

A NOVEL ANT COLONY ALGORITHM FOR MULTICAST ROUTING IN WIRELESS AD HOC NETWORKS

Sunita Prasad¹, Zaheeruddin² and D. K. Lobiyal³,

¹Center for Development of Advanced Computing, India

²Department of Electrical Engineering, JMI, Delhi, India

³School of Computer and System Sciences, JNU, Delhi, India

ABSTRACT

The Steiner tree is the underlying model for multicast communication. This paper presents a novel ant colony algorithm guided by problem relaxation for unconstrained Steiner tree in static wireless ad hoc networks. The framework of the proposed algorithm is based on ant colony system (ACS). In the first step, the ants probabilistically construct the path from the source to the terminal nodes. These paths are then merged together to generate a Steiner tree rooted at the source. The problem is relaxed to incorporate the structural information into the heuristic value for the selection of nodes. The effectiveness of the algorithm is tested on the benchmark problems of the OR-library. Simulation results show that our algorithm can find optimal Steiner tree with high success rate.

KEYWORDS

Multicast routing, Steiner Tree, Wireless Ad Hoc Networks, Ant Colony Optimization (ACO)

1. INTRODUCTION

The rapid developments in multimedia applications like video/audio conferencing and distance education require multicast communication. In multicast communication, the message is sent concurrently to all members of the multicast group. The multicast routing problem intends to find a minimum cost routing tree which is rooted at the source and links all the destinations. The underlying model for multicast routing is Steiner tree. Given a undirected graph $G = (V, E, c)$ consisting of V as the node set and E as the edge set and a positive edge cost $c : E \rightarrow R^+$. A subset of terminal nodes $T \subseteq V$ is also defined, the objective is to find a Steiner tree S which is a subnetwork of G such that (i) there is a path between every pair of terminals T and (ii) total cost $\sum_{e_i \in S} c(e_i)$ is minimized where $e_i \subseteq E$. The vertices $V \setminus T$ in S are called Steiner nodes.

Steiner tree is a well known NP-hard combinatorial optimization problem [1][2]. For most NP-hard problems, the performance of deterministic algorithm is not satisfactory due to high computational time required for even small instances. Metaheuristics such as Genetic Algorithm (GA) [11] and the Ant colony Optimization (ACO) [3][4][5] has been applied to solve the Steiner tree problem. In this paper, we present a novel ant colony algorithm based on problem relaxation for unconstrained multicast routing. The framework of the proposed algorithm is based on ant colony system [6]. Each ant is initially placed at the terminal node. In the first phase of the algorithm, each ant builds a complete path from the destination node to the source node. In the

second phase, the algorithm iteratively adds an entire path to the partially constructed tree rather than edges to form the tree. The probability of node selection is influenced by both pheromone and heuristic information. We utilize the structural information provided by problem relaxation to guide the decision of ants for node transition.

The ant algorithm described Singh et. al. is based on ant system (AS). Each ant starts its journey from the terminal node. The ants merge when one ant collides with other ant or it steps into the route of the other ant. In [5], collision detection mechanism is incorporated as there is a possibility of the collision of the ants. But our algorithm does not require any such mechanism.

In DCACS [3], is based on Prim's algorithm in the framework of ant colony system (ACS). The algorithm is applied on a distance complete graph (DCG). The ant starts with a randomly chosen terminal node. The ant probabilistically builds the solution after which the both the actual and the virtual edges are subjected to pheromone updation.

The rest of the paper is organized as follows. Section 2 discusses the Ant Colony Optimization (ACO) metaheuristics. Section 3 describes the proposed algorithm in detail along with the formulation. Section 4 presents the results, followed by conclusion in section 5.

2. ANT COLONY OPTIMIZATION (ACO) METAHEURISTICS

ACO was proposed by [6] and is a population based stochastic optimization technique. It is inspired by the foraging behaviour of ants and is based on stigmergic learning. In this, a population of artificial agents (ants) work collectively to generate the shortest path from the source to the destination. The solution is built step by step going through several probabilistic decisions which depends on (i) long term joint population memory (pheromone) and (ii) some additional information about the problem (heuristic information). After the solution has been constructed by the ants, some pheromone is deposited on the edges of the path which is biased towards better solution i.e. more pheromone is deposited on the edges of good solutions. Gradually, the concentration of pheromone on the edges corresponding to good solutions builds up evolving a global optimum solution. The exploitation of the pheromone value on the edges of the good solution may lead to premature convergence. To facilitate the exploration of the entire search space, pheromone trail evaporation is also incorporated in ACO. [12] gives an overview of the recent developments in ACO. Convergence proofs for ACO can be found in [7]. For better results and faster convergence, ACO are usually combined with local search algorithms. In this paper, we use problem relaxation to gain insights into the structural information of the problem.

2.1 Problem Relaxation

The minimum spanning tree (MST) with edge cost is essentially a Steiner tree without Steiner nodes. The Steiner tree heuristics are based on MST heuristics [8][9]. The approximate Steiner tree is obtained in two steps (i) generate the MST of the network and (ii) prune the MST. The distributed versions of the classical MST algorithm – Prim and Kruskal are used to obtain the Steiner tree using this method. There are two disadvantages of this technique (i) the computational cost is high since all the nodes are involved in the execution of MST algorithm and (ii) the result obtained is suboptimal.

The approach proposed in this paper uses problem relaxation to gain insights into the structure of the Steiner tree. The edges contained in MST are very likely to be part of the Steiner tree. The paper incorporates this information into the heuristic value of the ants. Thus, the transition probability of the ants is guided by the edge information provided by MST.

3. PROPOSED ALGORITHM

The proposed algorithm applies ant colony optimization to obtain Steiner tree for multicast routing. The algorithm is initialized by placing each ant on the terminal node. The algorithm consists of two phases (i) Forward set initialization and (ii) Merge path.

Forward Set Initialization: In this phase, each ant starts from the terminal node and builds a shortest path from the terminal node to the source node. The node transition probability depends on pheromone and the heuristic information. Since the input graph is not complete, it is possible that the set N of the entire available alternative that go out from the node v lead to already visited nodes. In this case, the ant is relocated to a node within its own tabu list such that it is nearest to a node in the tabu list of any other ant.

Merge Path: In this phase, we merge the path to obtain the minimum cost Steiner tree. Given a terminal t_i we first find all the nodes in the path P_i from the source to the destination t_i that are already in the existing tree. The path P_i can be joined to the tree at any of these points. The node that joins the subpath at the minimum cost is selected as the point of attachment.

2.1. Ant Colony Based Tree Construction

In this section, we describe the search behaviour of ant to build a tree. The algorithm is as follows:

Step 1: Initialization

The multicast group consist of a source node s , and a set of terminal nodes $T = \{t_1, t_2, \dots, t_m\}$. Let $n = |T|$ be the number of group members. The number of ants *antnum* is equal to n . The pheromone value on the link is initialized to a constant τ_0 . The iteration is set to a constant *MAX*. Each ant maintains its own tabu list to record the list of nodes already visited. This avoids the ant revisiting the same node again and forming a cycle. The ants are placed at each destination node t_i where $1 \leq i \leq n$ that needs to be connected and the tabu list of the ant is initialized with it.

Step 2 : State transition probability

The ant m at node i , probabilistically determines the next node j based on the state transition rule given below:

$$j = \begin{cases} \arg \max_{k \in \text{tabu}(m)} [\tau_{i,k}]^\alpha [\eta_{i,k}]^\beta, & \text{if } q \leq q_0 \quad (\text{exploitation}) \\ J & \text{otherwise} \quad (\text{exploration}) \end{cases} \quad (1)$$

where

- $\tau_{i,k}$ is a positive real quantity of the pheromone value associated with the edge connecting node i and k where k is a set of feasible nodes in the neighbourhood of node i . The pheromone value $\tau_{i,k}$ represents the accumulated knowledge about the goodness of the edge and indicates how useful it is to move to a feasible node j from the current node i .

- $\eta_{i,k}$ is the heuristic function which represents the desirability of choosing a feasible node j from current node i . The heuristic value for ant m is defined as:

$$\eta_{i,j}^m = \frac{1 + \Omega_{ij}}{c(i,j) + \gamma \cdot \psi_i^m} \quad (2)$$

where γ is a constant and ψ_i^m is the minimum cost path from node i to all the vertices in the tabu list of other ants. This causes the current ant m merge into the path of other ants as quickly as possible to form the tree. Ω_{ij} is 1 if the edge is included in MST, 0 otherwise.

- Parameter α and β weigh the relative importance of pheromone value and the heuristic function.
- q is a random number chosen with a uniform probability in $[0,1]$ and q_0 is a parameter such that $0 \leq q_0 \leq 1$. If q is smaller than q_0 , the ant will choose the next unvisited node with the maximum product of pheromone and heuristic value (exploitation step). Otherwise, the next node j is chosen as given by (3) with a probability distribution (exploration step)

$$p_m(i,j) = \begin{cases} \frac{[\tau_{i,j}]^\alpha [\eta_{i,j}]^\beta}{\sum_{k \in \text{tabu}_m} [\tau_{i,k}]^\alpha [\eta_{i,k}]^\beta} & \text{if } j \notin \text{tabu}_m \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The next node j is determined stochastically but the process favours the minimum cost edges having high amount of trail.

Step 3: Pheromone updation rule

The updating of the trail intensity on the edges is defined as follows.

$$\tau_{i,j} = (1 - \rho) \tau_{i,j} + \rho \Delta \tau_{i,j} \quad (4)$$

where ρ is a constant, called the trail evaporation rate. The increment in updating is given by the following formula.

$$\Delta \tau_{i,j} = \begin{cases} \frac{Q}{c(S_t)} & \text{if } (i,j) \in E_t \\ 0 & \text{otherwise} \end{cases}$$

where $c(S_t)$ is the cost of the current tree S_t , E_t is the edge set of the current tree and Q is a constant that matches the tree cost.

The high level description of the proposed algorithm is shown in Fig. 1. The notation used in the algorithm are given as follows

1. *JoinPath* (P_i, S_{i-1}, u) : joins the path P_i to the existing tree S_{i-1} at point u to return the current Steiner tree S_i .

2. *FindCommonNode* (P_i, S_{i-1}) : Given an existing tree S_{i-1} and P_i be the path from the source to the destination t_i . The function returns a sequence of nodes in path P_i that are already a part of the existing tree S_{i-1} .
3. *Subpath_Cost* (u, t_i) : the function returns the cost of the subpath from the common node u to the selected destination t_i .
4. *Shortest_Subpath* (u, t_i) : The function returns the subpath p_i that joins the destination t_i to the tree S_i at a tree node u .

Main procedure

Input : A connected graph $G = (V, E, c)$, terminal set T and a source s

Output : A minimal cost Steiner tree S

1. /* Initialization phase */

 Place the ant on each node in the terminal set T and put the node into its tabu-list

 Compute the MST of G

2. /* Main Algorithm */

 while $loop < MAX$ do

 ConstructSteinerTree (G, T, s)

 Update the trail intensity on every edge (i,j) by (4)

 Update the current best solution

$loop++$

 Return the current best solution

Fig. 1. Ant Colony Based Algorithm for Steiner Tree

```

Procedure ConstructSteinerTree ( $G, T, s$ )
Input : A connected graph  $G = (V, E)$  and a terminal set  $T$  and a source  $s$ 
Output : A Steiner tree  $S$ 

1. /* Phase 1 : Construction of the initial forwarding path from the destination  $t_i$  to the source */
for  $m=1$  to  $antnum$ 
    currentnode= $T_m$ 
    while currentnode $\neq s$  do
        determine the nextnode  $j$  based on (1)
        if nextnode  $\neq \emptyset$ 
            currentnode=nextnode
        else
            relocate( $m$ )
        end-if
        Add the edge  $(i, j)$  into the path  $P_m$  of ant  $m$ 
    end-while
end-for

2. /* Merge Path */
 $S_1 = P_1$ 
for  $i=2$  to  $n$ 
    mincost=inf;
     $Z = FindCommonPoint (P_i, S_{i-1})$ 
    if  $|Z| > 0$ 
        for each  $u$  in  $Z$  do
            cost = Subpath_Cost( $u, t_i$ )
            if mincost > cost
                mincost=cost
                 $p_i = Shortest\_Subpath (u, t_i)$ 
            end-if
        end-for
         $S_i = JoinPath (P_i, S_{i-1}, u)$ 
    else
         $S_i = JoinPath (P_i, S_{i-1}, s)$ 
    end-if
end-for

3. Prune (T) /* Prune the tree to obtain the minimal Steiner tree */

```

Fig. 2. ConstructSteinerTree subprocedure

3. RESULTS

The effectiveness of the proposed algorithm is tested using MATLAB simulations. The problem set B from the OR-library is used as the data set [10]. The parameters of ant colony is set empirically as $\alpha = 1$, $\beta = 4$, $\rho = 0.1$, $q_0 = 0.9$, $Q = 100$. The trail on all edges is initialized to a very small value τ_0 at the beginning of the algorithm. The maximum iteration is set as 500. The stopping criterion of our algorithm is either the maximum iteration or a fixed number of generations without improvement in the solution. Such a number is fixed as 100. Initially the movement of ants is primarily based on the heuristic information but subsequently the pheromone information is also used to build the solution. The simulation scenario for B01 is shown in Fig. 3. The nodes are randomly placed in an area of $50 \times 50 \text{ m}^2$. The obtained results are tabulated in Table 1. The results of the proposed algorithm are compared with the ant based algorithm reported in [5] using a fixed sequence approach for selection. The results suggest that the proposed algorithm is able to find the optimal results with high success rate.

Table 1: Results for B-Test Data

Graph Data				Results		
Test Data Set	V	E	T	Ant Algo [5]	Proposed Algo Best Value	Proposed Algo Average Value
B01	50	63	9	82	82	82
B02	50	63	13	83	83	83
B03	50	63	25	138	138	140
B06	50	100	25	-	122	125
B08	75	94	19	110	104	104
B09	75	94	38	230	225	226
B11	75	150	19	103	88	88
B12	75	150	38	-	176	179
B14	100	125	25	242	235	236
B15	100	125	50	350	320	321
B16	100	200	17	145	127	132

[-] results not available in [5]

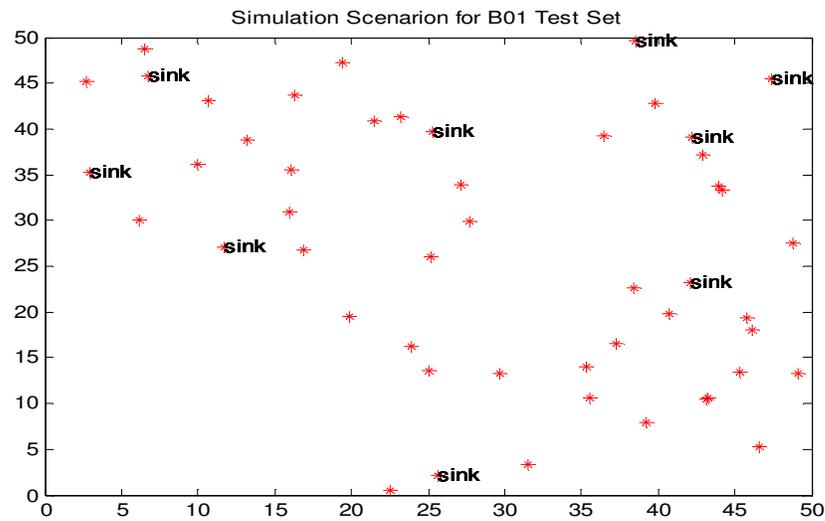


Fig 3. The simulation scenario for B01 test data set

4. CONCLUSION

The paper proposed a novel ant colony based algorithm for unconstrained Steiner tree in wireless ad hoc networks. The proposed ant based algorithm uses problem relaxation to incorporate the structural information into the heuristic value for node transition. The algorithm was tested on the standard test data set of the OR-library. The results suggest that the proposed algorithm is able to find the optimal results with high success rate. The future work is to further enhance the algorithm for constrained Steiner tree in wireless ad hoc networks and also extend it for dynamic multicast groups.

REFERENCES

- [1] M. R. Garey and D.S. Johnson, "Computers and Intractability : A Guide to the Theory of NP Completeness", W. M. Freeman, 1979.
- [2] F. Hwang and D. Richards, "Steiner Tree Problems", Networks, vol 22, pp 55-89, 1992
- [3] X. Hu, J. Zhang and L. Zhang, "Swarm Intelligence Inspired Multicast Routing : An Ant Colony Optimization Approach", LNCS, pp. 51-60, 2009.
- [4] L. Luyet, S. Varone and N. Zufferey, "An Ant Algorithm for Steiner Tree Problem in Graphs", LNCS, pp. 42-51, 2007
- [5] G. Singh, S. Das, S. Shekhar and S. Pujar, "Ant Colony Algorithms for Steiner Trees: An Application to Routing in Sensor Networks", Book Chapter in Recent Developments in Biologically Inspired Computing, IGI global publishing, pp. 181-206, 2005
- [6] M. Dorigo, G. Caro and L. Gambardella, "Ant Algorithms for Discrete Optimization", Artificial Life, vol 5(2), pp. 137-192, 1999.
- [7] T. Stuetzle and M. Dorigo, "A Short Convergence Proof for a class of ACO Algorithms" IEEE Transactions on Evolutionary Computation, vol 6(4), pp. 358-365, 2002

- [8] L. Kuo, G. Markowsky and L. Berman, "A fast algorithm for steiner trees", *Acta Informatica*, vol 15, pp. 141-145, 1981.
- [9] H. Takahashi and Matsuyama, "An Approximate solution for Steiner Tree problem in Graphs", *Math Japonica*, vol 24(6), pp. 573-577, 1980
- [10] J. Beasley: "OR-Library-Distributing Test Problem by electronic mail" *Journal of Operational Research Society*, vol 41, pp. 1061-0172, 1990.
- [11] A. Haghghat et. al., "GA Based heuristic Algorithms for Bandwidth Delay Constrained Least Cost Multicast Routing", *Computer Communications*, vol 27(1), pp. 111-127, 2004
- [12] M. Dorigo and T. Stutzle, "Ant Colony Optimization", MIT Press, 2004