A New Ant Colony Optimization Meta-Heuristic Algorithm to Tackle Large Optimization Problem

Raj Gaurang Tiwari
AZAD IET, Lucknow
UP, INDIA
Mobile: +919415561502
rajgaurang@gmail.com

Mohd. Husain
AZAD IET, Lucknow
UP, INDIA
Mobile: +919415459591
mohd.husain90@gmail.com

Sandeep Gupta
VIET, G. B. Nagar
UP, India
Mobile: +919717577497
sandeepjicool@gmail.com

Arun Pratap Srivastava
VIET, G. B Nagar
UP, India
Mobile: +919451003872
arun019@yahoo.com

ABSTRACT

Multiple ant colonies optimization is an extension of the Ant Colony Optimization framework. It offers a good opportunity to improve the ant colony optimization algorithms by encouraging the exploration of a wide area of the search space without losing the chance of exploiting the history of the search. This paper proposes a new multiple ant colonies optimization algorithm that is based on ant colony system and utilizes average pheromone evaluation mechanism. The new algorithm divides the ants' populations into multiple ant colonies and can be used to tackle large volume combinatorial optimization problems effectively. Computational tests show promises of the new algorithm.

Categories and Subject Descriptors

G.1.6 [**Mathematics of Computing**]: Optimization - *Unconstrained optimization*

General Terms

Algorithms, Performance, Experimentation

Keywords

Artificial and Swarm Intelligence, Ant Colony Optimization, Combinatorial Optimization Problems, Meta-heuristic Algorithms.

1. INTRODUCTION

Ant colony optimization (ACO) is a recent family member of the meta-heuristic algorithms and can be used to solve complex optimization problems with few modifications by adding problem-dependent heuristics. ACO is a biological inspiration simulating the ability of real ant colony of finding the shortest path between the nest and food source. It is one of the successful applications of swarm intelligence which is the field of artificial intelligence that study the intelligent behavior of groups rather than of individuals such as the behavior of natural system of social insects like ants, bees, wasps, and termites. Swarm intelligence uses stigmergy which is a form of indirect communication through the environment.

Compute'10, Jan 22-23, 2010, Bangalore, Karnataka, India

The class of complex optimization problems called combinatorial optimization problems are found in many areas of research and development. Traveling Salesman Problem (TSP), Quadratic Assignment Problem (QAP), Vehicle Routing Problem (VRP), Graph Coloring Problem (GCP), Sequential Ordering Problem (SOP), Job Secluding Problem (JSP) and Network Routing Problem (NRP) are some examples of these problems. Combinatorial optimization problems arise when the task is to find the best out of many possible solutions to a given problem, provided that a clear notion of solution quality exists. In contrast to other optimization problems, combinatorial problems have a finite number of candidate solutions. Therefore, an obvious way to solve these problems is to enumerate all candidate solutions by comparing them against each other. Unfortunately, for most interesting combinatorial optimization problems, this approach proves to be impractical since the number of candidate solutions is simply too large. The main element of ACO success is the use of a combination of priori information (heuristics) about the quality of candidate solutions (also called greedy strategy) and posteriori information (pheromone) about the goodness of the previously obtained solutions. This seems reasonable since many researches that study the characteristics of some well known optimization problems show that there is a correlation between the solution quality and the distance from the optimal solution [2][8]. Several well known ACO examples are Ant System[5], Ant Colony System[6][7], Max-Min Ant System [15].These algorithms show interesting performance and are competitive with other state of the art optimization methods.

In this paper, a new ACO meta-heuristic algorithm is proposed. The new algorithm uses multiple ant colonies and can be efficiently used to tackle large optimization problems. The rest of this paper is organized as follows. Section 2 describes the ant colony system. MACO related works are reviewed in section 3. Section 4 proposes the new algorithm. The computational results of the algorithm testing are presented in section 5 and Section 6 presents the conclusion and suggested future work.

2. ANT COLONY SYSTEM

Ant Colony System (ACS) is one of best performing ACO algorithms and has given good results for TSP and some other problems [1][6][7]. Initially, artificial ants are placed randomly on the nodes of the problem graph. The solution construction in ACS is as follows: In each algorithm's iteration, each ant computes the probability of moving to a new city not visited yet using a pseudo-random proportional rule. This rule is a trade-off between exploration and exploitation. An ant either with probability q_0 exploit the available information about previous good solutions or with probability $(1-q_0)$ explore new areas of the solution space focusing on shorter edges with pheromone rate. An ant *k* located

at node i will choose the new node j to move to according to the following.

$$j = \begin{cases} \arg \max_{l \in N_i^k} \{P_{il} H_{il}^{\beta}\} & \text{if } q \le q_0 \\ S & \text{otherwise} \end{cases} \quad (1)$$

P_{ij} is the pheromone trail on connection between node i and j and H_{ij} is the problem dependent heuristic. N_i^k is the set of remaining nodes to be visited by the k^{th} ant located at node I. β is a parameter that determines the relative importance of pheromone versus heuristic, q is a random variable distributed in [0, 1] and q_0 is a parameter and $0 \le q0 \le 1$. S is a random variable selected according to the following probabilistic rule.

$$S = \begin{cases} \dfrac{P_{ij} H_{ij}^{\beta}}{\sum_{l \in N_i^k} P_{il} H_{il}^{\beta}} & \text{if } j \in N_i^k \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

After all artificial ants have completed their tours, only the ant that finds the global best tour (the so far best tour obtained from the beginning of the algorithm's execution) reinforces the pheromone trails on the edges belong to its tour. The amount of deposited pheromone is inversely proportional to the length of the global best tour. This is called *global pheromone update* and given by:

$$P_{ij} = (1-\sigma) P_{ij} + \sigma \Delta p_{ij}^{bs} \quad (3)$$

Where ΔP_{ij}^{bs} is the pheromone quantity added to the connection (i, j) that belongs to best solution L^{bs} and given by:

$$\Delta p_{ij}^{bs} = \begin{cases} 1/L^{bs} & \text{if } (i,j) \text{ belongs to} \\ & \text{the best tour} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

σ is the trail evaporation such that (1- σ) represents the pheromone persistence. This parameter is used to avoid unlimited accumulation of pheromone trails and allows the algorithm to forget previously done bad choices.

The global pheromone update will increases the probability for other ants to use the short edges that have greater amount of pheromone trail and in turn increase the probability to build better solutions. The pheromone evaporation mechanism is applied on only the edges that have been used by an ant. Every time an ant uses an edge, it decreases the pheromone intensity on this edge. This is called local pheromone update and given by:

$$P_{ij} = (1-\gamma) P_{ij} + \gamma p_0 \quad (5)$$

Where γ is another pheromone evaporation parameter and P_0 is the initial pheromone value. Local pheromone update encourages the exploration of new areas of the search space by reducing the importance of the visited edges. While, global pheromone update encourages the exploitation of previously good solutions by giving extra weight to the edges of global best solutions.

3. MULTIPLE ANT COLONY RELATED WORK

Multiple ant colony optimization (MACO) is an extension of the ACO framework where a number of ant colonies working together to solve some combinatorial optimization problem. Varela and Sinclair [16] adopted MACO for the problem of virtual wavelength path routing and wavelength allocation. Repulsion was proposed as one way of attraction between colonies which means that ants repelled by the pheromone of other colonies. Gambardella et al.[10] proposed Multiple ACS (MACS) for the vehicle routing problem with time windows. MACS has been designed to solve vehicle routing problems with two objective functions which are the minimization of the number of tours (or vehicles) and the minimization of the total travel time, where number of tours minimization takes precedence over travel time minimization.

4. THE PROPOSED MULTIPLE ANT COLONY ALGORITHM

The proposed algorithm is based on ACS. Each colony has its own pheromone that is used as an interaction between the ants of the same colony. The interaction between ant colonies can be organized in different terms. Average pheromone interaction is proposed in this paper. The pheromone intensity of an edge is calculated as an average of the pheromone of all colonies on this edge. This means that an ant will choose the new edge base on the average of the experiences of the ants of all colonies that used this edge in the past. The new MACO is referred hereafter as MACO-AVG. The MACO-AVG algorithm is described as follows. M colonies of m ants each are working together to solve some combinatorial problem. The probabilistic decision of the ant k belongs to the colony v to move from node i to node j is an extension of equations (1) and (2) and defined as:

$$j = \begin{cases} \arg \max_{l \in N_i^k} \{f(P_{il}) H_{il}^{\beta}\} & \text{if } q \le q_0 \\ S & \text{otherwise} \end{cases} \quad (6)$$

The random variable S is selected according to the following probabilistic rule:

$$S = \begin{cases} \dfrac{f(P_{ij}) H_{ij}^{\beta}}{\sum_{l \in N_i^k} f(P_{il}) H_{il}^{\beta}} & \text{if } j \in N_i^{kv} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Where N_i^{kv} is the set of remaining nodes to be visited by the k^{th} ant of colony v located at node i and P_{ij}^{v} is the pheromone of colony v on the edge (i,j). The pheromone evaluation function $f(P_{ij})$ on the edge (i, j) defined as the average of the pheromone of all colonies and is given by:

$$f(P_{ij}) = \frac{\sum_{v=1}^{M} P_{ij}^{v}}{M} \qquad (8)$$

After all ants of all colonies complete their tours (i.e one algorithm iteration), the ant that finds the so far best solution in its colony will be allowed to deposit an amount of the colony's pheromone on the edges of its tour according to the following global pheromone update:

$$P_{ij}^{v} = (1-\sigma)P_{ij}^{v} + \sigma\Delta p_{ij}^{v.bs} \qquad (9)$$

Where $\Delta P_{ij}^{v.bs}$ is the pheromone quantity added to the connection (i, j) belonging to the best solution of v^{th} colony $L^{v.bs}$ and is given by:

$$\Delta p_{ij}^{v.bs} = \begin{cases} 1/L^{v.bs} & \text{if } (i,j) \text{ belongs to} \\ & \text{the best tour of} \\ & \text{colony } v \\ 0 & \text{otherwise} \end{cases} \qquad (10)$$

Local pheromone update is applied by each ant on the visited edges. It is very important rule as it is performed during the solution construction this helps to yield different pheromone averages for the same edge in the same iteration at different solution construction steps and it is given by:

$$P_{ij}^{v} = (1-\gamma)P_{ij}^{v} + \gamma p_0 \qquad \textbf{(11)}$$

5. EXPERIMENTAL RESULT

Full implementation of the MACO-AVG and the original ACS was developed along with this research work using visual C++ under windows XP. MACO-AVG has been tested using two TSP benchmark instance which are kroA100 and lin318 taken from TSP library. Given *n* cities and distances between them, TSP is a problem of finding minimal length of closed tour that visit each city only once. The heuristic function is the inverse of the distance, i.e., $H_{ij}=1/d_{ij}$. Number of experiments was run using 2, 3, 4 and 5 colonies. For each experiment different setting of β is used to demonstrate the algorithm robustness.

The results are averaged over 20 trials with 3000 and 10000 iterations per trial for kroA100 and lin318 respectively. Table 1 and Table 2 show the testing results of the experiments run with kroA100 and lin318 respectively. The symbol s next to the number of colonies refers that all colonies have the same value of β which is equal to 2. Different values of β are also tested and referred to by the symbol d next to the number of colonies. In this case β=2 is set for the first colony and β=3 for the second one and so on. Other parameter setting are $\sigma = \gamma = 0.1$ and $q_0 = 0.9$. MACO-AVG outperforms the original ACS one colony algorithm with the same number of ants. The performance of ACS declines as the number of ants increases. The problem was in the coordination of ants' population work to make use of the increment in the ants' number. This drawback has been overcome by the use of more than one ant colony and the use of an appropriate pheromone evaluation mechanism.

Previous studies have shown that ACS gives the best result when using ten ants. Table 3 shows an additional experiment ran with ACS on Lin318 using 10 ants for 20 trials and 500000 iterations per trial. The overall average was 43420. ACS requires 20 trials*10 ants *500000 iterations=100,000,000 construction steps to reach this result. A similar result, which is 43421.96, was obtained using MACO-AVG with 4 colonies of 10 ants each working for 20 trials of 10000 iterations each. MACO-AVG requires 20 trials *4 colonies *10 ants *10000 iterations =8,000,000 construction steps. The superior of MACO-AVG in term of construction steps required to reach the same solution is obvious as MACO-AVG was 100/8=12.5 times faster than ACS with the same setting.

6. CONCLUSION AND FUTURE WORK

The proposed algorithm divides the ants' population into multiple colonies and effectively coordinates their works. An average pheromone evaluation function is used in the process of the ant's decision making. The results show that the proposed algorithm outperforms the ACS algorithm with similar number of ants. Future work is the use of the proposed algorithmic framework on some other combinatorial optimization problems. New pheromone evaluation mechanism is another possible future direction. Another interesting future work is in the global pheromone update mechanism. In this paper the global best solution of each colony is considered. It is interesting to test the case where some colonies consider global best solution while others consider iteration best solution in the global pheromone update mechanism.

7. REFERENCES

[1]. Blum, C. & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparisons. ACM Computing Surveys, 35(3), 268-308.

[2]. Blum, C. & Dorigo, M. (2005). Search bias in ant colony optimization: On the role of competition-balanced systems. IEEE Trans. on Evolutionary Computation, 9(2), 159-174.

[3]. Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). A new ranked-based version of the ant system: a computational study. Central European Journal for Operations Research and Economics, 7(1), 25-38.

[4]. Cordon, O., Fernandez, I., Herrera, F., & Moreno, L. (2000). A new ACO model integrating evolutionary computation concepts: The Best-Worst Ant System. In: M. Dorigo, M. Middendorf, and T. Stützle, editors, Abstract Proceedings of ANTS2000- From Ant Colonies to Artificial Ants: A Series of International Workshops on Ant Algorithms, 22-29.

[5]. Dorigo, M., Maniezzo, V. & Colorni, A. (1996). The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics,* Part B, 26(1), 29-41.

[6]. Dorigo, M. & Gambardella, L. (1996). Ant Colonies for the Traveling Salesman Problem. Technical Report IRIDIA/1996-3, Universit´e Libre de Bruxelles.

[7]. Dorigo, M. et al. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans. on Evolutionary Computation, 1(1), 53-66.

[8]. Dorigo, M. & Stützle, T. (2002). The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. In: F. Glover and G. Kochenberger (Eds.), Handbook of Metaheuristics. Kluwer Academic Publishers.

[9]. Dorigo M., et al. (2006) Ant Colony optimization: Artificial Ants as Computational Intelligence Technique. IEEE Computational Intelligence Magazine, Nov. 2006.

[10]. Gambardella, L., et al. (1999). MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In D. Corne, M. Dorigo, & F. Glover (Eds.), *New Ideas in Optimization*, London: McGraw Hill.

[11]. Kawamura, H., Yamamoto, M., Suzuki, K. & Ohuchi, A. (2000). Multiple ant colonies algorithm based on colony level interactions. In *IEICE Trans. Fundamentals*, E83-A(2).

[12]. Jong, J., & Wiering, M. (2001). Multiple ant colony system for the bus-stop allocation problem. In *Proceedings of the Thirteenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC'01)*, 141–148.

[13]. Sim, K. M., et al. (2002). Multiple ant colony optimization for network routing. In Proceeding of the First International Symposium on Cyber Worlds (CW'02), 153-164.

[14]. Sim, K. M., & Sun, W. H. (2003). Ant colony optimization for routing and load-balancing: Survey and new directions. *IEEE Trans. on System, Man, and Cybernetics- Part A: System and Human,* 33(5), 560-572.

[15]. Stützle T., & Hoos H. (2000). Max-Min Ant System. Journal of Future Generation Computer Systems, 16, 889 – 914.

[16]. Varela, N. & Sinclair, M. (1999). Ant colony optimization for virtual – Wavelength-path routing and wavelength allocation. In Proceeding of Congress Evolutionary Computation (CEC), 1809-1816.

Table 1: KroA100 Results

ACS			MACO-AVG		
Colony/ ant	**Overall average**	**Std. Dev.**	**Colony/ ant**	**Overall average**	**Std. Dev.**
1/20	21675.40	235.17	2s/10	21592.70	257.15
			2d/10	21486.00	211.50
1/30	21535.65	216.86	3s/10	21467.90	209.80
			3d/10	21447.00	166.00
1/40	21675.40	255.96	4s/10	21413.75	171.00
			4d/10	21451.00	136.00
1/50	21950.85	309.33	5s/10	21377.30	131.00
			5d/10	21454.40	88.2.00

Table 2: Lin318 Results

ACS			MACO-AVG		
Colony/ ant	**Overall average**	**Std. Dev.**	**Colony/ ant**	**Overall average**	**Std. Dev.**
1/20	46136.35	677.96	2s/10	44460.35	499.50
			2d/10	44073.92	375.38
1/30	46995.85	1143.19	3s/10	44001.95	456.97
			3d/10	43658.35	337.03
1/40	47320.00	662.25	4s/10	43896.47	332.37
			4d/10	43421.96	325.58
1/50	47617.10	805.20	5s/10	44702.8	700.16
			5d/10	43421.96	325.58

Table 3: Lin318 ACS and MACO-AVG Construction Steps Comparison

Algorithm	**Colony/ ant**	**Trial/ iteration**	**Overall average**	**Construction Steps**
ACS	1/10	20/500000	43420.00	100,000,000
MACO	4d/10	20/10000	43421.96	8,000,000