each step. This paper proposed ANT_XRLF that apply Randomize to ANT_RLF to solve problem. And wish to prove superiority of performance.

3. New Approach

3.1 XRLF in ANTCOL (ANT_XRLF)

Usually, Randomize had been used often in number setting way to shorten simple Heuristic retrieval time. Because do not search whole item, number of item to search searching items that select randomly decreases and then retrieval time was shortened. Johnson (1991)[12] proposed XRLF(eXtended RLF) that apply Randomize to RLF that is used in general search because idea that can shorten retrieval time if use Randomize. XRLF that Johnson proposes randomly selects first item to search in general RLF. Construction method of graph coloring that propose in this paper applies Johnson's XRLF to ACS and solve graph coloring problem. Main part of ANT_XRLF's algorithm, is as following.

First, create some candidate list (Candidate List: CL) in each color set Vi = {V1 ...Vk} and choose one of them. Each candidate lists create through XRLF and in each construction phase, randomly select nodes that color does not paint as constant number (Candidate Size : CSize) in W that is set of node that can paint present of nodes that color is not assigned. And color node that Pit is maximum among them.

ANT_XRLF randomly select nodes to search as constant number from whole candidate set (whole node number to search for coloring), and candidate set of node size reduce. Procedures ANT_XRLF described in <Table 2>.

<Table 2> ANT_XRLF

$q = 0;$ // number of colors used
$W = V;$ /* uncolored vertices which can be included in the stable set under construction */
$k = 0;$ // number of colored vertices
while $k < |V|$ *do*
$k = k + 1; q = q + 1;$
$B = \emptyset;$ /* uncolored vertices which can no longer belong to the stable set Vq */
select first vertex v randomly in W
$V_q=\{v\};$ /* the neighborhood $N_w(v)$ of vertex v includes all vertices w∈W that are adjacent to v. */
while $W / (N_w(v) \cup \{v\}) \neq \emptyset$ *do{*
$k = k + 1; B = B \cup N_w(v); W = W / (N_w(v) \cup \{v\});$
$CL = \emptyset;$
/* coloring candidate node list in uncolored node*/
if $|W| \geq CSize$ {
for i = 1 to CSize
select randomly $v_0 \in W$
$CL = CL \cup \{v_0\}$ }
else $CL = CL \cup W$
Choose $v \in CL$ *Pit(k, v)*
with $\tau_1(s[k-1],v) = \tau_2(s[k-1],v,q)$ *and* $\eta_1 = deg_B(v);$
// $deg_W(v)$: *number of* $N_w(v)$
$V_q = V_q \cup \{v\};$
$W = B \cup N_w(v)$

3.2 XRLF and Research (ANT_XRLF_R)

Randomly select candidate node from whole candidate node set as constant number (Csize) in ANT_XRLF. So, coloring result is worse than ANT_XRLF and other construction method (ANT_Random, ANT_LF, ANT_RLF, etc) when node number of graph is small. Re-search whole colored node during shorten time by using randomize for improvement of coloring result that use ANT_XRLF. And search nodes that can assign the same color and do re-coloring. Propose such method that is ANT_XRLF_R. Define ANT_XRLF_R process, is as following.

Constructive method that is used on 'color the graph by means of a constructive method - (A)' part at ANTCOL algorithms are ANT_Random, ANT_LF, ANT_SL, ANT_DSATUR, ANT_RLF, that apply Random, Largest First (LF) [8], Smallest Last (SL) [9], DSATUR [10] Recursive Largest First (RLF) [11] to ACS. Costa and Hertz (1997)[8] compared execution time with coloring result that use several constructive method. And he shows that coloring result of ANT_RLF is best. Analyze result, when use ANT_Random, execution time was shortest, but coloring result is worst. He shows ANT_RLF's run-time is longer than other construction method when the graph sizes increase. First sort coloring sets. Selects the two (Vj, Vj-1) sets of smallest coloring nodes. Combine copy of second smallest coloring node set (V'j-1) and the smallest set (Vj), new coloring set call 's'. E (V'j) is set of inter connected line in coloring node set V'j. Define f(s) is sum of nodes E(V'j) when j = 1 to k and earch value of f(s) using local search. If f(s) is zero, then union of coloring nodes is valid. And store coloring node set V'j. If f(s) larger than 0, there is no decrease of coloring number. So selects new two sets of small coloring nodes and repeat by same step. Search colored nodes during the shortened time and re-color nodes that can assign the same color.

So we can expect improvement with coloring result. ANT_XRLF_R algorithm described in <Table 3>. And add to ANTCOL algorithm's part **'Research and local Search (proposal) - (B)'**

<Table 3> ANT_XRLF_R

Research (V, E, i, {V_1, ..., Vi})
do {
Sort V as $|V_1| \geq |V_2| \geq ... \geq |V_{i-1}| \geq |V_i|$;
k = i-1
for j =1 to k $V'j = Vj$;
s = {V'j, ..., V'k ∪ Vi};
Local Search(k, s)
if f(s) = 0 then
for j = 1 to k; $Vj = V'j$
i = k
} while f(s) = 0
Local Search(k, s)
NImprove = 0
// NImprove : number of improve node
while (f(s) > 0 and NImprove < NImpIter)
// NImpIter : number of iteration
make all possible attempts to switch v to different color - (1)
if (1) success then NoImprove = 0
else NoImprove = NoImprove +1

4. Experimental Result

In this paper, establish equally and experimented environment and variable to compare with coloring result and execution time of Costa and Hertz (1997) [8]. And evaluate performance by change the Candidate List (CL) size (Csize).

-Random Graph G (n, p) is graph that has n nodes, and edges by independent probability p. Search time measures by CPU second of computer. Experiment with variables of Pit, Pre in ACS are fixed by $\beta_{1,2} = 4$, p = 0.5, nants ∈ {100, 300}, CL = 3.

-Experiment with "Michael Trick's Graph Coloring Instances"

LEI: Leighton Graphs (Leighton, 1979)

MYC: Graphs based on Mycielski transformation (Michael Trick)

REG: Graphs based on register allocation for variables in real code(Gary Lewandowski)

SGB: Graph from Donald Knuth's Stanford GraphBase

LEI is consist of graphs of 25 150-node graph and 12 450-nodes that coloring result is 5-25 and average degree is 11 - 77.

MYC and REG are set of graphs that are well known optimization graph of each 5 and 14.

4.1 Performance by Construction

Fig.1, <Table 4> and <Table 5> show result of coloring number in Random Graphs and Optimization Graphs. When experiment in Random Graph, ANT_XRLF_R coloring result is best and ANT_XRLF result is second.

Optimization Graphs coloring result is same with Random Graph result that ANT_XRLF_R is better than ANT_XRLF. <Table 6> show result of run-time in optimization graphs. When experiment in optimization graphs, ANT_XRLF_R run-time result is best and ANT_XRLF result is second. If arrange results that use ANT_RLF, ANT_XRLF, ANT_XRLF_R, ANT_XRLF_R that propose addition adds re-search step increase than simple ANT_XRLF run-time about 40% but coloring result improve. ANT_XRLF_R improve run-time and coloring result.

<Table 4> Result of color number using Random Graphs

Gn, 0.5 (nants)	ANT_RLF	ANT_XRLF	ANT_XRLF_R
G100 (100)	15.4	16.6	**15.3**
G300 (100)	36.3	38.3	**35.8**
G500 (100)	56	52.6	**52.5**
G100 (300)	15.2	16.2	**15.1**
G300 (300)	35.7	38.2	**35.1**

<Table 5> Result of color number using Optimal Graphs

name	ANT_RLF	ANT_XRLF	ANT_XRLF_R
LEI	18.3	17.8	**16.9**
MYC	6.0	6.0	**6.0**
REG	37.9	37.8	**37.4**
SGB	31.1	32.0	**26**

<Table 6> Result of run time using Optimal Graphs

name	ANT_RLF	ANT_XRLF	ANT_XRLF_R
LEI	21.50	**10.52**	14.56
MYC	0.47	**0.24**	0.31
REG	16.65	**8.93**	12.85
SGB	1.05	**0.45**	0.62

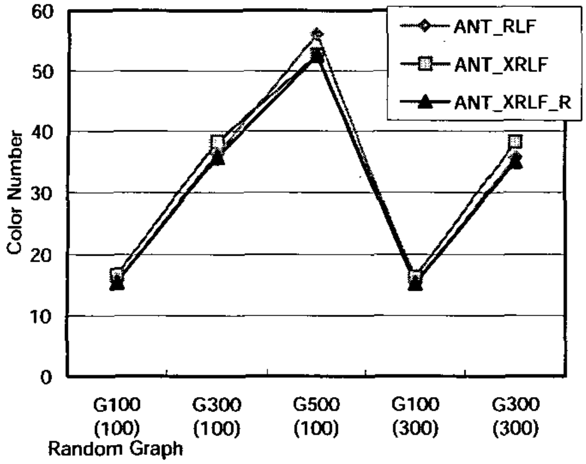

Fig. 1 Result of color number using Random Graphs

4.2 Candidate list

Usually, set candidate list size is fixed 3 and experimented. In this chapter evaluate performance by change the candidate list size (CSize). Random Graph G (n, p)'s experiment variables are fixed p = 0.5, n = {100, 300}, nants = 300, CSize = {2, 3, 4, 5, 6, 7, 8, 9}.

And show result that experiment in Fig.2 According to Fig.2, in case CSize = 5, coloring result is best.

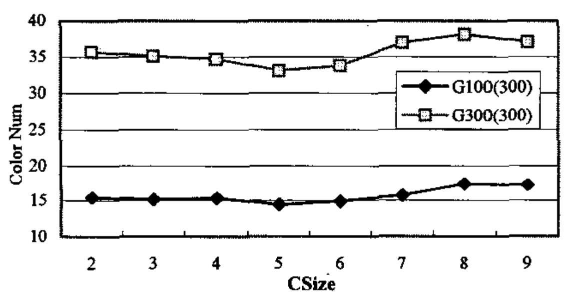

Fig. 2 Coloring result comparison by CSize

5. Conclusion

Introduce graph coloring problem and ANTCOL to solve graph coloring problem using ACS and ACS in this paper. Compare coloring result and execution time that use ANT_XRLF_R, ANT_XRLF, and ANT_RLF as a constructive method. And also compare coloring result by CSize change.

When used ANT_XRLF_R method, coloring result and excute time was improved. And in case CSize = 5, coloring result is best.

We will compare other graph coloring algorithms and coloring result and execution time. Also, we will study more about application possibility availability in ATP of other form.

6. Reference

[1] Colorni A, Dorigo M, Maniezzo V "Distributed optimization by ant colonies" Proceeding of the first European Conference on Artificial Life. Paris, MIT Press Bradford Books, Cambridge pp.134-142 (1991)

[2] Colorni A, Dorigo M, Maniezzo V "An investigation of some properties of an ant algorithm" Proceeding of the Second Conference on Parall Problem Solving from Nature. Brussels, North-Holland, Amsterdam pp. 509-520 (1992)

[3] Maniezzo V, Colorni A, Dorigo M "The ant system applied to the quadratic assignment problem" IRIDIA, Universite Libre de Bruxelles, Belgium (1994)

[4] Colorni A, Dorigo M, Maniezzo V "Ant System for job shop scheduling" Belgian J Opns Res. Stat and Comp Sci Vol 34. pp.39-53 (1994)

[5] Dorigo M, Maniezzo V, Colorni A "The ant system: optimization by a colony of cooperation agents", IEEE Transactions of Systems, Man and Cybernetics-Part B, vol. 26, no. 2, pp. 29-41 (1996)

[6] Ferland J A, Hertz A and Lavoie A "An object oriented methodology for solving assignment type problem with neighborhood search techniques" Opns Res Vol 44. pp.347-359 (1996)

[7] D. Costa and Hertz A. "Ant can colour graphs", J. Op. Res. Society, pp. 295-305 (1997)

[8] Welsh D. J. A and M. B. Powell "An Upper Bound for Chromatic Number of a Graph and its Application to Timetabling Problems" Comput. J. vol.10 pp.85-86 (1967)

[9] Matula D. W, G Marble and J .D. Issacson "Graph Coloring Algorithms" on Graph Theory and Computing, R. C. Read, Academic Press, New York, pp.104-122 (1972)

[10] Bralaz D "New methods to color vertices of a graph" Communications of the ACM Vol 22 pp. 251-256 (1979).

[11] Leighton F"A graph coloring algorithm for large scheduling problems" J. National Bureau of Standards Vol 84 pp.489-505 (1979)

[12] Johnson D. S, Aragon C. A, "Optimization by Simulated Annealing: An Experimental Evaluation-Part2 (Graph Coloring and Number partitioning)", Operations Research, vol. 31, pp.378-406 (1991)