Design and Implementation Issues in Ant Colony Optimization

Mahesh Mulani[1] and Vinod L. Desai[2]

[1]Research Scholar, Department of Computer Science, RK University, Rajkot, India.

Assistant Professor, Department of Computer Science, KSKV Kachchh university, Bhuj, Gujarat, India.

[2]Assistant Professor, Department of Computer Science, Gujarat Vidhyapith, Ahmedabad, India.

Abstract

Swarm Intelligence is a branch of Artificial Intelligence, widely used in solving complex problems such as combinatorial problems and NP-Hard problems. Ant Colony Optimization, Particle Swarm Optimization, Artificial Bee Colony Optimization Algorithm and African Buffalo Optimization are some of the concepts of swarm intelligence which are widely used in optimization problems. The paper focuses on Ant Colony Optimization and its applications in optimization problems. The drawbacks of Ant Colony Optimization are analysed and studied by implementing it on Travelling Sales Person problem. By various simulation experiment runs it is finally been concluded that the limitations of ACO are permanent and cannot be overcome by any of the methodologies. The optimum solutions of any problem are guaranteed by adjusting and parameters which has limited range for the choice.

Keywords: Swarm Intelligence, Ant Colony Optimization, Travelling Sales Person Problem

INTRODUCTION

Swarm Intelligence

Swarm intelligence is a branch of Artificial Intelligence. It is a concept based on self organizing and collective behaviour (1) of swarm particles like ants, bird folks, fishes, cockroach and many more. Such swarm components interact with each other stochastically and their biological behaviour converted into mathematical model. Various mathematical models of swarm intelligence technique (2) like Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Cockroach Swarm Optimization (CSO), Artificial Bee Colony (ABC) Optimization and African Buffalo Optimization (ABO) etc. Are successfully applied to the combinatorial problems and the problems which NP-Hard in nature.

This paper focuses on Ant Colony Optimization (ACO) technique and its optimization issues. The purpose of this research is to dig the performance of ACO by taking all the parameters associated with algorithm via self made simulation tool by taking Travelling Sales Person (TSP) problem as NP-Hard problem case study.

Ant Colony Optimization

One of the promising swarm intelligence algorithm is Ant Colony Optimization (ACO), which is based on the simulating the food searching behaviour of real ants. ACO approach is invented by Marco Dorigo and his team in 2004 (3). The algorithm, its variants and applications of ACO are discusses in the next section.

Background

To understand the concept of ACO one needs to understand the behavior of real and artificial ants (3). Social insect societies (Ant Colony) are distributed systems that present a highly structured social organization. Ant Algorithms studies models derived from the observation of real ants' behavior, and uses these models as a source of inspiration for the design of novel algorithms for the solution of optimization and distributed control problems. The main idea as demonstrated in fig.1, is that the self-organizing principles which allow the highly coordinated behavior of real ants can be exploited to coordinate populations of artificial agents that collaborate to solve computational problems(3).

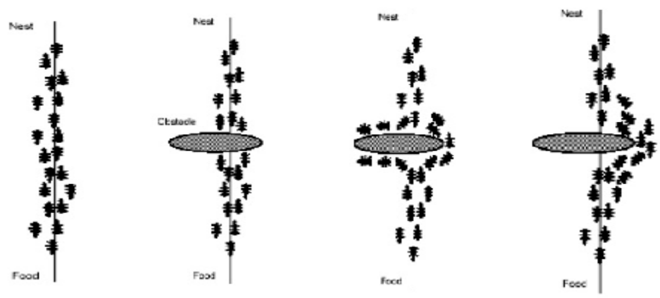

Figure 1. Real Ant's Behaviour

One of the most successful examples of ant algorithms is known as Ant Colony Optimization (ACO) (3). ACO is inspired by the foraging behavior of ant colonies and targets discrete optimization problems.

ACO Metaheuristics

Ant colony optimization (3) is a metaheuristic for difficult combinatorial optimization problems modelled after the stigmergetic communication of ants finding shortest paths to food sources. The first ACO-algorithm was Ant System (AS), introduced by Dorigo (3) in 1992. He later generalized it into the ACO metaheuristic.

A metaheuristic is a set of algorithmic concepts that can be used to define heuristic methods application to a wide set of different problems. The use of metaheuristic has increased the ability to finding very high quality solution to hard, practically relevant combinatorial optimization problems in a reasonable time. The ACO metaheuristic has been proposed as a common framework for the existing application and algorithmic variants of varieties of ant algorithms.

Ant colony optimization is a metaheuristic in which a colony of artificial ants cooperates in finding good solutions to difficult discrete optimization problems.

ACO algorithms can be used to solve both static and dynamic combinatorial optimization problems. Static problems are those in which the characteristics of the problem are given once and for all when the problem is defined, and do not changes while the problem is being solved. Dynamic problems are defined as a function of some quantities whose value is set by the dynamics of an underlying system. The problem instance changes therefore at run time and the optimization algorithm must be capable of adapting online to the changing environment.

Algorithm

Several ACO algorithms are available for optimization. The original ant colony optimization algorithm is known as Ant System (3). The Pseudo code for general ant colony optimization is given below:

Begin

Initialize

While stopping criterion not satisfied **do**

Position each ant in a starting node

Repeat

For each ant **do**

Choose next node by applying the state transition rule

Apply local pheromone update

End for

Until every ant has built a solution

Update best solution

Apply global pheromone update

End While

End

Each ant builds, starting from a source node, a solution for every problem by applying a step by step decision policy. Each node has local information is stored and the outgoing node starting from that node is read by the ant and by decision to move next node is taken stochastically. At the beginning constant pheromone is taken into consideration and then using the probability formula decision is taken that which node to move next. The following formula is used to decide which node to be visited next.

$$\rho^{k}{}_{ij} = \frac{[\tau ij]^{\alpha}[\eta ij]^{\beta}}{\sum_{l \in \mathcal{N}i^{k}}[\tau ij]^{\alpha}[\eta ij]^{\beta}} \qquad \text{if } j \in \mathcal{N}^{k} \text{ …… … (iii)}$$

The parameters used above can be specified as under,

$\eta_{ij} = 1/d_{ij}$ which specifies a heuristic value decided before start of the algorithm

α, β = Used determines the relative effect of the pheromone value and the heuristic data.

$\mathcal{N}_i^k$ = probable neighbourhood of ant k when being at at the current city it is a set of city that k^{th} ant has yet not visited.

After every run of the single ant, the pheromone value is updated and the next round of food searching criteria is solely based on the updated value of the pheromone. The process gets repeated until the best solution is found or there is no pheromone updates. Two constants α and β decide the pheromone evaporation values and it highly affects the performance of the algorithm.

LITERATURE REVIEW

Ant colony are been widely applied since 1995 to till date on the various combinatorial problems like Travelling Sales Person problem, Job Shop Scheduling, queuing and many more. The reviewed literature aimed to focus on application aspects of ant colony optimization algorithm and its hybridization with other concept like genetic algorithm or neural network concepts.

Initially, ACO algorithm is applied to solve combinatorial optimization problem. As discussed in (3), Travelling Sales Person (TSP) problem is selected for the demonstration purpose. Instance taking from online benchmark repository, TSPLIB is tested on ant algorithm and successful results discovered in comparison with other evolutionary algorithm.

Then after various application were discovered like use of ACO in Area Traffic Control(4), scheduling manufacturing units(4) etc. ACO is tested on the field of genomics (4) and forecasting of traffic control as well. Variants of ACO are also tested on document classification systems(5).

ACO algorithm is successfully implemented on project resource optimization algorithm(6). In this project, ACO perform well in accordance with the constants used in the algorithm like α, β and evaporation rate constants. Author discovered the best solution by adjusting these parameter and in future suggest the balancing act between exploitation and exploration features of the algorithm.

A concept called population based ACO (P-ACO) was implemented on the problems of TSP and QAP (7), by adding the local searching mechanism. They defend the removal of early stagnation behaviour of algorithm using this P-ACO algorithm. Job Shop Scheduling problem (8) was implemented using Ant Colony and the performance is compared with greedy algorithm. In reference (9), the applications of ACO in the problem like Travelling Sales Person Problem (TSP), Vehicle Routing Problem (VRP), Job Shop Scheduling (JSP) were discussed and suggested the use of the algorithm in the field of telecommunication especially in mobile ad-hoc network. D. Merten et. al. (10)implemented ACO algorithm in the field of data mining and suggested to analyse the gap between the current methodology and futuristic methodologies.

Multi-objective problems (11) such as finance, automobile, aircraft design etc can be implemented by ACO with integration of genetic algorithm and other evolutionary algorithm as well. They concluded that it was yet to be decided that which algorithm suits for multi-objective problems. ACO was applied to the field of segmentation(12) too. Images of brain tumour were taken as input the sharp and successful MRI image segmentation was done using ACO. They also suggested the use of artificial neural network with ACO for more classification. An experimental study was taken using ACO for Travelling Sales Person (TSP) problem (13). The main purpose was to check the parameter setting especially α and β to generate easy and hard instance of the problem. Their target was to analyse the situation where algorithm generate the best possible solution by suitable range of α and β parameters. They suggested more work for parameter prediction in ACO. In (14), Ant-Miner algorithm were introduced for data classification rules and results were compared with CN2 algorithm. They concluded that as far as accuracy is concern there was no improvement in the predictive accuracy using ACO. A real word university time-tabling problem (15) was tested using ACO. Problem was converted into graph instances and successful scheduling was derived using ACO. Due to the parameter selection issues they ignored the comparison of genetic algorithm with the ACO. In reference (16), they had serious note on parameter adaptation methodologies of ACO algorithm. They claimed that improper selection of parameters may even worsen the performance of the algorithm for some well known combinatorial problems. A probabilistic Travelling Sales Person (PTSP) (17) were tested using ACO and they investigate that the concept was useful for the small number of instances and recommended to continue the investigation by taking deterministic approach only. In (18), traditional ACO was improved by taking heuristic parameter setting to avoid rapid convergence speed of ACO algorithm and avoid to fall algorithm into local optima. TSP problem was generalised (19) and ACO was implemented by adding some local search techniques and mutation process to avoid the algorithm to get locked into local optima.

In the latest study perfomed by Dr. Carsten (20), ACO was hybridised with neural network concept – Self Organizing Map (SOM), to get the best parameter setting and to retrieve best possible solution for combinatorial problems. The main component of ACO, a pheromone concentration was updated to improve the performance of ACO (21). The changed approach helped the algorithm to avoid solution to fall into local optima for the problem like TSP. In (22), they put force on the drawback of ACO viz. Stagnation, low convergence speed and local optimum problem. They suggest the idea of parameter tuning to improve the performance of ACO and remove the said drawbacks.

A hybridized approach (23) is introduced for ACO to correct the pheromone evaporation strategy and avoid the solution to stuck into local optima. TSP problem instances were taken from TSPLIB and tested on ACO with this new correction strategy. A improved solution for dynamic TSP is proposed in (24), by modifying the current ACO to adapt the dynamic change in the TSP problem. By setting some parameters limit, successful results are derived to implement dynamic problems using ACO. ACO with re-initialization strategy was introduced by Matej Ciba and Ivan Sekaj (25) to prevent ACO performance from pheromone saturation and consecutive stagnation. They limit the variable parameter range and the standard parameters of ACO for the test case purpose. In the recent survey (26), again the TSP problem was implemented using ACO by taking small, medium and large instances. They concluded that by appropriately setting the values of α, β and evaporation rate solution of the problem can be improved.

The entire review of literature conclude that the Travelling Sales Person (TSP) problem is a model combinatorial problem for ACO and by setting the parameters associated with the algorithm one can avoid the limitations of the algorithm. Three main limitation of the algorithm are the stagnation phase, exploration and exploitation rate and convergence speed of the algorithm.

The next section describes the simulation runs of ACO algorithm on TSP by taking different ant colony size and varied parameters.

IMPLEMENTATION OF ACO

To implement the ACO algorithm, Travelling Sales Person (TSP) Problem and its varied instances are taken for investigation. The simulation runs are performed on simulation tool developed in PhP language and machine having Windows 7 or higher operating system.

For the experimental investigation the parameter taken into consideration are size of the ant colony, number of cities, number of iterations to be performed and other parameters like α, β.

As seen in the table 1 -2 , the results does not have much deviations by keeping the value of parameter $\alpha = 0.5$ and changing the value of β from 0.1 to 1.1. In this particular implementation number of cities and ant colony size are same and 10 iteration runs are performed.

Table 1. Result analysis for 10 iterations for α = 0.5 and β = 1.1

Parameters	α = 0.5	, β = 1.1	, PDF = 0.1	, PBF = 0.15	
No. Of Cities	10	20	30	40	50
Ant Colony Size	10	20	30	40	50
Average Tour	354	779	1263	1736	2207
Best Tour Found	340	714	1209	1683	2142

Table 2. Result analysis for 10 iterations for α = 0.5 and β = 0.1

Parameters	α = 0.5	, β = 1.1	, PDF = 0.1	, PBF = 0.15	
No. Of Cities	10	20	30	40	50
Ant Colony Size	10	20	30	40	50
Average Tour	345	799	1264	1768	2332
Best Tour Found	334	765	1218	1739	2228

As seen in the table 3 – 4, by changing the values of α from 0.5 to 1.5, much better results are obtained. Initially the value of β, does not affect the performance.

Table 3. Result analysis for 10 iterations for α = 1.5

Parameters	α = 1.5	, β = 1.1	, PDF = 0.1	, PBF = 0.15	
No. Of Cities	10	20	30	40	50
Ant Colony Size	10	20	30	40	50
Average Tour	372	746	1276	1768	2257
Best Tour Found	364	729	1227	1739	2236

Table 4. Result analysis for 10 iterations for α = 1.5

Parameters	α = 1.5	, β = 0.1	, PDF = 0.1	, PBF = 0.15	
No. Of Cities	10	20	30	40	50
Ant Colony Size	10	20	30	40	50
Average Tour	364	800	1312	1766	2315
Best Tour Found	355	792	1297	1751	2242

For better analysis, the solution is represented in graphical form for large number iterations. The values of α = 1.5 for all the iterations performed as seen in fig. 1-4.

For medium size TSP problem we can see the convergence in the result for the particular value of α. It produces successful results for large number of TSP city problem also.

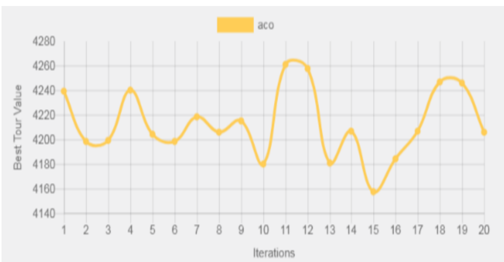

Figure 1. 20 iteration runs for 90 cities and 90 ants

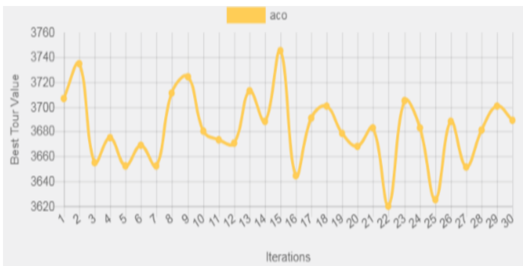

Figure 2. 30 iteration runs for 80 cities and 80 ants

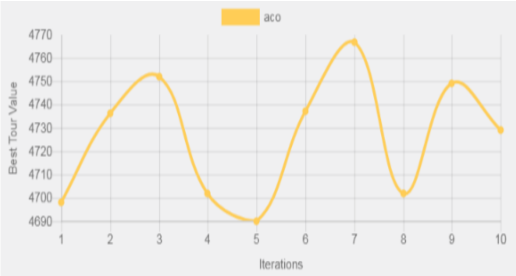

Figure 3. 10 iteration runs for 100 cities and 100 ants

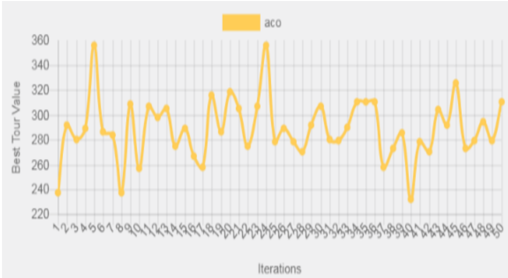

Figure 4. 50 iteration runs for 10 cities and 100 ants

If we keep α = -1.5, for less number of cities and more number of ants, result falls into stagnation phase as shown in fig.6. As seen in fig.5, it shows less stagnation for less number of ants compared to earlier result.

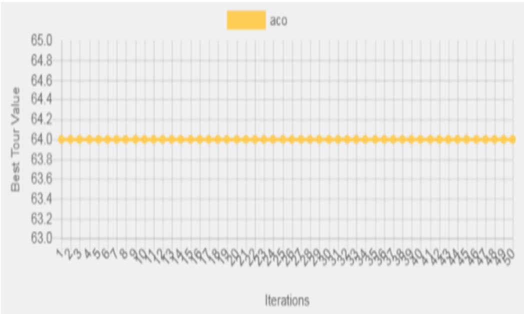

Figure 5. 50 iteration runs for 10 cities and 500 ants

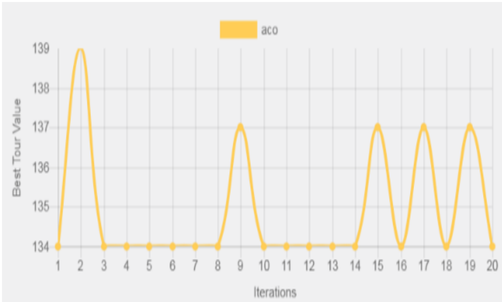

Figure 6. 20 iteration runs for 10 cities and 200 ants

ISSUES AND PROBLEMS

For the literature study and the results produced the main issues concern with the ACO algorithm are as follows:

1. The convergence speed of the algorithm changes rapidly as number of iterations goes high.
2. The major part of the algorithm is range of values of the constants α and β. For better results one must consider the suitable value of α, mostly it is 1.5. Hence, for better visualization of results one should take care of α, whatever the problem in any case.
3. For the negative value of α , say -1.5, the solution get trapped easily into stagnation phase where no further modification can be done. This is applicable mostly for the less number of cities and more number of ants.

CONCLUSION

In the article, we tried to determine the best possible solution for the ACO algorithm with respect to its varying parameters. Being a probabilistic algorithm ACO is highly dependent on its two constants α and β. It is highly desirable for any user to maintain the standard values of this parameter, whatever the problem say TSP or any other combinatorial problem like Job Shop Scheduling and Queuing. Stagnation phase of ACO algorithm put the solution into local optima which cannot be recovered without changing the values of α. The convergence speed of the solution is also rapidly changes for the case of same city and same number of ants even if we keep standard values of α and β. The final conclusion remark is to perform ACO on any NP-Hard or Combinatorial problem where near optimum solution are desirable but one should take care of these constants used into algorithm. The optimized solution cannot be achieved simply by ignoring the ranges of these constants and other static aspects of the algorithm. Author recommends using other optimization algorithm like Particle Swarm Optimization (PSO) or African Buffalo Algorithm (ABO) for more optimized solution rather than using ACO.

REFERENCES

[1.] Kennedy J, Eberhart R, Shi Y. Swarm intelligence. 2001;

[2.] Chu SC, Huang HC, Roddick JF, Pan JS. Overview of algorithms for swarm intelligence. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 2011;6922 LNAI(PART 1):28–41.

[3.] Dorigo M, Stützle T. Ant Colony Optimization [Internet]. Encyclopedia of Machine Learning. 2004. 1-24 p. Available from: http://link.springer.com /content/pdf/10.1007/978-0-387-30164-8_22.pdf

[4.] Angelo JS, Augusto DA, Barbosa HJC. Ant Colony Optimization - Techniques and Applications. 2013. 63-83 p.

[5.] Lim CP, Jain LC, Dehuri S. Innovations in swarm intelligence. Springer. 2009. 254 pp p.

[6.] Priyanto A a, Maharani W. Implementation of Ant Colony Optimization Algorithm on the.

[7.] Oliveira S, Hussin MS, Roli A, Dorigo M. A Detailed Analysis of the Population-Based Ant Colony Optimization Algorithm for the TSP and the QAP St u IRIDIA – Technical Report Series Technical Report No . 2011;(February):13–4.

[8.] Zhang J, Hu X, Tan X, Zhong JH, Huang Q. Implementation of an Ant Colony Optimization technique for job shop scheduling problem. 2006;1.

[9.] Di Caro, Gianni Gianni Di Caro FD and LMG, Ducatelle F, Gambardella LM. Swarm Intelligence for Routing in Mobile Ad Hoc Networks. Organization. 2005;76–83.

[10.] Martens D, Baesens B, Fawcett T. Editorial survey: Swarm intelligence for data mining. Mach Learn. 2011;82(1):1–42.

[11.] Afaq H, Saini S. Swarm Intelligence based Soft Computing Techniques for the Solutions to Multiobjective Optimization Problems. Comput Sci Issues. 2011;8(3):498–510.

[12.] Kamble T, Rane P. Brain Tumor Segmentation using Swarm Intelligence Approach. Int J Sci Eng Res [Internet]. 2013;4(5):716–9. Available from: http://www.ijser.org

[13. Nallaperuma S, Wagner M, Neumann F. Ant Colony Optimisation and the Traveling Salesperson Problem – Hardness , Features and Parameter Settings. GECCO 2013 Proceeding fifteenth Annu Conf Genet Evol Comput Conf. 2013;(I):13–4.

[14.] Parpinelli RS, Lopes HS, Freitas AA. Data mining with an ant colony optimization algorithm. IEEE Trans Evol Comput. 2002;6(4):321–32.

[15.] Dino Matijaš V, Molnar G, Čupić M, Jakobović D, Dalbelo Bašić B. University course timetabling using ACO: A case study on laboratory exercises. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 2010;6276 LNAI(PART 1):100–10.

[16.] Pellegrini P, Stützle T, Birattari M. A critical analysis of parameter adaptation in ant colony optimization. IRIDIA – Tech Rep Ser [Internet]. 2012;TR/IRIDIA/(September 2011). Available from: http://link.springer.com/article/10.1007/s11721-011-0061-0

[17.] Dorigo M, Gambardella LM, Bianchi L, Appliqu S. ANT COLONY OPTIMIZATION AND LOCAL SEARCH FOR THE PROBABILISTIC TRAVELING SALESMAN PROBLEM : A CASE STUDY IN. 2006;

[18]. Chi Z, Su S, Khine MA. Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm. 2011;1(5):404–9.

[19.] Yang J, Shi X, Marchese M, Liang Y. An ant colony optimization method for generalized TSP problem. Prog Nat Sci [Internet]. National Natural Science Foundation of China and Chinese Academy of Sciences; 2008;18(11):1417–22. Available from: http://dx.doi.org/10.1016/j.pnsc.2008.03.028

[20.] Mueller C. Hybrid Approach for TSP Based on Neural Networks and Ant Colony Optimization. 2015;1431–5.

[21.] Yue Y, Wang X, Application C. An Improved Ant Colony Optimization Algorithm for Solving. 2015;10(12):153–64.

[22.] Amuthan A. Improved Ant Colony Algorithms for Eliminating. 2017;97–101.

[23.] Tuba M, Jovanovic R. Improved ACO algorithm with pheromone correction strategy for the traveling salesman problem. Int J Comput Commun Control. 2013;8(3):477–85.

[24.] Mavrovouniotis M, Yang S. Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors. Appl Soft Comput J [Internet]. Elsevier B.V.; 2013;13(10):4023–37. Available from: http://dx.doi.org/10.1016/j.asoc.2013.05.022

[25.] Ciba M. Ant Colony Optimization with Re-Initialization. Autom Control Intell Syst [Internet]. 2013;1(3):59. Available from: http://www.sciencepublishinggroup.com/journal/paperinfo.aspx?journalid=134&doi=10.11648/j.acis.20130103.14

[26.] Cheong PY, Aggarwal D, Hanne T, Dornberger R. Variation of ant colony optimization parameters for solving the travelling salesman problem. 2017 IEEE 4th Int Conf Soft Comput Mach Intell [Internet]. 2017;60–5. Available from: http://ieeexplore.ieee.org/document/8279598/