Ant Algorithms Unification and Improvement

Hossein Hajimirsadeghi
Control and Intelligent Processing Center of Excellence
School of ECE, University of Tehran, Tehran, Iran
h.hajimirsadeghi@ece.ut.ac.ir

Abstract: *Ant Colony Optimization (ACO) a nature-inspired metaheuristic algorithm has been successfully applied in the traveling salesman problem (TSP) and a variety of combinatorial problems. ACO algorithms have been modified in recent years to improve the performance of the first algorithm, posed by Dorigo. In this paper we compare different ACO algorithms and combine them in order to collect their advantages in an extended ACO algorithm.*

Keywords: Ant Colony Optimization (ACO), Travelling Salesman Problem (TSP).

1. Introduction

Ant Algorithms are a class of population-based meta-heuristic algorithms for solving Combinatorial Optimization. Ant Colony Optimization (ACO) is biologically inspired from the foraging behaviour of real ants. ACO is an iterative process in which repeatedly, probabilistic candidate solutions are constructed by heuristic knowledge of the problem and pheromone trails as communication mediums. The main points of ACO are distributed computation, positive feedback and greedy construction heuristics. After the first ACO algorithm proposed by Dorigo at el. (1992) [1], different types of ACO have been developed, most pursuing new ways of exploration or exploitation. Moreover, the combination of ACO and local search algorithms has led to successful results and obtained better performance on variety of problems. To date, ACO has been applied in many combinatorial problems, including Travelling Salesman Problem (TSP) [1, 2, 3] quadratic assignment [4], vehicle routing [5], graph coloring [6], routing for telecommunication networks [7], sequential ordering [8], scheduling [9], data mining [10], and so on.

The Travelling Salesman Problem (TSP) is an NP-hard combinatorial problem which has been the target of a great deal of research. It's an easily understood hard discrete problem which can be a good representation of many other NP complete combinatorial problems. Generally, TSP is the problem of finding shortest tour, starting from a city, visiting all the cities and finally going back to the first city. Indeed, TSP is the problem of finding the shortest Hamiltonian graph of a set of vertices. TSP has received much attraction of mathematicians and Computer Scientists because it represents the class of Combinatorial Optimization Problems and is formulated in so many other applications. ACO algorithms successfully have been applied to TSP, although the first proposed ACO wasn't competitive with the state of the art algorithms for TSP. In this paper, we compare different well-known ACO algorithms, experimenting on symmetric TSP.

Section 2 provides a quick review of Travelling Salesman Problem. In section 3, Ant system as the first member of the class of Ant algorithms is introduced, while ACO extensions and improvements are described in section 4. Next in section 5, our experiment is explained and the results are summarized. Finally, conclusions are drawn in section 6.

2. Travelling Salesman Problem

TSP is to find the cheapest round-tour, starting from a city, passing through all the cities and then returning to the starting point (Fig. 1). What is important about TSP is that it's a representative of a large and versatile class of problems known as

Combinatorial Optimization Problems, which are declared to be NP-complete. Indeed, TSP is easy and intuitive to understand and formulation, so if one can find an efficient algorithm for solving Travelling Salesman Problem it will be easily generalized to other Combinatorial Problems. A common formulation of TSP is a graph with points as cities and weighted edges as roads with their costs. The objective is to find the least costly Hamilton Cycle in the graph. There are two famous kinds of TSPs: symmetric and asymmetric TSP. In symmetric the difference between two cities is the same in both direction, while in asymmetric it might be different.

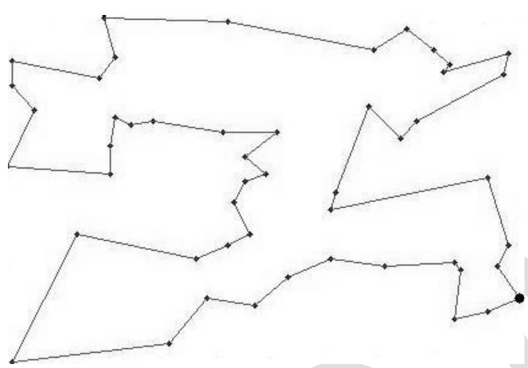

Fig. 1. TSP with 50 cities.

A common approach to solve TSPs is by using heuristics. These algorithms construct feasible solutions for the problem, satisfying the upper bounds in the problem, but neglecting how their solutions are far from the optimum. Heuristic methods for TSP usually fall into 3 categories. 1) tour construction methods, 2) improvement methods, and 3) composite methods [22]. On the other hand, a metaheuristic is a general framework for heuristics, combining user-given black box procedures in an efficient way for solving hard problems. Metaheuristics like Genetic Algorithm [23], Simulated Annealing [24], Neural Networks [25], Tabu search [26], and Ant Colony Optimization have been widely used to solve combinatorial problems, especially TSP.

There are a large variety of problems that can be formulated as TSP and so it is used in different fields of study. Vehicle routing (Christofides, 1985), job scheduling (Gilmore and Gomory, 1964), analysis of the structure of crystals (Bland and Shallcross, 1987), the overhauling of gas turbine engines (Pante, Lowe and Chandrasekaran, 1987), material handling in a warehouse (Ratliff and Rosenthal, 1981), cutting stock problems, (Garfinkel, 1977), the clustering of data arrays, (Lenstra and Rinooy Kan, 1975), are examples in which TSP formulation has been successfully applied.

In Electrical Engineering there are two major exploitation of TSP formulation. The first common application is in Telecommunications Network Design. One example is routing of sonet rings, which provide communications links through a set of sites organized in a ring. These rings are recommended to raise the safety of the network in a case of link failure because traffic can be rerouted in the reverse direction. The second application is VLSI chip board manufacturing. In this process the holes are the cities to be drilled in a board and the cost of travel is the time to move the drill head from one hole to the next, while the objective is to reduce the cost of the whole procedure.

3. Ant System

The first ACO algorithm called Ant System applied to Travelling Salesman Problem (TSP) by Dorigo [1]. AS makes up the main framework of other ACO algorithms and is considered as a prototype. In TSP each of m artificial ants generates a complete tour by a probabilistic rule (1), which is the probability that ant k in city i select visits city j.

$$p_{i,j}^k = \begin{cases} \frac{[\tau_{i,j}]^\alpha.[\eta_{i,j}]^\beta}{\sum_{l\in N_i^k}[\tau_{i,l}]^\alpha.[\eta_{i,l}]^\beta}, & \forall j \in N_i^k \\ 0\ , & otherwise \end{cases} \qquad (1)$$

Where τ is pheromone, $\eta_{i,j}$ is heuristic function and is equal to $\frac{1}{d_{i,j}}$ the inverse of the difference between city i and j, N_i^k is the set of cities that haven't been visited by ant k, α and β are parameters which shows the relative importance of pheromone versus heuristic or exploitation versus exploration.

Equation (1) shows that ants prefer paths with shorter length and higher amount of pheromone, so they independently generate tours by pre-knowledge of the problem and cooperative informative communication.

Once all the ants complete their tours the pheromone trails updates, using Equation (2) and Equation (3).

$$\tau_{i,j} = (1-\rho).\,\tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{i,j}^k \qquad (2)$$

$$\Delta\tau_{i,j}^{k} = \begin{cases} \frac{Q}{L_k}, & (i,j) \in tour\ done\ by\ ant\ k \\ 0\ , & otherwise \end{cases} \quad (3)$$

Where ρ is evaporation rate, L_k is the length of tour taken by ant k, Q is a constant, and m is the number of ants.

4. ACO Extensions

After Ant System, Researchers started to improve the performance of ACO. A first improvement of ACO was elitist strategy ($\mathrm{AS_{elite}}$) [2], which was simply considered more emphasis on the global-best tour. Another improvement was $\mathrm{AS_{rank}}$ as an offspring of $\mathrm{AS_{elite}}$, proposed by Bullnheimer, Hartl and Strauss [11]. It sorts the ants and then the trails are updated by only the first $\omega - 1$ ants according to Equation (4).

$$\tau_{i,j} = (1-\rho).\,\tau_{i,j} + \sum_{k=1}^{\omega-1}(\omega-1).\,\Delta\tau_{i,j}^{k} + \omega.\,\Delta\tau_{i,j}^{gb} \quad (4)$$

Where $\Delta\tau_{i,j}^{k} = \frac{1}{L_k}$ and $\Delta\tau_{i,j}^{gb} = \frac{1}{L_{gb}}$.

Stüzle and Hoos introduced MAX-MIN Ant System (MMAS) [12]. In MMAS trails are limited to an interval $[\tau_{min}\,,\tau_{max}]$, so it help ants not to converge to local optimum. Further, in MMAS, only the best ant (iteration-best or global-best) is allowed to deposit pheromone. Sometimes, for more exploration an additional mechanism called Pheromone Trail Smoothing is applied to MMAS.

Gambardella and Dorigo in 1996 proposed Ant Colony System (ACS) [13], which was a simplified version of Ant-Q. Ant-Q is a link between reinforcement learning and Ant Colony Optimization. However, ACS simply and more efficiently describes the same behaviour as Ant-Q. Two strategies are used in ACS to increase the previous information exploitation. At first, trails are updated by the best ant, like MMAS, and secondly, ants select the next city, using a pseudo-random proportional rule [14]. The rule states that With probability q_0 the city j is selected, where $j = arg\,max_{j\in N_i^k}\{[\tau_{i,j}]^{\alpha}.[\eta_{i,j}]^{\beta}\}$, while with the probability $1 - q_0$ a city is chosen, using Equation (1). Furthermore, there is a distinct difference between ACS and other ACO algorithms and that is trails are updated, while the solutions are built. It's similar to ant-quantity and ant-density approaches that update pheromone trails synchronize to making tours. However, in ACS ants eat portion of the trails as they walk on the path. So the probability that the same solutions are constructed in an iteration decreases.

AS Local Best Tour (AS-LBT) [15], is another improved kind of AS, in which only local information is used to reinforce trails. It means that each ant updates its trail by the best tour it has found to date. This approach shows more diversity than AS.

Some other improvements in the field of ACO are the Multiple Ant Colonies Algorithms [16], which exploits interactions between colonies, Population-based ACO (P-ACO) [17], which makes up a special population of good solutions, and Omicron ACO (OA) [18], that is inspired by MMAS and elitist strategy.

In addition a number of hybrid algorithms have been developed that use good features of ACO. For example the combination of Genetic Algorithm (GA), and ACO, called Genetic Ant Colony Optimization (GACO) have been used to solve different combinatorial problems [19, 20].

Moreover, ACO algorithms often exploit Local Search to improve their performance. 2-opt and 3-opt [21], algorithms are commonly add to ACO algorithm for a faster and more accurate convergence [14].

5. Experiment and Results

To evaluate the performance of the algorithms discussed in the previous section, we tested the extended algorithms on eil51.tsp, where the optimal tour length is 426. The population size of about 50 ants was used that was suggested in [2]. According to [11], We used parameter settings, α = 1, β = 5, ρ = .5. Furthermore, Q was set to the best tour length, found up to that iteration [15], and ω (also the number of elitist ants) was set to 7, approximately equivalent to $\frac{1}{6}$ of ants number [15]. Experimentally, we estimated eat rate = .9, and the trails with MMAS mechanism were set to interval, [0.02 , 2], with an initial value of 2.

10 trials were conducted, and all the tests were carried out for 1000 iterations. The results are presented in Table 1. Note that some of the proposed algorithms can lead to optimal solutions, increasing the number of iterations to about 2000. However, for the case of comparing the results, less number of iterations was designated to construct Table 1.

Table 1. Experiment results

Method	best	Avg.	Std. Dev.	Std. Dev. %	Best err. %	Avg. err. %
AS	455.91	462.24	3.36	0.73	7.02	8.50
ASLBT	443.99	458.36	5.30	1.16	4.22	7.60
ACS	437.29	457.72	12.37	2.70	2.65	7.45
ASrank	443.07	453.09	5.81	1.28	4.01	6.36
ACSMM	430.45	436.07	4.25	0.97	1.04	2.36
ASeliteMM	430.35	435.86	8.16	1.87	1.02	2.31
ASeliteLBTMM	429.53	435.39	4.41	1.01	0.83	2.20
ACSLBTMM	428.98	434.67	3.84	0.88	0.70	2.03
ASrankLBTMM	428.98	433.77	4.22	0.97	0.70	1.82
ACSrankMM	428.87	433.5	5.08	1.17	0.67	1.76
ACSrankLBTMM	428.98	433.36	4.15	0.96	0.70	1.73
ASrankMM	429.98	433.16	2.53	0.58	0.94	1.68

Shortly, ACSMM is a mixture of ACS and MMAS, while in ACSrankMM, bounded pheromone trails are updated, using Equation (4). Likewise, in ASeliteLBTMM, a number of elitist ants update trails with the best tours found to date, considering the limits on the amount of pheromone. Average of the best tours found per iteration for the proposed methods over the ten experiments is depicted in Fig. 2. The results show the performance of evaluated methods. In fact, limited boundaries for pheromone trails seem to be necessary to have efficient results. Moreover, good performance of incorporating LBT and rank-based approaches into ACO algorithms can be concluded from the table 1.

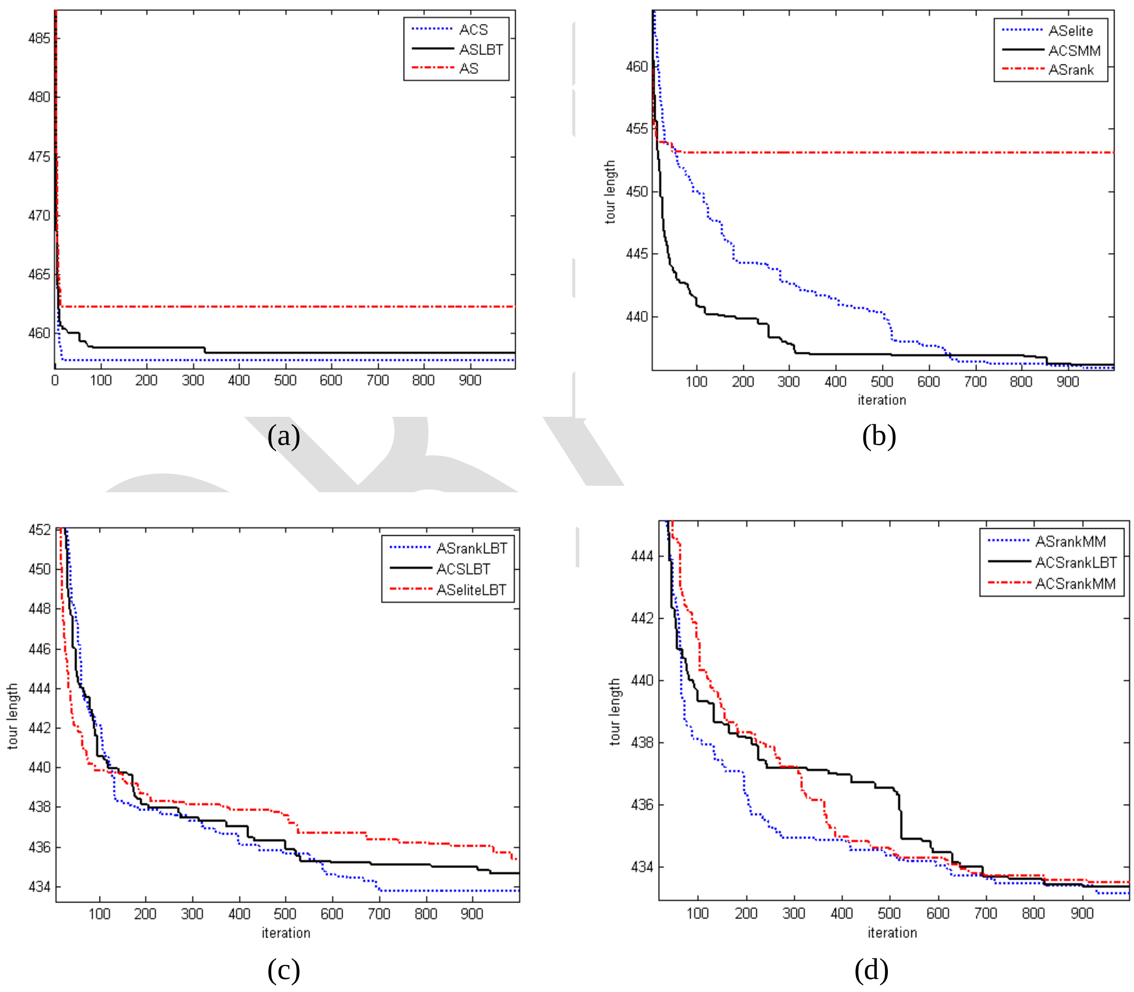

Fig. 2. Comparison between the proposed methods

6. Conclusion

This paper has evaluated the performance of different ACO algorithms and their combinations in solving Traveling Salesman Problem. In fact, special features of different approaches have been incorporated to construct more effective algorithms. Consequently, it has been shown that the idea of increasing exploration, using LBT method and also, confiding the pheromone trails into upper and lower bounds is a wise step to increase the efficiency of ACO algorithms.

Further work can be done in the direction of exploiting local search methods in favour of investigating the performance of each of the ACO algorithms accompanied by local search mediums. In addition, a more advanced data structure known as candidate list can be embedded into ACO algorithms. This structure provides a list of preferred cities to be visited in each step [27], and so the number of feasible solutions decreases.

References

[1] M. Dorigo. "Optimization, Learning and Natural Algorithms (in Italian)". PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992. pp.140.

[2] M. Dorigo, V. Maniezzo, and A. Colorni. "The Ant System: Optimization by a colony of cooperating agents." *IEEE Transactions on Systems, Man, andCybernetics* – Part B, 26(1):29–41, 1996.

[3] L. M. Gambardella and M. Dorigo. "Ant-Q: A reinforcement learning approach to the traveling salesman problem." *In A. Prieditis and S. Russell, editors, Proceedings of the Twelfth International Conference on Machine Learning (ML-95)*, pages 252–260. Morgan Kaufmann Publishers, Palo Alto, CA, 1995.

[4] V. Maniezzo and A. Colorni. "The Ant System applied to the quadratic assignment problem." *IEEE Transactions on Data and Knowledge Engineering*, 11(5):769–778, 1999.

[5] B. Bullnheimer, R. F. Hartl, and C. Strauss. "Applying the Ant System to the vehicle routing problem." *In S. Voß S. Martello, I. H. Osman, and C. Roucairol, editors, Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pages 285–296. Kluwer Academic Publishers, Dordrecht, 1999.

[6] D. Costa and A. Hertz. "Ants can colour graphs." *Journal of the Operational Research Society*, 48:295–305, 1997.

[7] G. Di Caro and M. Dorigo. "Extending AntNet for best-effort Quality-of-Service routing." Unpublished presentation at ANTS'98 - From Ant Colonies to Artificial Ants: *First International Workshop on Ant Colony Optimization.* http://iridia.ulb.ac.be/ants98/ants98.html, October 15-16 1998.

[8] L. M. Gambardella and M. Dorigo. HAS-SOP: "An hybrid Ant System for the sequential ordering problem." Technical Report IDSIA-11-97, IDSIA, Lugano, Switzerland, 1997.

[9] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian. "Ant System for jobshop scheduling." *JORBEL - Belgian Journal of Operations Research, Statistics and Computer Science*, 34(1):39–53, 1994.

[10] Parpinelli R S, Lopes H S and Freitas A A "Data mining with an ant colony optimization algorithm, "*IEEE Transactions on Evolutionary Computing*, vol.6, pp. 321-33, 2002

[11] B. Bullnheimer, R. F. Hartl, and C. Strauss. "A new rank-based version of the Ant System: A computational study." *Central European Journal for Operations Research and Economics*, 7(1):25–38, 1999.

[12] T. Stüzle and H. H. Hoos. "The MAX–MIN Ant System and local search for the traveling salesman problem." *In T. Bäck, Z. Michalewicz, and X. Yao, editors, Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, pages 309–314. IEEE Press, Piscataway, NJ, 1997.

[13] L. M. Gambardella and M. Dorigo. "Solving symmetric and asymmetric TSPs by ant colonies." *In Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, pages 622–627. IEEE Press, Piscataway, NJ, 1996.

[14] M. Dorigo and L. M. Gambardella. "Ant Colony System: A cooperative learning approach to the traveling salesman problem." *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.

[15] Tony White, Simon Kaegi, Terri Oda. "Revisiting Elitism in Ant Colony Optimization."

[16]Kawamura, H., Yamamoto, M., Suzuki, K. and Ohuchi, A. “Multiple Ant Colonies Algorithm Based on Colony Level Interactions”, *IEICE Transactions*, Vol. E83-A, No. 2, pp. 371–379 (2000).
[17]Michael Guntsch and Martin Middendorf. “A Population Based Approach for ACO.” *In Stefano Cagnoni, Jens Gottlieb, Emma Hart, Martin Middendorf, and Gunther Raidl, editors, Applications of Evolutionary Computing, Proceedings of EvoWorkshops2002*: EvoCOP, EvoIASP, EvoSTim, volume 2279, pages 71–80, Kinsale, Ireland, 3-4 2002. Springer-Verlag.
[18]Osvaldo G´omez and Benjam´ın Bar´an. “Reasons of ACO’s Success in TSP.” In M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. St¨utzle, editors, Proceedings of ANTS 2004 – *Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence*, volume 3172 of LNCS, Brussels, September 2004. Springer-Verlag.
[19]Zne-Jung Lee. “A hybrid algorithm applied to travelling salesman problem.” *Networking, Sensing and Control, 2004 IEEE International Conference* on Volume 1, Issue , 21-23 March 2004 Page(s): 237 - 242 Vol.1
[20]Tiao-ping Fu, Yu-shu Liu, Jian-hua Chen, "Improved Genetic and Ant Colony Optimization Algorithm for Regional Air Defense WTA Problem," icicic, pp. 226-229, *First International Conference on Innovative Computing, Information and Control* - Volume I (ICICIC'06), 2006.
[21]J.L. Bentley. “Fast algorithms for geometric traveling salesman problems.” *ORSA Journal on Computing*, 4(4):387–411, 1992.
[22]Coy, S., B.L. Golden, G. Runger, and E. Wasil, “See the Forest Before the Trees: Fine Tuned Learning and its Application to the Traveling Salesman Rroblem”, *IEEE Transactions on Systems, Man and Cybernetics* - Part A: Systems and Humans, Vol. 28, No 4, 1998, pp. 454-464.
[23]J.V. Potvin. "Genetic Algorithms for the Traveling Salesman Problem", Annals of Operations Research 63, 339-370, 1996.
[24]E.H.L. Aarts, J.H.M. Korst and P.J.M. Laarhoven. "A Quantitative Analysis of the Simulated Annealing Algorithm: A Case Study for the Traveling Salesman Problem", J. Stats. Phys. 50, 189-206, 1988.
[25]J.V. Potvin. "The Traveling Salesman Problem: A Neural Network Perspective", *INFORMS Journal on Computing* 5 328-348, 1993.
[26]C.N. Fiechter. "A Parallel Tabu Search Algorithm for Large Scale Traveling Salesman Problems" Working Paper 90/1 Department of Mathematics, Ecole Polytechnique Federale de Lausanne, Switzerland, 1990.
[27]Reinelt, G., “The traveling salesman: Computational solutions for TSP applications” (Springer-Verlag, Berlin), 1994.