Application of An improved Ant Colony Optimization on Multicast Routing Problem

Liu Yanchun\(^1,2\), Xu Zhendong\(^1,2\), Yang Bo\(^2\), and Zhang Yi\(^3\)
\(^1\)JiLin University, ChangChun, JiLin Province P.R.of China
Email: jlclyc@126.com
\(^2\)Aviation University of Air Force, ChangChun, JiLin Province P.R.of China
\(^3\)Jilin Business and Technology College Changchun , China
Email: xzd6806665@163.com, yangbojlc@163.com, whdz2000@vip.sina.com

Abstract—Three improvements on ant colony optimization (ACO) are presented in this paper. The improvements are given as follows: A novel optimized implementing approach is designed to reduce the processing costs involved with routing of ants in the conventional ACO. Based on the model of network routing, the set of candidates is limited to the nearest \(c\) points in order to reduce the counting of other points. Marked the flags on the blocked points in order to prevent selecting these points. Simulations show that the speed of convergence of the improved algorithm can be enhanced greatly compared with the traditional algorithm.

Index Terms—Ant colony algorithm; Multicast Routing; parallel strategy; self-adaptive; route strategy.

I. INTRODUCTION

Multicast consists of simultaneous data transmission from a source node to a subset of destination nodes in a computer network\(^1\). Multicast routing algorithms have recently received great attention due to increased use of recent point-to-multipoint applications, such as radio and TV transmission, on-demand video, teleconferences and so on. A dynamic multicast problem considers several traffic requests. In order to avoid hot spots and to balance the network load, a common approach is used to minimize the most heavily link utilization in the network or maximize link utilization. Therefore, cost minimization of the tree of each multicast group, which is given by the sum of the cost of the used links, is also desired. It is known that the complexity of computing the minimum cost tree for a given multicast group is NP-hard\(^20\).

In 1991, M. Dorige presented the Ant Colony Optimization (ACO). Some important strategies such as positive feedback and hidden parallel were proposed. By using positive feedback strategy, the ACO can find the better result through parallel pheromone exchanging between ants. And by using hidden parallel strategy, jumping into the optimal solution can be prevented and the ACO is also very efficient. The researches and applications on ACO algorithm have made great progress in the past ten years. Many scholars presented some efficient methods to solve these problems\(^1,13\). A number of results have demonstrated the validity of the algorithm and its advantages in some fields\(^1,12\). However, it still has some basic problems that have only been partially solved, such as searching time is too long and it may easily jump into local optimal solution\(^10\).

In this paper, we design a multicast routing optimization algorithm based on Ant Colony Optimization which can be suitable to the networks with uncertain parameters. The focus is on determining multicast routes from a source to a set of destinations. The goal of this paper is to develop an algorithm and to find out multicast routes by simultaneously optimizing. The rest of the paper is organized as follows. Section 1 introduces the multicast routing problem. Section 2, 3 presents the multicast routing optimization algorithm based on ACO. Some simulation results are provided in Section 4. The conclusion and future research for the paper are presented in Section 5.

II. MULTICAST ROUTING PROBLEM

A network is usually represented as a weighted digraph \(G = (N, E, C)\), where \(N\) denotes the set of nodes, \(E\) denotes the set of communication links connecting the nodes and \(C\) denotes the function \(C(e) : C \subseteq R^+\) , where \(e \subseteq E\) and cost function can include bandwidth and delay. \(|N|\) and \(|E|\) denote the number of nodes and links in the network respectively\(^1,20\). We consider the multicast routing problem constraints from one source node to multi-destination nodes. Let \(M = \{n_0, u_1, u_2, \ldots, u_m\} \subseteq N\) be a set of form source to destination nodes of the multicast tree. Where \(n_0\) is source node, and \(U = \{u_1, u_2, \ldots, u_m\}\) denotes a set of destination nodes. Multicast tree \(T = (N_T, E_T)\), where \(N_T \subseteq N, E_T \subseteq E\), there exists the path \(P_T(n_0, d)\) from source node \(n_0\) to each destination node \(d \in U\) in \(T\)^\(^1,20\).

Definition 1: The cost of multicast tree \(T\) is:
\[
C(T) = \sum_{e \in E_T} C(e)
\]

Definition 2: The bandwidth of multicast tree \(T\) is the minimum value of link bandwidth in the path from source node \(n_0\) to each destination node \(d \in U\), i.e.
\[
B(T) = \min(B(e), e \subseteq E_T).
\]

Because the complexity of computing the minimum cost tree for a given multicast group is NP-hard problem, and some intelligent algorithms can get the better solution.
In this paper, we try to get the better solution by an improved ACO.

III. ANT COLONY OPTIMIZATION

In this paper, we use the traveling salesman problem (TSP) to show our optimization. Let \( V = \{1, \ldots, n\} \) be a set of cities, \( A = \{ (i,j) \mid i,j \in V \} \) be the edge set, and \( d(i,j) = d(j,i) \) be a cost measure associated with edge \((i,j) \in A\). The TSP is the problem of finding a minimal cost closed tour that visits each city once.

The object function of the traveling salesman problem is:

\[
\text{min } D = \sum_{i=1}^{n-1} d(i,i+1) + d(n,1)
\]

where \( d(i,j) \) (\( i,j=1,2,\ldots,n \)) is the tour length from city \( i \) to city \( j \).

Ant colony algorithm can be described briefly as follow. Initially put \( m \) ants into \( n \) cities randomly and each edge has an initial pheromone \( \tau_{ij}(0) \) between two cities. The first element of tabu table of each ant is initialized with the initial city of each ant, and then each ant begins to select a tour to be next city. According to the following probability function, ants select the next city.

\[
p^k_j = \frac{\left[\sum_{t \in \text{allowed}} \left\{ \frac{[r_{ij}(t)]}{[\eta_{ij}]} \right\}^{\alpha} \right]}{\sum_{t \in \text{allowed}} \left\{ \frac{[r_{ij}(t)]}{[\eta_{ij}]} \right\}^{\alpha}}, \quad j \in \text{allowed},
\]

\[= 0\], \quad \text{otherwise.} \tag{1}

where \( p^k_j \) is the probability with which ant \( k \) chooses to move from city \( i \) to city \( j \). \( \tau_{ij} \) is the pheromone, \( \eta_{ij} = 1/d_{ij} \) is the inverse of the distance, \( \alpha \) and \( \beta \) are the parameters which determine the relative importance of pheromone versus distance \((\alpha, \beta > 0)\). In this way we favor the choice of edges which are shorter and have a greater amount of pheromone. After \( n \) times circles, all ants’ tabu have been filled, at this time, we compute every ant’s tour length to find the shortest one and save and record it in order to change pheromones. Repeat this until the stop condition is reached.

In ant system, the global updating rule is implemented as follows. Once all ants have built their tours, pheromone is updated on all edges according to

\[
\Delta \tau_{ij}(t+n) = \rho \tau_{ij}(t) + \Delta \tau_{ij} \tag{2}
\]

And

\[
\Delta \tau_{ij} = \sum_{k=1}^{m} \Delta \tau_{ij}^k \tag{3}
\]

Where \( \Delta \tau_{ij} \) is the sum of new increased pheromones at this edge degree of dissipating of pheromones. \( \Delta \tau_{ij}^k \) is the amount of pheromones of the \( k \)th ant at its edge between \( t \) and \( t+n \). Calculate \( \Delta \tau_{ij}^k \) using the following equation

\[
\Delta \tau_{ij}^k = \frac{Q}{L_k} \text{ if } k \text{-th ant use edge}(i,j) \text{ init tour} \quad \text{otherwise.} \tag{4}
\]

Where \( Q \) is a const, \( L_k \) is the length of the tour performed by ant \( k \).

Because the ACO algorithm is a typical random search algorithm, variables of ACO are often set by trial and error \([11]\). The chief variables are \( \alpha \) and \( \beta \) which are related to pheromones and distance respectively. Both variables have effects on the quality of solutions. Variable \( \rho \) involves in the speed of dissipating of pheromones. The parameter \( \tau_{ij} \) is also a key value. The way of changing pheromones in ACO has some influences on efficiencies and the ability of getting results of algorithm. Dorigo gave three models: ant-cycle system, ant-quantity system and ant-density system \([11]\). The difference of the three models is the function (4). In this paper, we use ant-cycle system model because it changes pheromones using whole information and it has the best result on TSP program. The complexity of this algorithm is \( O(NC^2n^2) \), where \( NC \) is the maximum cycle times.

IV. An improved Ant Colony Optimization on Multicast Routing Problem

Many scholars presented some efficient methods to solve this problem \([15-19]\). A number of results have demonstrated the validity of the algorithm and its advantages in some fields. However, it still has some basic problems that have only been partially solved. These unsolved problems go like this: the time complexity of most these algorithms is often \( O(m+n\log n) \), searching time is too long and it may easily jump into local optimal solution \([10]\). In our improved algorithm, we optimized the AOC by reducing both the executing frequency and time complexity of routing algorithm.

The operation of choosing a point to move, namely routing, is the most frequent operation in running of ACO. The total running time of ACO depends on the time complexity of routing algorithm and its executing frequency in ACO. Therefore, the running time of ACO increases with the number of points.

Our improved algorithm named ORSR, we optimized the AOC by reducing both the executing frequency and time complexity of routing algorithm. The improvements are given as follows:

Reducing the time complexity of routing: In contrast to selecting the next point from all the points not visited, the set of candidates is limited to the nearest \( c \) points. By this optimization, the time complexity is reduced greatly. The experiment results show that this improvement performed faster in large-scale network routing problem.

The routing procedure is called \( m \times n \times (n-1)/2 \) times in one cycle of ACO, where \( m \) is the number of ants, \( n \) is the number of points. Although revised ACO improved by (1) has reduced the time complexity of the routing procedure, the times of running becomes unacceptable when \( n \) and \( m \) get larger. We introduce a method to reduce the routing frequency in each cycle of ACO. In this algorithm, the ant-cycle \([11]\) algorithm is implemented
by simulating the parallel traveling of ants at the same speed. In one cycle of the algorithm, the ant who turns back to the source first wins and other ants stop routing and this cycle stopped. In each step of a cycle, the ant whose path length is the shortest chooses a route and moves to the next point and increases its path length. This procedure repeats until an ant turns back to the city which it started from.

This is an alternative improvement to reduce the routing frequency. In this method, a variable $L_{min}$ is employed to record the current minimum solution. $L_{min}$ is initialized by $\infty$. In the running, each ant is selected to find its solution in order. When an ant finished routing, a comparison is performed on its current path length with $L_{min}$, if $L_{min}$ is the less, then the ant is abandoned in the current cycle, otherwise the ant survives.

According to the characters of the network models, we only count the distances of the points which are on the shortest path between two points. In this way, the algorithms can speedup. We flag the blocked points in order to prevent selecting the useless points in the algorithm. We use a parameter named FLAG for flagging unblocked between the two points in our algorithm. If the function $C(e)<0$, the parameter FLAG will turn into zero from one, the algorithm will not select this point.

In the experiment, we find that for large maps in which the number of points is huge, the modified ACO does not converge very fast. Therefore, the 2-opt[12] is employed in the end of each cycle of ACO to perform local optimization. Experiments showed that the route optimization can speedup the convergence of ACO and enhance the quality of solution.

V. An improved Ant Colony Optimization on Multicast Routing Problem

We compare the performance of the ACO algorithm and the ORSR algorithm in terms of performance, speed of convergence and quality of solution by several experiments. Experiments were conducted on a 2.1GHz Celeron PC with 1G bytes memory under Windows 2003 using the VC6.0 compiler. The map we used in experiments is the model of [15]. In the model, we use $G(N, E, C)$ to show a connected map without direction.

![Figure 1. The simulated example of route](image)

To compare the performance of the ACO algorithm and the ORSR algorithm, we run each algorithm ten times using the same parameter set. In the experiment, we use the ACO algorithm and the ORSR algorithm to find best solutions and count the running time, and two algorithms set with the same source point and the same destination point. Simulation results are shown in Table I. The parameter set is $m=20$, $NC=10$, $\alpha=1$, $\beta=5$, and $\rho=0.67$.

From Table I it can be seen that the improved algorithm can always get the best solutions with speedup. Therefore, the ORSR can perform better for this problem.

In the following experiments, we test the quality of solution of two algorithms in blocked network. According to reference [15], we find the min cost of the shortest route, and $C(e)$ of a edge must be 0 by cost of every edge in the shortest route deduct the min cost. The network will be blocked without doubt. Results are shown in Table II. The parameter set is taken as $m=20$, $NC=5$, $Q=1000$. All the results of running time are the mean values of 10 times of experiments.

VI. Conclusions

This paper introduces a new approach based on ACO to solve the multicast routing problem. The new proposals are able to solve a multicast routing problem in a better way. Experiment represents that its convergent speed is fast and reliable. Especially in large network, the algorithm can greatly decrease routing computation time, satisfying the topology structure of real-time communication environment, high dynamic and the requirement of network multicast routing.

Acknowledgment

The authors are grateful to the support of Education Project from Jilin Province (2007248, 2007251 and 2008411)

References


<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>ACO Length</th>
<th>ACO Time (us)</th>
<th>ORSR Length</th>
<th>ORSR Time (us)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>16</td>
<td>30.9</td>
<td>13</td>
<td>25.7</td>
</tr>
<tr>
<td>2</td>
<td>19</td>
<td>16</td>
<td>27.4</td>
<td>14</td>
<td>23.3</td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td>15</td>
<td>15.3</td>
<td>12</td>
<td>14.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Blocked Point</th>
<th>Blocked Length</th>
<th>ACO Length</th>
<th>ORSR Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>{16,20}</td>
<td>21</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>19</td>
<td>{7,9}</td>
<td>14</td>
<td>6,7</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td>{7,9}</td>
<td>15</td>
<td>{13,11}</td>
<td>11</td>
</tr>
</tbody>
</table>

TABLE II. COMPARISON OF THE SOLUTIONS OF TWO ALGORITHMS IN BLOCKED


