

# Implementation of Different Ant based Techniques for Network Load Analysis

Nidhi Nayak  
T.I.T, Bhopal, India

Bhupesh Gour, Ph. D  
Professor, Dept. of C.S.E.  
T.I.T, Bhopal, India

Asif Ullah Khan, Ph. D  
Professor, Dept. of C.S.E.  
T.I.T, Bhopal, India

## ABSTRACT

Network Load balancing is a technique of balancing at each node the number of packets received and the number of packets forward to the other node so that the chance of network congestion problem has been reduced and bandwidth is utilized. Although there are many techniques implemented for the balancing of nodes based on maintaining a routing table at each node and is updated as the packet get forward from that node. Ant Colony Optimization is one of the techniques used in the network for the balancing of number of packets at each node. Here in this paper is proposed a comparative study of different ant colony optimization techniques implemented for the analysis of the network load balancing. Here the ant based techniques are implemented are simulated for different conditions and on the basis of which proposed the best ant based techniques for the network load balancing.

## Keywords

ACO, multi congestion, QOS, hierarchical routing, pheromone, particle velocity

## 1. INTRODUCTION

Ant Colony Optimization (ACO) is a paradigm for designing meta heuristic algorithms for combinatorial optimization problems [1]. The first algorithm which can be classified within this framework was presented 1991[21, 13] and, since then, many diverse variants of the basic principle have been reported in the literature. The necessary attribute of ACO algorithms is the combination of a priori information about the structure of a promising solution with posterior information about the structure of previously obtained good solution. An enhanced Ant colony optimization algorithm is used to resolve this difficulty in this paper. Ant Colony Optimization (ACO) is based on the behavior of ants seeking a path between their colony and a source of food, and proposed by Italy scholar M. Dorigo . The original idea is to solve a wider class of numerical problems, until now, various aspects are studied about the behavior of ants. ACO can be briefly introduced as follows. In the natural world, the behavior of ant is very simple; ants wander randomly to find food and then back to their colony while laying down pheromone trails. Once other ant's find the path, they are likely to follow the trail, but not to keep wandering at random as before. Also the followed ants can reinforce the trail if they get the food successfully. Thus, when a good path is discovered by one ant from the colony to a food source, other ants have a larger probability to pursue that path, and constructive feedback eventually lead all the ants following a single path at last.

With the rapid growth of information applications as well as the increasing bandwidth necessities, it is clear that optical networks scale to multi-layer and multi-domain. In the MRN/MLN optical transport network, traffic Engineering

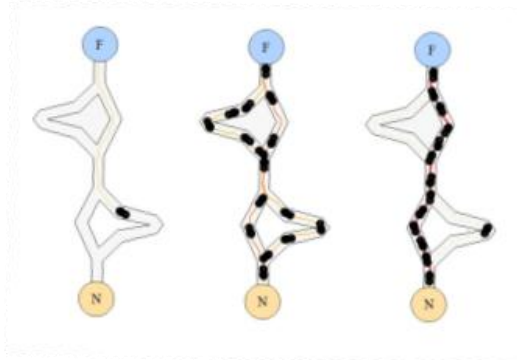
(TE) turns to be an essential requirement for Internet Service Provider (ISPs) to improve the utilization of the total network resources and to maintain a desired overall Quality of Service (QoS) with limited network resources.

Load balancing technique may improve the performance and scalability of Internet to a great extent. Many researchers focal point on intra-domain load balancing which distributes traffic over multiple paths or server farms in a single domain. However, resources in inter-domain are more limited than intra-domain, thus load balancing is an effective strategy to avoid the resources congestion in inter-domain. Many multi-level and multi-domain route algorithms have been proposed aiming at load balancing to reduce the service request blocking, they only can generate the optimal solutions for some specific network, but original route algorithms (such as hierarchical routing algorithm) in multi level and multi-domain can't compute the global optimization path.

## 1.2 ANT COLONY OPTIMIZATION

ACO [2, 3] is a class of algorithms, whose primary part, called Ant System, was originally planned by Colomni, Dorigo and Maniezzo [4, 5, and 6]. The main underlying idea, loosely inspired by the behavior of actual ants, is that of a parallel search over several productive computational threads based on local problem data and on a dynamic memory structure containing information on the quality of previously obtained result. The collective performance emerging from the interaction of the different search threads has proved effective in solving combinatorial optimization (CO) problems. Furthermore, an ACO algorithm includes two more mechanisms: trail disappearance and, optionally, daemon events. Trail vanishing decreases all trail value over time, in order to keep away from infinite accumulation of trails over some component. Daemon actions can be used to implement centralized actions which cannot be performed by solo ants, such as the invocation of a local optimization process, or the revise of global information to be used to decide whether to bias the search process from a non-local perspective [7]. More specifically, an ant is a simple computational agent, which iteratively constructs an explanation for the instance to resolve. Partial problem solutions are seen as states. At the center of the ACO algorithm lies a loop, where at every iteration, each ant move (performs a step) from a state  $i$  to another one  $y$ , consequent to a more complete fractional solution. That is, at each step  $s$ , each ant  $k$  computes a set  $AK_s(i)$  of possible expansions to its present state, and move to one of these in probability. The probability allocation is specified as follows. For ant  $k$ , the probability  $p_{iy}^k$  of stirring from state  $i$  to state  $y$  depends on the grouping of two values:  $\tau_{iy}$  the attractiveness  $h(iy)$  of the progress, as compute by some heuristic signifying the a priori desirability of that move; the trail level  $t(iy)$  of the progress, signifying how capable it has been in the past to make that particular move: it represents therefore an a

posteriori indication of the desirability of that travel. Trails are updated frequently when all ants have finished their solution, growing or declining the level of trails corresponding to moves that were part of "good" or "bad" solution, correspondingly. The universal framework just presented has been specified in different ways by the authors working on the ACO approach. The remainder of Section 2 will summarize some of these contributions.



**Figure1: Ant Colony Optimization**

There are different properties known as ant's generation and activity:

- An ant searches for a minimum (or maximum) cost solution to the optimization problem being addressed.
- Each ant has a memory use to store up all connections used to date, and so that the path can be evaluated at the completion of solution construction.
- An ant can be assigning an initial position, for example an initial city in a TSP.
- An ant can go to any possible vertex until such time that no feasible moves exist or a termination criterion is met (usually correlating to the completion of a candidate solution).
- Ants move according to a mixture of a pheromone value and a heuristic value which is connected with every edge in the problem, the choice of where to move is usually a probabilistic one.
- When going from one vertex to a different vertex the pheromone value associated with the edge connecting these vertices can be altered (known as online step-by-step pheromone update).
- An ant can repeat a constructed path at the completion of a solution updating the pheromone values of all edges used in the solution (known as online delayed pheromone update).
- Once an answer is created, and after finishing online delayed pheromone update (if required) an ant dies, freeing all allocated resources.

## 2. RELATED WORK

In 2010 by En-Jui Chang, Kai-Yuan Jheng, Hsien-Kai Hsin, Chih-Hao Chao and An-Yeu Wu [8] gives the concept about Ant Colony Optimization (ACO) is a bio-inspired algorithm

extensively applied in optimization problems propose an ACO-based Cascaded Adaptive Routing (ACO CAR) by combining two features: 1) table reforming by eliminating redundant information of far destinations from full routing table, and 2) adaptive searching of cascade point for more exact network information. The experimental results show that ACO-CAR has lower latency and higher saturation throughput, and can be implemented with 19.05% memory of full routing table.

In 2011 by Le Lu, Shanguo Huang, Wanyi Gu [1] gives the concept about ant colony optimization algorithm based on load balancing is anticipated. Ants pursue path not just depend on pheromone alone, Here also taken available resources on the link as a aspect too. Simulations show the proposed method might decrease the traffic blocking probability, and understand load balancing inside the network.

In 1997 by Ruud Schoonderwoerd Owen Holland and Janet Bruten[9] gives the concept about a simulated network models a typical distribution of calls between arbitrary nodes; nodes carrying an excess of traffic can become congested, causing calls to fail. In calculation to calls, the network also supports inhabitants of simple mobile agents with behaviors modeled on the trail laying abilities of ants. The agents move across the network between arbitrary pairs of nodes, select their path at each transitional node according to the distribution of simulated pheromones at every node. As they go they leave simulated pheromones as a function of their distance from their source node, and the congestion encounter on their journey. Calls among nodes are routed as a function of the pheromone distributions at each intermediate node. The performance of the network is measured by the proportion of calls which fail. The ant-based system is shown to drop fewer calls than the additional methods, while exhibit many striking features of distributed control.

In 1995 by Gambardella & Dorigo[10] and In 1996 by Dorigo, Maniezzo & Colomi[11] proposed the metaphor of trail laying by ants has previously been successfully applied to certain combinatorial optimization problems such as the Traveling Salesman Problem and Job Shop Scheduling These investigations were concerned with finding one good solution to a static problem. However, the problem of load balancing in telecommunication networks is essentially dynamic. The stochastic nature of calls, and the variation in call distributions, means that the problem to be solved constantly changes with time, as different call combinations give rise to congestion in different areas of the network. It is essential to maintain network performance throughout the response of the load balancing system to a change in call distributions; therefore interested in the performance of the algorithm over a certain period of time, and not merely in the eventual performance of some fixed solution.

Markov chain Monte Carlo techniques have recently also been used to establish conditions for the success of the Metropolis algorithm in the context of optimization in 2010 by Sanyal, S, and Biswas [12]. The Metropolis algorithm is a very convenient algorithm for MCMC techniques as for this algorithm it is very easy to compute the stationary distribution.

In 2006 by Neumann and Witt [13], Doerr, Neumann, Sudholt, and in 2007 by Witt [14], and Doerr and Johannsen [15] studied a simple algorithm 1-ANT that constructs a pseudo Boolean solution according to a straightforward construction graph where an ant makes independent choices for each bit. The 1-ANT records the best solution found so

far. In case a new solution is found which is not worse, the new solution replaces the old one and pheromones are updated with respect to the new solution. This mechanism implies that each new best-so-far solution leads to only one pheromone update. The mentioned studies have shown that in case  $\rho$  is too small this leads to a stagnation behavior as the knowledge gained through improvements cannot be adequately stored in the pheromones. There is a phase transition from polynomial to exponential optimization times for decreasing  $\rho$ .

In a different line of research, Gutjahr and Sebastiani [16] and Neumann, Sudholt, and Witt [17] studied an algorithm called MMAS, where the current best-so-far solution is reinforced in every generation. This holds regardless of whether the best-so-far solution has been changed or not. This means that the algorithm might reinforce the same solution over and over again, until the best-so-far solution is replaced. In stark contrast to the 1-ANT, the increased greediness of MMAS leads to polynomial upper bounds on simple pseudo-Boolean functions. Besides these results also analyses for hybridization with local search [18] and for ACO in combinatorial optimization have appeared.

In 2008 by Neumann and Witt [19] investigated ACO algorithms for finding minimum spanning trees. They considered two different construction procedures and proved that for one procedure the use of heuristic information leads to a performance that is better than the performance of a simple evolutionary algorithm [20], in terms of the number.

In 2009 by Zhou [21] considered ACO for very simple instances of the TSP. This study was significantly extended by Kotzing, Neumann, Roglin, and Witt in 2010 [22] who considered two different construction procedures and presented an average-case result for the performance of ACO. Kotzing, Lehre, Oliveto, and Neumann [20] investigated the performance of ACO for the minimum cut problem, but they only presented negative results for pheromone-based construction procedures.

### 3. PROPOSED METHODOLOGY

The model is fully distributed, i.e. every node behaves separately as well as each ant or agent, and this denotes that every node or ant is autonomous. Figure represents the table attached to each node or ant. In the model, each node contains a table that includes information about other nodes in the system. At the initial state, the table entries are Null. In each ant tour, the ant will carry the updated information about all nodes that the ant has been passed throughout. Upon arrival of the ant at every node, the following events will be done:

1. If the node does not have the information contained in the ant table, these information will be passed to the node table as it is.
2. If the node contains information that does not be present in the ant's table, the ant table will be updated.
3. If both of them share the similar information, the recently updated one will replace the other.

#### 3.1 Max-min algorithm

Pseudocode for the Negamax version of the minimax algorithm (using an evaluation heuristic to terminate at a given depth) is specified below. The code is based on the inspection that

$$\max(a, b) = -\min(-a, -b)$$

This avoids the need for the algorithm to treat the two players separately but cannot be used for games where a player may have two turns in succession.

```

function integer minimax(node, depth)

    if node is a terminal node or depth <= 0:

        return the heuristic cost of node

     $\alpha = -\infty$ 

    for child in node:           # evaluation is identical for
    both players

         $\alpha = \max(\alpha, -\text{minimax}(\text{child}, \text{depth}-1))$ 

    return  $\alpha$ 
    
```

#### 3.2 Fuzzy rule based system

```

For each particle database.

Initialize particle

End For

Do

For each particle

compute fitness value of the particle fp

/*updating particle's best fitness value so far*/

If fp is better than pBest

set current value as the new pBest

End For

/*updating population's best fitness value so far*/

Set gBest to the finest fitness value of all particles

For each particle

compute particle velocity according equation (1)

Update particle position according equation (2)

End For

While maximum iterations OR minimum error criteria

is not attained
    
```

#### 3.3 Rank based Algorithm

```

1 G := set of pages
2 for each page p in G do
3   p.auth = 1 // p.auth is the authority score of the page p
4   p.hub = 1 // p.hub is the hub score of the page p
5 function HubsAndAuthorities(G)
6 for step from 1 to k do // run the algorithm for k steps
7   for each page p in G do // update all authority values
first
8     p.auth = 0
9   for each page q in p.incomingNeighbors do //
p.incomingNeighbors is the set of pages that link to p
10    p.auth += q.hub
11  for each page p in G do // then update all hub values
12    p.hub = 0
13  for each page r in p.outgoingNeighbors do //
p.outgoingNeighbors is the set of pages that p links to
14    p.hub += r.auth

```

#### 4. Result Analysis

Here the result analysis of different Ant Based System algorithm is presented. The comparison between Max-Min Ant Algorithm, Rank Based Ant Algorithm and Fuzzy Logic Based Ant Algorithm on the basis of different parameters such as number of packets transferred and on the basis of number of ants is given.

The table shown below is the comparative analysis of different ant colony algorithms on the basis of no. Of ants used to traverse the network and to find the average length of the best tour.

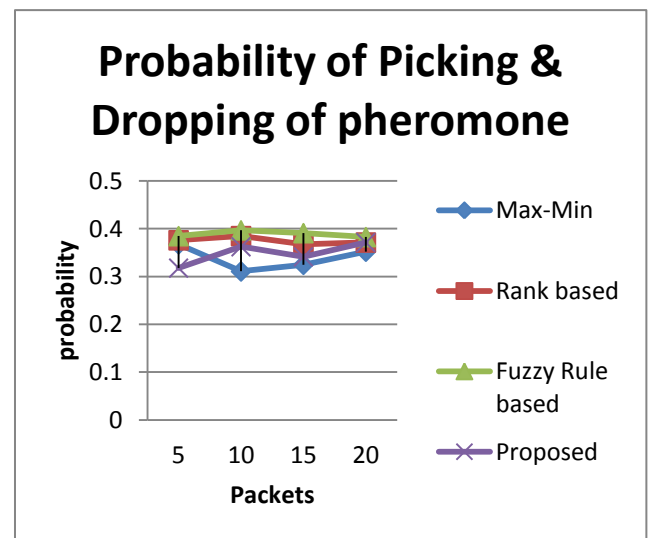
Algorithm	No. of Ants	Average Length of Best Tour
Max Min	10	37
Rank Based	10	44
Fuzzy logic based	10	45
Proposed Method	10	25

Table 1. No. Of Ants Vs Average Length of Best Tour

The table shown below is the analysis of the time complexity of different ant based algorithms as the set up the pheromone and the evaporation rate of the ants in the network. The results analysis shows the time required to forward the packets if the evaporation rate and pheromone value is set.

Algorithm	Time	Evaporation Rate	Pheromone	Packets
Max Min	38333 ms	2	5	10
Rank Based	38226 ms	2	5	10
Fuzzy logic based	38450 ms	2	5	10
Proposed Method	38473 ms	2	5	10

Table 2. Time Complexity of Ant Based Algorithms



Graph 1. Probability of Picking & Dropping

#### 5. CONCLUSION

The paper shows the comparative analysis of different ant based optimization algorithms. Here in this paper three ant based algorithms for the optimization of the network is given and on the basis of the different parameters of the ants in the network a study is shown. The result analysis shows the Average length and the time complexity of the algorithms, so in the future these ant based optimization technique can be used for various applications such that the efficiency of the network is increased.

## 6. REFERENCES

- [1] Le Lu, Shanguo Huang, Wanyi Gu, "A Dynamic Ant Colony Optimization for Load Balancing in MRN/MLN", SPIE Vol. 8310, 831010. SPIE-OSA IEEE. CCC code: 0277-786X/11/\$18. doi: 10.1117/12.904391 SPIE-OSA IEEE/Vol. 8310 831010-1 100876 China, © 2011.
- [2] M. Dorigo, Ant colony optimization web page, <http://iridia.ulb.ac.be/mdorigo/ACO/ACO.html>
- [3] M. Dorigo, G. Di Caro & L.M. Gambardella. Ant Algorithms for Discrete Optimization. *Artificial Life*, 5(2):137-172. 1999.
- [4] A. Colomi, M. Dorigo, and V. Maniezzo, Distributed optimization by ant colonies, *Proceedings of ECAL'91*, European Conference on Artificial Life, Elsevier Publishing, Amsterdam, 1991.
- [5] M. Dorigo, V. Maniezzo, and A. Colomi, The ant system: an autocatalytic optimizing process, Technical Report TR91-016, Politecnico di Milano, 1991.
- [6] M. Dorigo, Optimization, learning and natural algorithms Ph.D. Thesis, Politecnico di Milano, Milano, 1992.
- [7] M. Dorigo, T. Stützle. The ant colony optimization metaheuristic: Algorithms, applications and advances. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*. Kluwer Academic Publishers, To appear in 2002.
- [8] En-Jui Chang, Chih-Hao Chao, Kai-Yuan Jheng, Hsien-Kai Hsin, and An-Yeu Wu "ACO-Based Cascaded Adaptive Routing for Traffic Balancing in NoC Systems" Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan
- [9] Ruud Schoonderwoerd, Owen Holland and Janet Bruten, "Ant-like agents for load balancing in telecommunications networks", *Agents'97 Marina del Rey CA USA ACM* 1997.
- [10] Gambardella & Dorigo, "Ant colonies for the traveling salesman problem" Accepted for publication in *BioSystems*, TR/IRIDIA/1996-3 Université Libre de Bruxelles Belgium 1997.
- [11] Luca M. Gambardella, Marco Dorigo "Ant-Q: A Reinforcement Learning approach to the traveling salesman problem."
- [12] S. Sanyal, R. S, and S. Biswas. Necessary and sufficient conditions for success of the metropolis algorithm for optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '10)*, page 1417–1424. ACM, 2010.
- [13] F. Neumann and C. Witt. Runtime analysis of a simple ant colony optimization algorithm. In *Proceedings of the 17th International Symposium on Algorithms and Computation (ISAAC '06)*, volume 4288 of LNCS, pages 618–627. Springer, 2006.
- [14] B. Doerr, F. Neumann, D. Sudholt, and C. Witt. On the runtime analysis of the 1-ANT ACO algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '07)*, pages 33–40. ACM, 2007.
- [15] B. Doerr and D. Johannsen. Refined runtime analysis of a basic ant colony optimization algorithm. In *Proceedings of the Congress of Evolutionary Computation (CEC '07)*, pages 501–507. IEEE Press, 2007.
- [16] W. J. Gutjahr and G. Sebastiani. Runtime analysis of ant colony optimization with best-so-far reinforcement. *Methodology and Computing in Applied Probability*, 10: 409–433, 2008.
- [17] F. Neumann, D. Sudholt, and C. Witt. Analysis of different MMAS ACO algorithms on unimodal functions and plateaus. *Swarm Intelligence*, 3(1):35–68, 2009.
- [18] F. Neumann, D. Sudholt, and C. Witt. Rigorous analyses for the combination of ant colony optimization and local search. In *Proceedings of the Sixth International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS '08)*, volume 5217 of LNCS, pages 132–143. Springer, 2008.
- [19] F. Neumann and C. Witt. Ant Colony Optimization and the minimum spanning tree problem. In *Proceedings of Learning and Intelligent Optimization (LION '07)*, volume 5313 of LNCS, pages 153–166. Springer, 2008.
- [20] T. Kotzing, P. K. Lehre, P. S. Oliveto, and F. Neumann. "Ant colony optimization and the minimum cut problem. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '10)*, pages 1393–1400. ACM, 2010.
- [21] Y. Zhou. Runtime analysis of an ant colony optimization algorithm for TSP instances. *IEEE Transactions on Evolutionary Computation*, 13(5):1083–1092, 2009.
- [22] T. Kotzing, F. Neumann, H. R. "oglin, and C. Witt. Theoretical properties of two ACO approaches for the traveling salesman problem. In *Seventh International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS '10)*, volume 6234 of LNCS, pages 324–335. Springer, 2010.
- [23] Daniel Angus, "Ant Colony Optimisation: From Biological Inspiration to an Algorithmic Framework," Technical Report No. TROI3, Swinburne University of Technology, Melbourne, Australia, 2006.
- [24] M. Dorigo, V. Maniezzo, and A. Colomi, "Ant System: Optimization by a colony of cooperating agents," *IEEE Trans. on SMC*, pp. 29-41. 1996.
- [25] M. Dorigo, and C. Blumb, "Ant colony optimization theory: A survey," *Theoretical Computer Science*, vol. 344, pp. 243-278. 2005.
- [26] Sun, Y, He, P., Zhang, H., et al. "Dynamic multicast QoS routing algorithm based on ant colony algorithm," *Journal of Chongqing University of Posts and Telecommunications (Natural Science)*, 92-95, 2007.