2007 IEEE International Conference on Signal Processing and Communications (ICSPC 2007), 24-27 November 2007, Dubai, United Arab Emirates

Robust and Efficient Ant Colony Algorithm; Using New Local Updating Rule

Nima Taherinejad and Hossein Miar Naimi
Electronic Engineering Faculty-Babol University of technology
Dr.Shariati Str – Babol, Tele Fax: 0111- 32 39 214
h_miare@nit.ac.ir

Abstract
In this paper, two new robust ant colony algorithms with better results will be presented. The main approach to improve older algorithms is to use an intelligent local updating method. Here all agents haven't the same effects on the paths; local updating is done based on the situation and rout passed by agents. In order to evaluate and compare the results of new algorithm many standard problems of ***TSP Library*** and some random problems were tested. The experiments proved the better results of new algorithm and also its considerable better robustness.

Key words: Ant Colony Algorithm, Local Updating, Optimizing, Heuristic Algorithms

1. Introduction

Many optimization problems were found to be Non-Polynomial and their computational complexity, needed to find the best result is exponential or factorial function of the problem size. Using search algorithms is a common approach to tackle these type problems. Genetic Algorithm (GA)[2], Tabu Search[3], Ant Colony Algorithm[4,5] and … are the main search algorithms that can be mentioned.

Ant Colony Algorithm is a mathematical model of Ants behavior, finding the shortest path between nest and food. The search capability of ants –using no visual sign- is the most attractive aspect of their behavior. Passing the paths, ants distribute pheromone on the paths where analyzing pheromone amount make them able to find the shortest path[6, 7, 8]. In addition, they can find the shortest path again after the former shortest path is destroyed, i.e. because of an obstacle[9]. Ants' behavior especially about pheromone and their path finding capability were inspired by scientist to build an algorithm to solve optimization problems. The best and most successful one is done by Deneubourg and his assistants [10].

Studying search algorithm, the first problem to study should be Traveling Salesman Problem (TSP) which is used as a basic standard problem to evaluate many search algorithms especially Ant Colony Algorithm. The rest of the paper is as follows: in section 2 the Ant Colony Algorithm structure and the definition of TSP are introduced. In section 3 a very brief history of ant algorithm is introduced. Section 4 is dedicated to proposed algorithm and finally some experiments on standard TSP problems of TSP Library[1] and some random problems were done which their results and comparisons with other algorithms are reported in section 5.

2. Ant Colony Algorithm

As mentioned before, ant behavior -which resulted in Ant Algorithm- and TSP problem have many common aspects. Although the algorithm can be implemented on many different problems but ants natural behavior is resolving some kind of TSP problem and so before studying Ant Algorithm structure it's better to introduce Traveling Salesman Problem.

2.1. Traveling Salesman Problem (TSP):

Considering cities C_1, C_2, C_3, …, C_n with $d(C_i,C_j)$ as the distance between C_i and C_j , we have a complete graph of cities with connecting lines as distance between each pair of cities. If $d(C_i,C_j)=d(C_j,C_i)$ then the problem is symmetric and otherwise is asymmetric. Here the problem is finding the shortest Tour, where a Tour is a path which passes each city once and only once! How can we systematically find the shortest tour? Answering these questions equals to solve the TSP.

In different papers it's proved that TSP is a NP-Hard Problem [11] and we can simply find out, "A problem with n cities, have (n-1)! Tours". This demonstrates the explosive nature of the solution space of the problem, where deterministic algorithms for such problems weren't found. As a result search algorithms are focused to solve such problems and one of important search algorithms in this field is Ant Colony Algorithm, which is main aim of this paper as well.

2.2. The Ant Colony Algorithm; A Brief Introduction

As a top view, the algorithm can be described as below. A number of agents (ants) move through the path and leaves pheromone on their passed path and so affect other ants while selecting the next city of their tour path. In fact an ant will choose more probably the path which has more amount of pheromone. Figure 1 shows the algorithm structure.

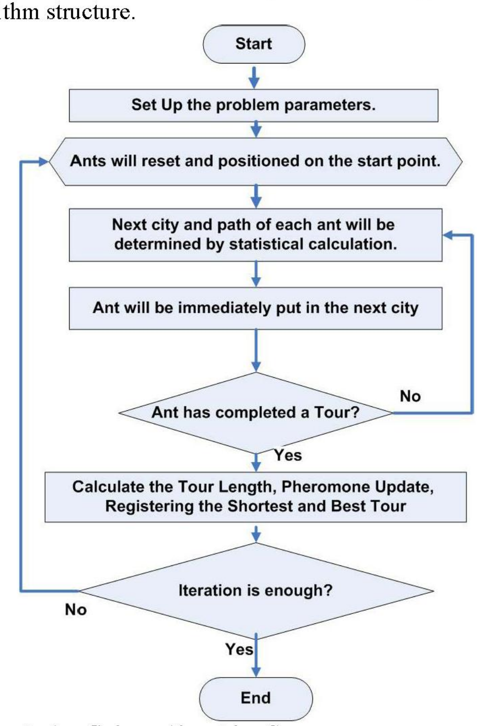

Figure 1. Ant Colony Algorithm Structure

3. A Brief History of Ant Algorithm

Different versions of Ant Algorithms differ in each section of above mentioned structure. The complete discussion about historical aspect of the algorithm is too long to be considered in this paper.

[12] is an introductory paper about ant colony where the Ant-Cycle algorithm is introduced and some primitive ideas, like pheromone update and probability of selecting a path, are

described. Then some investigation about parameter tuning was done and results were reported.

Paper [13] was published in IEEE and in where after presenting ants' behavior and the artificial algorithm (Ant-Cycle), two new algorithms were presented and studied, 1-Ant-Density, 2-Ant-Quantity

The paper [14] is about another derivation of Ant Algorithm, called Ant Colony System (ACS) which is an implementation of Ant-Q on TSP [15]. They also ran ACS on some bigger problems. For these runs they implemented a slightly modified version of ACS which incorporates a more advanced data structure known as *candidate list*, a data structure normally used when trying to solve big TSP problems [16].

In [17] elitist ants' idea is exchanged with another approach which is more adaptive for parallelization and multiprocessing. In this approach global best tour (which is used in elitist ants' algorithm) is located with local best tour of agents.

The paper [18] discuss about some Ant Algorithm implementation like Ant System (AS), Ant Colony System (ACS) and Approximated Non-deterministic Tree Search (ANTS). First and second ones were presented before and the last one which is based on partial solutions will be presented in this part.

4. New Algorithms

In this paper two new algorithms will be presented, where the algorithms has been changed to more heuristic ones and the transactions are done more genius. The main approach is enhancing the *local update* rule.

Reviewing the history of Ant Colony Optimization Algorithm, obviously it could be found out; where the models of ants' behavior changed to more heuristic one it was led to better results. Now the question, how heuristic is to add a constant amount of pheromone to each edge of graph when an agent (ant) has passed through it while completing its tour?

When a tour starts, all paths have the same amount of pheromone and so they've equal chance to be selected by ants. When an ant selects a path and passes through it, the pheromone amount of path will be increased obeying the *local update* rule (which is reverse proportional to the length of path). This process will make the edge more desirable for other ants that have this edge as a choice in their path. More amount of pheromone on an edge, more desirable to select.

But is it adequate to give all the agents the same possibility of affecting the edges' pheromone? Consider an ant in its primitive part of tour, while the ant arrives to a city and wants to choice its next path, the choice is vast because it has pass just some cities and just few paths are prohibited to select and it can freely choose the most desirable path (more pheromone, less length) as its next path.

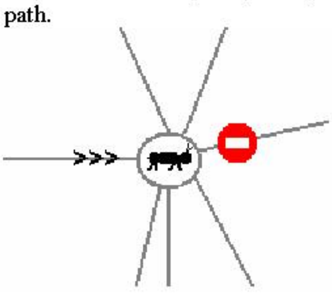

Figure2.a

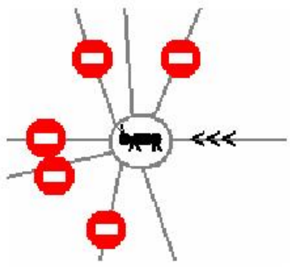

Figure2.b

Figure2. An agent in primitive parts of its tour (Figure2.a) is freer to select the more desirable path, but in final parts (Figure2.b) has less possibility of selecting more desirable path.

Now consider the same agent in its final parts of tour where it has passed most cities and now have few choices to do, and with huge amount of probability the selected path is not the best possible thereby. And so it seems wiser to give it less part in updating pheromone of its selected path.

In the other hand where the ants pass through the cities the probability of mistaking will increase. Consider an inadequate choice in primary parts of its tour (cause of probability of *selecting path rule* or some obligation of paths) which may lead in other inadequate selection in latter parts of the tour and this process will keep on and may increase errors and finally cause to bad result. And so a good start may lead in bad result, therefore it seems better to let agents have more effect on pheromone update where they've started the path, and less when they're going to finish it.

Based on upper discussions two new rules were designed for local updating process and experiments were done to compare new algorithms with former ones.

In the first algorithm called **Kcc-AntS** furthermore, local updating is done based on the following equation.

$$\tau(r,s) = \tau(r,s) + \frac{K \cdot cc \cdot \tau_0}{\frac{cc}{Cl^{\eta}}} \qquad (1)$$

Where "cc" stands for Current City number (i.e. the number of cities passed till now), Cl Stands for Current Length which is the current length of passed path for each ant and finally K & η are two tuning parameters which tune the effectiveness of number of passed cities (cc) against length of past paths (Cl).

In the second algorithm called **ELU** furthermore, *local update* rule for a problem with M cities (node) obeys Equation 2.

$$\tau(r,s) = \tau(r,s) + \tau_0 \cdot e^{-\frac{5cc}{M}} \qquad (2)$$

As it's obvious, the second term of mentioned Equation will exponentially decreases toward zero and when cc = M the term is almost zero ($e^{-5} \approx 0$). So the ants play fewer roles in local pheromone update when they are in their final part of tour.

As it was discussed before, the main idea was not to permit ants to equally affect pheromone update whether in first cities or last ones.

One of the logical choices was the exponential function which acts the same. But as discussed before when a tour starts the edges have the same pheromone and probability is more effective respect to edges' pheromone and after some steps the pheromone would demonstrate its effect, so it could be better to increase ants' effect in local pheromone update when the pheromone has made its effect on edges and after some step decrease ants affect. Also the length of passed paths could show how elite is an ant. Considering all these reasons, Equation 1 could be satisfying.

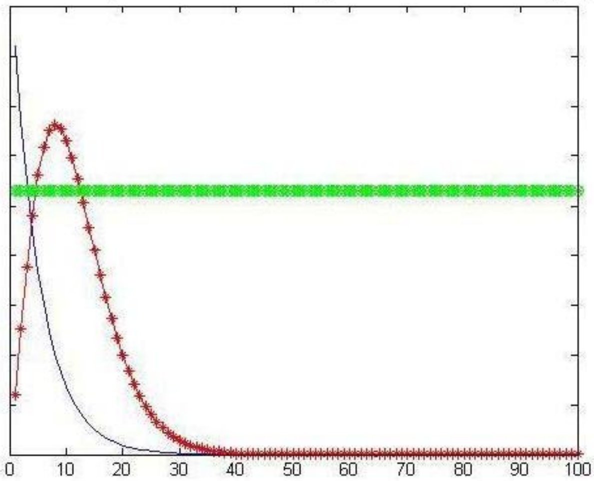

Figure 3. Virtual effect of Old Algorithms (Green-Wide), Kcc-AntS (Red-Star) and ELU-AntS (Blue-Narrow) on local pheromone update along their tour.

Figure 3 clarifies the different between three *local update* rules applied in former algorithms and latter ones (Kcc-AntS and ELU-AntS). In this virtual problem with 100 cities, ants have a constant effect on *local update* where they are, but agents in ELU-AntS have less chance (almost zero) when they passed some parts of their tour and finally in Kcc-AntS the agents have some starting chance for *local update* which increases for a while and then will decreases toward zero.

5. Experiments

Testing new algorithm, experiments were done on 17 *standard TSP problems* caught from ***TSP Library*** and the results were compared in order to determine elite algorithm. In addition the τ_0 parameter was varied in order to compare algorithms' result while the primary parameter set up changes and analyzing their robustness against parameter tuning.

The experiments were done in similar situation except in *local update* rule which was different for each algorithm. The algorithms were structured as:

Primary setup: Random distribution.
Selecting next city: State Transition Rule [15].
Global pheromone update: it was done using Equation used in [14].
Local Pheromone Update: it was done using

- *Old Algorithm:* [14].
- *Kcc-AntS:* Equation 1.
- *ELU-AntS:* Equation 2.

Parameters Setup:

PDS[1] = 0.2	$\rho = 0.9$
$\lambda = 0.1$	$\alpha = 0.1$
$q_0 = 0.9$	$\beta = 2$
$\eta = 9$	$K = 0.1$

It should be mentioned that α, β, ρ & q_0 were selected as was advised in [12], [13] and other parameters were selected optionally.

For each value of τ_0, problem was run 15 times and the average was assigned as the test result for that value of τ_0. Also the algorithms were iterated on each problem for 18 different value of τ_0.

The results demonstrate that Old Algorithm has worst results and large amount of variation respect to new algorithms and also the offered value for τ_0 is not optimum in all situations and for whole problems. As an example Figure 4 should be attended.

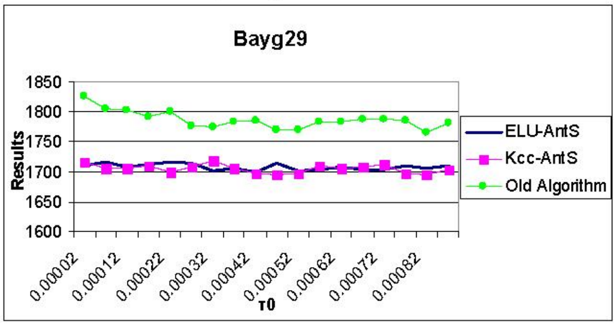

Figure 4. New Algorithms has better results & less variation.

Although there were some problems with better results given by Old Algorithm (Only 2 problems) but as shown in Figure 5, the variance of Old Algorithm was more than new ones (it was more than two times greater than ELU-AntS Variance).

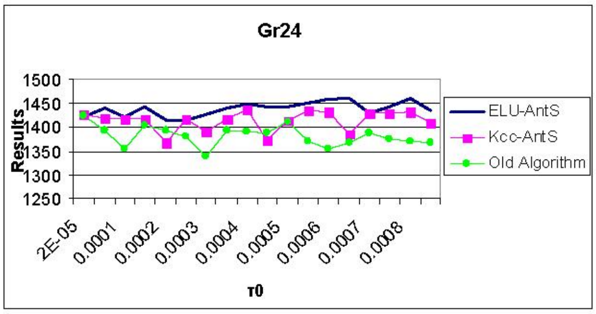

Figure 5. Even in those few problems which Old Algorithm had better results, it had worse variance.

By the way when the problem dimension gets larger (number of cities increases) new algorithms give better results.

Table 1 includes the average of 18 algorithms' results for different values of τ_0, assigned as Average, Minimum of these 18 averages (each one stand for 15 iterations with one value of τ_0) and finally Variance/Average multiplied by 100.

Table 1. Average, Minimum & Variation Coefficient of results caught in experiments different algorithms. New ones have obviously better results.

Problem	Average		
	Old	Kcc	ELU
Gr24	**1381.2**	1414.0	1438.3
Fri26	992.3	**940.5**	941.8
Bayg29	1786.6	**1704.8**	1707.9
Bays29	2296.9	**2137.7**	2224.2
Dantzig42	**813.3**	843.4	843.9
Swiss42	1478.4	**1446.2**	1446.8
Gr48	5952.0	**5718.5**	5759.5
HK48	5990.7	**5722.1**	5772.7
Brazil58	24838.5	23735.2	**23540.1**
Pr76	75408.0	**72519.9**	73117.0
Eil101	255.6	**228.4**	229.5
Bier127	55469.6	**48278.3**	49126.1
KroB150	18795.2	17278.2	**17046.0**
KroB200	20911.7	19626.5	**19527.9**
Tsp225	2056.4	**1882.4**	1892.3
A280	56048.7	**47931.3**	48880.5
Lin318	24927.4	**23059.0**	23085.7

Table 1. Continued

Problem	Minimum		
	Old	Kcc	ELU
Gr24	**1338**	1366.8	1412.9
Fri26	974.8	937.7	**937.5**
Bayg29	1766.7	**1694.9**	1698.9
Bays29	2235.4	**2134**	2187.6
Dantzig42	**799.8**	820.4	832
Swiss42	1451.5	1436.2	**1426**
Gr48	5869.8	5665.4	**5613.2**
HK48	5898.8	**5653.1**	5662
Brazil58	24123.3	23314.4	**23110.8**
Pr76	73190.8	**71012.4**	72060.6
Eil101	247.6	**221.6**	225.3
Bier127	54150.2	**47286.6**	47778.2
KroB150	18540.3	17069.3	**16888.4**
KroB200	20633.7	19383.1	**19280.9**
Tsp225	2019.3	**1841.8**	1849.5
A280	54601.2	**46696.1**	47297.6
Lin318	24566.5	**22885.7**	22890.0

[1] Pheromone Density Setup

Table 1. Continued

Problem	100*Variance/Average		
	Old	Old	Old
Gr24	34.44	34.44	34.44
Fri26	9.84	9.84	9.84
Bayg29	11.86	11.86	11.86
Bays29	28.44	28.44	28.44
Dantzig42	6.21	6.21	6.21
Swiss42	19.02	19.02	19.02
Gr48	**21.28**	**21.28**	**21.28**
HK48	59.30	59.30	59.30
Brazil58	513.95	513.95	513.95
Pr76	1898.6	1898.6	1898.6
Eil101	3.68	3.68	3.68
Bier127	1020.16	1020.16	1020.16
KroB150	114.48	114.48	114.48
KroB200	105.09	105.09	105.09
Tsp225	37.50	37.50	37.50
A280	858.74	858.74	858.74
Lin318	196.51	196.51	196.51

It can be find out that new algorithms have better results and variation respect to Old Algorithms.

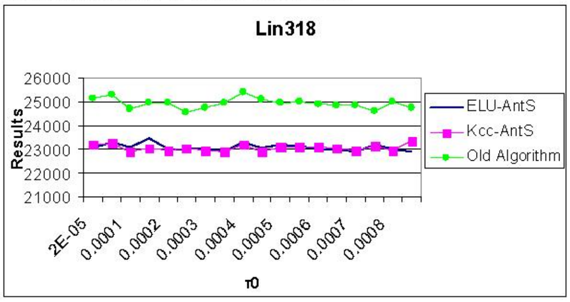

Figure 6. Two new algorithms have very similar behavior.

Comparing two new algorithms it can be understood that Kcc-AntS have better Averages respect to ELU-AntS but in Minimums competition ELU-AntS could find better result although not better averages and still Kcc-AntS is more elite but in variance coefficient they become equal.Generally as it's shown in Figure 6, two new algorithms have very similar behavior but Kcc-AntS is slightly better. Selecting the better algorithm to use depends on situations, although the algorithms are robust against τ_0 changes and it could be ignored as parameter for tuning (i.e. using new algorithms there is no critical need to tune τ_0) but as it's clear in Equation 1, in Kcc-AntS there are two parameters to tune (K & η) but in ELU-AntS (Equation 2) there is NO parameter to tune.

It's interesting to be mentioned that we could find shorter tours and improve results in six problems, respect to what exists in ***TSP Library***. The problems which better results were found are: Brazil58, Pr76, Bier127, KroB150, KroB200, TSP225 And in one problem (Fri26) the same result with ***TSP Library*** was caught.

As a final note, it should be mentioned that Kcc-AntS was run for K=1 and K= 1/cc but generally they couldn't be better than 0.1cc-AntS and ELU-AntS.

6. Conclusion & Further Researches

Conclusion:

Two new algorithms were presented in this paper and were compared with best former algorithms in this genre and could find better results.

It was shown that offered τ_0 in former percents was not adequate but in new algorithms the error percent against τ_0 variation is usually less than 2% which should be ignored and there is no need to tune this parameter in these algorithms.

Comparing two new algorithms, both have very similar behavior but Kcc-AntS (with K=0.1) is slightly better (if ELU-AntS was not better had less than 2% error) but it had two parameters to tune, while ELU-AntS had NO tuning parameter and so while we face with combinational algorithm which tune themselves [19] or we've tuning possibility the Kcc-AntS is offered and otherwise ELU-AntS.

Further researches:

Further researches should be done on:

- K & η tuning in Kcc-AntS and tuning other parameters of algorithms.
- Studying Algorithm behavior against problem specifications and find an exact relation to explain it.
- Decreasing tuning parameters and make algorithm robust against parameter tunes. (As it was done in this paper for τ_0)

References:

[1]http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/tsplib.html

[2] Nagata, Y. & S. Kobayashi. (1997). Edge Assembly Crossover: A High-power Genetic Algorithm for the Traveling Salesman Problem. Proceedings of Seventh International Conference on Genetic Algorithms (ICGA'97), East Lansing, Michigan, pp. 450-457.

[3] Glover, F. and Laguna, M. (1997). "Tabu Search" Boston: Kluwer Academic Publishers

[4] C. Blum, "Ant Colony Optimization: Introduction and recent advances", Fifth International conference on Hybrid Intelligent Systems (HIS'05), Rio de Janeiro - Brazil, November 6-9, **2005.**

[5] An Introduction to Ant Algorithms and to Ant Colony Optimization, ANTS'2000 - From Ant Colonies to Artificial Ants: Second International Workshop on Ant Algorithms, Brussels, Belgium, September 7, **2000**.

[6] R.Beckers, J L Deneubourg and S Goss , "Trails and U-turns in the selection of the shortestpath by the ant Lasius Niger", Journal of Theoretical Biology, vol 159 , pp 397-415

[7] S. Goss, S. Aron, J.L. Deneubourg, and J.M. Pasteels, "Self-organized shortcuts in the argentine ant," *Naturwissenschaften*, vol. 76, pp. 579–581, 1989.

[8] B. Holldobler and E.O Wilson , The Ants , Springer – Verlag , Berlin , 1990

[9] Beckers, Deneubourg and Goss, 1992; Goss, Aron, Deneubourg and Pasteels, 1989

[10] Deneubourg , Aron , Goss , & Pasteels , 1990 , Goss et al 1989

[11] E.L.Lawler , J.K.Lenstra , A.H.G.Rinnooy Kan and D.B.Shmoys , editors. The *Traveling Salesman Problem*. Wiley ,1985

[12] Alberto Colorni , Marco Dorigo , Vittorio Maniezzo , al . "An investigation of some properties of Ant Algorithm",Brussels , Belgium, PPSN 1992 , Elsevier Publishing, 509-520

[13] Alberto Colorni , Marco Dorigo , Vittorio Maniezzo , al,"The ant system optimization by a colony of cooperating agent" IEEE 1996

[14] Marco Dorigo , Luca Maria Gambardella , al "Ant Colonies for Travelling Salesman Problem" , TR/IRIDIA/1993-6 , Universite` Libre de Bruxelles , Belgium

[15] Marco Dorigo, Luca Maria Gambardella, "A STUDY OF SOME PROPERTIES OF ANT-Q", TR/IRIDIA/1996-4
*Université Libre de Bruxelle,*Belgium, published in *PPSN IV–Fourth International Conference*

[16] Reinelt, G., 1994, *The traveling salesman: Computational solutions for TSP applications* (Springer-Verlag, Berlin).
Johnson, D.S. and McGeoch, L.A., in press. The Travelling Salesman Problem: A Case Study in Local
Optimization, in: *Local Search in Combinatorial Optimization*, E.H.L. Aarts and J.K. Lenstra (eds.) (Wiley, New York).

[17] Simon Kaegi, Tony White, "Using Local Information To Guide Ant Based Search",School of Computer Science, Carleton University,Ottawa, Ontario, Canada, **2003**

[18] Vittorio Maniezzo , Luca Maria Gambardella , FabioDe Luigi ,al , "Ant Colony Optimization" , Aco**2004**

[19] Meibodi.M, Noferesty.S, "A combinational algorithm (Ant colony + Learning Automata) to resolve Steiner Static Tree Problem", ICEE **2006**