IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 14, Issue 2 (Sep. - Oct. 2013), PP 47-52
www.iosrjournals.org

Comparison of different Ant based techniques for identification of shortest path in Distributed Network

Nidhi Nayak[1], Dr. Bhupesh Gour[2] , Dr. Asif Ullah Khan[3]
[1,2,3] (C.S.E, T.I.T/ R.G.P.V., Bhopal, India)

Abstract : *A Distributed network is one in which the data is distributed and the data from source to destination can be transferred through several nodes. When huge amount of packets are transferred through particular node congestion may occur which may result loss of packets and bandwidth also can't utilize. Hence the shortest path is chosen and routing is done dynamically so that the node can't suffer from congestion. Ant based routing techniques is an efficient one in which routing is done on the behavior of the ants and a shortest path is selected such that the packets can be send quickly and bandwidth also utilizes. Here in this paper we compare different ant based techniques for the shortest path selection from source to destination in a distributed network. Here on the basis of different Ant based techniques such as Max-Min, Rank based and Fuzzy rule based ant technique an efficient algorithm of ant technique is implemented which performs better as compared to other existing ant based techniques.*
Keywords : *ACO, multi congestion, QoS, hierarchical routing, pheromone.*

I. INTRODUCTION

With the rapid growth of information applications as well as the increasing bandwidth requirements, it is obvious 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. Ant Colony Optimization (ACO) is a paradigm for designing met heuristic algorithms for combinatorial optimization problems [1]. The first algorithm which can be classified within this framework was presented 1991[2, 3] 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 work.

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 [4]. 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.

Since the underlying topology and path-traversing mechanism in ACO exhibits many similarities to urban water distribution systems (UWDS), the concepts and methodology employed by the ACO meta heuristic can find a direct application in pipe routing optimization. If one substitutes the search for shortest path (ACO) to the search for shortest or longest path (UWDS) and treats ACO ants, states, connections and cost function to UWDS's water flow, operational state, pipes/valves and customers serviced respectively then the ACO meta heuristic can be employed in solving for the shortest (or longest) path in connected, acyclic graphs (such as pipe networks).

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.

1.1 Ant Colony Optimization

ACO [5, 6] is a class of algorithms, whose primary part, called Ant System, was originally planned by Colorni, Dorigo and Maniezzo [6, 7, and 8]. 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 [9]. More specifically, an ant is a simple computational agent, which iteratively constructs a 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 piy k of stirring from state i to state y depends on the grouping of two values: · 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.

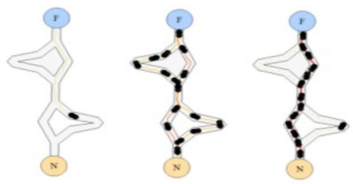


Figure1: Ant Colony Optimization

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 a answer is created, and after finishing online delayed pheromone update (if required) an ant dies, freeing all allocated resources.

II. RELATED WORK

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 2006 by Neumann and Witt [2], Doerr, Neumann, Sudholt, and in 2007 by Witt [10], and Doerr and Johannsen [11] 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 ρ 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 ρ.

In 2009 by Zhou [3] considered ACO for very simple instances of the TSP. This study was significantly extended by Kotzing, ¨ Neumann, Roglin, and Witt in 2010 [12] who considered two different construction procedures and presented an average-case result for the performance of ACO. Kotzing, Lehre, Oliveto, ¨ and Neumann [13] investigated the performance of ACO for the minimum cut problem, but they only presented negative results for pheromone-based construction procedures.

In 2010 by En-Jui Chang, Kai-Yuan Jheng, Hsien-Kai Hsin, Chih-Hao Chao and An-Yeu Wu [14] 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 1997 by Ruud Schoonderwoerd Owen Holland and Janet Bruten [15] 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[16] and In 1996 by Dorigo, Maniezzo & Colorni [17] 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 [18]. 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 a different line of research, Gutjahr and Sebastiani [19] and Neumann, Sudholt, and Witt [20] 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 more again, until the best-so-far solution is replace. 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 [21] and for ACO in combinatorial optimization have appeared.

In 2008 by Neumann and Witt [22] 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 140simple evolutionary algorithm [23], in terms of the number.

III. 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

1. Generate construction graph
2. Set the range of pheromone value to $\alpha_{min,}$ α_{max} where $\alpha_{min}>0$
3. Set m ants at randomly chosen vertices on the construction graph
4. Initialize trails to α_{max}
5. Ant arbitrary moves on the graph to constructs its solutions
6. If iteration completed then the pheromone trails consisting of the best solution will be updated
7. The pheromone trail constructs $\alpha_{min,}<= \alpha_{(i,j)}<=\alpha_{max}$ where $\alpha_{(i,j)}$ shows the pheromone trails for the connection
8. It will be imposed such the
   If $\alpha_{(i,j)}<\alpha_{min}$ then $\alpha_{min,}= \alpha_{(i,j)}$
   If $\alpha_{(i,j)}>\alpha_{max}$ then $\alpha_{max}= \alpha_{(i,j)}$
9. Continues till the termination criteria is not met

3.2 Ant colony optimization with fuzzy logic

1. Obtain a problem & represent it as a graph so that it is covered by ants
2. Assign a heuristic preference to each choice that the ant has to take in each step to generate the solution
3. Initialize the pheromone value
4. Define fitness function
   Do for each ant
   Calculate the fitness value of the ant f_a

/*updating ants best fitness value so far*/
If f_a is better than a_{best} then set current value as the new a_{best}
/*updating population best fitness value so far*/
Set g_{best} to the best fitness value of all ants
5. Repeat until the termination criteria is not met

3.3 Rank based Algorithm

1. Generate construction graph
2. Initialize pheromone value
3. While not stop condition
4. Generate m ants for a tour
5. Perform sorting on ants by their length such that
   $l_1<=l_2<=l_m$
6. An ant to the trail update is weighted according to the rank R of the ant
7. The n best ant is chosen based on the rank R
8. If W is the weight of the trail level involvement of the best tour length than it should not be exceeded by any other ant weight

IV. 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	15	42
Rank Based	15	48
Fuzzy logic based	15	51
Proposed Method	15	29

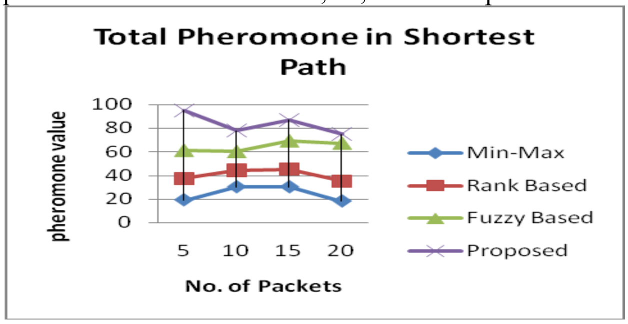
Table 1. No. Of Ants Vs Average Length of Best Tour

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	38478 ms	3	10	15
Rank Based	38678 ms	3	10	15
Fuzzy logic based	38792 ms	3	10	15
Proposed Method	38367 ms	3	10	15

Table 2. Time Complexity of Ant Based Algorithms

As shown in the graph below is the total pheromone deposited when a number of ants are moved from source to destination. The ant when moved from source to destination will drop pheromone. Here we have chosen some value of pheromone and value of evaporation for a limited amount of time, as the packet reach to the destination the amount of pheromone is calculated for 5, 10, 15 and 20 packets.



Graph 1. Total Pheromone used in shortest path

V. 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.

References

[1]. Le Lu, Shanguo Huang, Wanyi Gu, "A Dynamic Ant Colony Optimization for Load Balancing in MRN/MLN", Communications and Photonics Conference and Exhibition, pp. 1-6, 2011.

[2]. F. Neumann and C. Witt. "Runtime analysis of a simple ant colony optimization algorithm" 17th International Symposium on Algorithms and Computation (ISAAC '06), pp. 618–627, 2006.

[3]. Y. Zhou. "Runtime analysis of an ant colony optimization algorithm for TSP instances" IEEE Transactions on Evolutionary Computation, pp. 1083–1092, 2009.

[4]. M. Dorigo, Ant colony optimization web page, http://iridia.ulb.ac.be/mdorigo/ACO/ACO.html.

[5]. M. Dorigo, G. Di Caro & L.M. Gambardella. "Ant Algorithms for Discrete Optimization", Artificial Life, pp137-172. 1999.

[6]. A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies", European Conference on Artificial Life ECAL'91, pp. 134-142, 1991.

[7]. M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: an autocatalytic optimizing process", Technical Report TR91-016, 1991.

[8]. M.Dorigo, "Optimization, learning and natural algorithms", 1992.

[9]. M. Dorigo, T. Stützle. "The ant colony optimization metaheuristic: Algorithms, applications and advances", Handbook of Metaheuristics international Series in Operations Research & Management Science, Vol. 57, pp 250-285, 2003.

[10]. B. Doerr, F. Neumann, D. Sudholt, and C. Witt. "On the runtime analysis of the 1-ANT ACO algorithm" Genetic and Evolutionary Computation Conference (GECCO '07), pp. 33–40. 2007.

[11]. B. Doerr and D. Johannsen "Refined runtime analysis of a basic ant colony optimization algorithm" Congress of Evolutionary Computation (CEC '07), pp. 501–507, 2007.

[12]. T. Kotzing, F. Neumann, H. R ¨ oglin, and C. Witt. "Theoretical properties of two ACO approaches for the traveling salesman problem", Seventh International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS '10), pp. 324– 335, 2010.

[13]. T. Kotzing, P. K. Lehre, P. S. Oliveto, and F. Neumann. "Ant colony optimization and the minimum cut problem" Genetic and Evolutionary Computation Conference (GECCO '10), pp. 1393–1400, 2010.

[14]. 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" Green Circuits and Systems (ICGCS), 2010 International Conference on , pp. 317 – 322, June 2010.

[15]. Ruud Schoonderwoerd1,Owen Holland and Janet Bruten, "Ant-like agents for load balancing in telecommunications networks", First international conference on Autonomous agents AGENTS'97, pp. 209 -216, 1997.

[16]. Gambardella & Dorigo, "Ant colonies for the traveling salesman problem" Accepted for publication in BioSystems, TR/IRIDIA, 1996.

[17]. Luca M. Gambardella, Marco Dorigo "Ant-Q: A Reinforcement Learning approach to the traveling salesman problem" Twelfth International Conference on Machine Learning, pp. 252-260, 1995.
[18]. S. Sanyal, R. S, and S. Biswas. "Necessary and sufficient conditions for success of the metropolis algorithm for optimization", Genetic and Evolutionary Computation Conference (GECCO '10), pp. 1417–1424, 2010.
[19]. W. J. Gutjahr and G. Sebastiani "Runtime analysis of ant colony optimization with best-so-far reinforcement" Methodology and Computing in Applied Probability, pp. 409–433, 2008.
[20]. F. Neumann, D. Sudholt, and C. Witt. "Analysis of different MMAS ACO algorithms on unimodal functions and plateaus" Swarm Intelligence, pp. 35–68, 2009.
[21]. F. Neumann, D. Sudholt, and C. Witt. "Rigorous analyses for the combination of ant colony optimization and local search" Sixth International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS '08), pp. 132–14, 2008.
[22]. F. Neumann and C. "Witt. Ant Colony Optimization and the minimum spanning tree problem. In Proceedings of Learning and Intelligent Optimization (LION '07)" pp. 153–166, 2008.
[23]. M. Dorigo, and C. Blumb, "Ant colony optimization theory: A survey", Theoretical Computer Science", pp. 243-278, 2005.