Cross Entropy Accelerated Ant Routing in Satellite Networks

Jinhua Cao and Margareta Stefanovic

Abstract—Low Earth Orbit (LEO) satellites are becoming increasingly important in many applications, and the Cross Entropy method has recently been proposed as a heuristic method for solving combinatorial optimization problems. In this paper, we briefly review the Cross Entropy method and then develop a new on-demand routing system named Cross Entropy Accelerated Ant Routing System (CEAARS) for regular constellations LEO satellite network. We compare the basic CEAARS algorithm (Cross Entropy Ant Routing System, also novel development) with the enhanced CEAARS algorithm by implementing simulations on an Iridium-like satellite network. The results show that CEAARS not only remarkably improves the convergence speed of achieving optimal or suboptimal paths, but also highly decreases the number of overhead ants (management packets).

I. INTRODUCTION

Satellite constellations composed of Low Earth Orbiting (LEO) satellites can meet a variety of data communication needs of business, government and individuals. Due to the presence of intersatellite links (ISLs), LEO satellites connections can be routed without requiring any terrestrial resources. Typically, one satellite has four to eight ISLs to its nearest neighbors [1]. There are three types of ISLs. The intraplane ISLs connect a satellite to two or four of its nearest neighbors within the same plane. The interplane ISLs connect a satellite to its nearest neighbors in adjacent, co-rotating planes. The cross seam ISLs, which are a special case of interplane ISLs, connect two satellites from two counter-rotating planes.

Since the distance between satellite planes changes with the movement of the satellites, the interplane ISLs are longest when satellites are over the equator and shortest when they are over the polar region boundaries. Moreover, in the near polar (or Walker star) constellations, such as Iridium and Teledesic, interplane ISLs are not maintained as satellites approach the poles, due to the adverse pointing and tracking conditions, and are re-established when satellites move to lower latitudes. Since the satellite movements cause changes in the network topology, special routing algorithms for satellite networks are needed, which can establish and maintain an optimal path between source and destination in a dynamically varying topological setting.

Recently, several routing algorithms for LEO satellite networks have been developed, which mostly focus on the static snapshot path setup and snapshot transition phase [2] [3]. Satellites forward the packets according to their routing tables, which are generated in a ground switch centrally, and the routing tables are updated according to a switching table, as the satellite network topology changes. In the above scheme, the adaptivity and robustness of a satellite network are not optimal. Moreover, high energy consumption is required to maintain all snapshot routes subject to variable topology requirements, which is inappropriate for satellite, especially nano-satellites with strict energy constraints.

The Cross Entropy (CE) method was developed by Rubinstein as a tool for rare event estimation and later adapted as a tool for combinatorial optimization [4]. When it is applied to the network optimization, the CE method can be described in terms of two steps:

1) Generate a random paths sample according to an appropriately defined mechanism.

2) Iterate and update the parameters of random mechanism based on the available data to generate a “better” sample.

Some other heuristic search methods such as genetic algorithms or simulated annealing are based on having a proximity relation that facilitates efficient local search. They are likely to get stuck and not suitable on the problems having many discontinuities. Also, since the penalty function is highly complicated because of many discontinuities, guided local search algorithm with some penalty functions on traversed areas is not advisable. Compared with the above methods, the CE method includes global methods that are typically based on iteratively creating collections of solutions, improving the quality of the collection from iteration to iteration. Moreover, the CE method guarantees convergence to the optimal solution in the large sample limit, which provide some theoretical guarantees on the performance of the algorithm [5].

In [6], Helvik and Wittner presented a distributed and asynchronous version of Rubinstein’s CE algorithm based on autoregressive updating of path performance. In [7], Dorigo compared the Cross Entropy method with Ant Colony Optimization (ACO). In this paper, a robust Cross Entropy Ant Routing System (CEARS) and its improved version Cross Entropy Accelerated Ant Routing System (CEAARS) are introduced for regular constellations LEO satellite network such as near polar (or Walker star) constellations. Both CEARS and CEAARS are adaptive to the change of network topology, and also robust to the path recovery due to links and nodes failure. CEARS is based on the global random search, so it takes much longer time to find an optimal or suboptimal path at the beginning, and consumes more...
energy. Due to the capability of sensing of the direction of destination, CEAARS has a fast convergence speed with better energy consumption. It is worth mentioning that, despite the fact that the satellite orbital movement is well defined, a probabilistic solution to the routing problem may be preferred (rather than a deterministic one, such as a pre-computed look-up strategy). The reason is the uncertainty due to the satellite orbital drifts, drag, solar pressure etc.

The paper is organized as follows. In Section II, a classical LEO satellite network architecture is analyzed by comparing the number of minimum-hop paths and total paths. In Section III, the CE method and CEAARS algorithm are presented. In Section IV, the simulations are performed in the case of topology change and failure recovery. Discussion of the results, some conclusive remarks and directions for future study are included in Section V.

II. LEO SATELLITE NETWORK ARCHITECTURE ANALYSIS

In this section, an Iridium-like constellation, shown in Fig. 1, which consists of 66 satellites with 6 planes of 11 satellites per plane in a near polar LEO, is considered as an example. Each satellite or node is capable of having at most 4 links (some links may be difficult to maintain if the satellites rotate in opposite directions), which include two intraplane ISLs forward and back and two interplane ISLs to the satellites in the left and right neighboring planes. However, cross seam ISLs are inactive, and interplane ISLs are inactive in the near-polar region. This topology can be considered as a variation of the bidirectional seamless Manhattan network [8], shown in Fig. 2. In Fig. 2, we see that any two communicating nodes can be considered as being located at the corners of a rectangular section of mesh, and the number of minimum hop paths between these two nodes can be calculated as follows:

$$N_{hops}(n1, n2) = \begin{cases} \frac{[(r(n1)−r(n2)) + |c(n1)−c(n2)|]}{|r(n1)−r(n2)| + |c(n1)−c(n2)|}, & |r(n1)−r(n2)| < 6; \\ (1−|r(n1)−r(n2)| + |c(n1)−c(n2)|)! \left| c(n1)−c(n2) \right|, & otherwise. \end{cases}$$

(1)

where $N_{hops}(n1, n2)$ denotes the number of minimum hop paths between node $n1$ and $n2$. The symbols $c(n)$ and $r(n)$ denote the column and row number of node $n$, respectively. The minimum hop paths distribution of any two nodes is shown in Fig. 3, in which the vertical axis value represents the number of minimum hop paths between node 1 and node 2, whose indices are respectively shown in two horizontal axes. From Fig. 3 we can find that a fat rectangular section with the two communicating nodes at the corners contains...
relatively more minimum hop paths. Moreover, we can use the power property of the adjacency matrix to calculate the number of total paths [9]. Then, dividing the number of minimum hop path by this number, we get the ratio between the number of minimum hop paths and the number of total paths, as shown in Fig. 4. This ratio is very small, which means it is a rare event to find an optimal path by doing a random walk.

III. CROSS ENTROPY ACCELERATED ANT ROUTING ALGORITHM

A. Cross Entropy Ant Routing System (CEARS)

In [4], Rubinstein introduced the Cross Entropy method, which is motivated by an adaptive algorithm for probabilities estimation of rare events in complex stochastic networks. Therefore, it can be used for the stochastic least cost path problem with some similarities to ACO [7]. In the first phase of the CE method, ants (or agents) walk randomly through the network from the source node to the destination node, measure the cost of the paths and deposit pheromones to represent the quality of the traversed paths. In the second phase, according to the cost of the discovered paths, the cross entropy is minimized to generate the optimal (lowest cost) path. The cost of the path at the $k^{th}$ visit can be defined as:

$$L(\pi_k) = \sum_{i,j \in \pi_k} C_{ij}$$

(2)

where $\pi_k$ is the complete path from the source to the destination at visit $k$, and $C_{ij}$ is the cost of link $(i,j)$.

A straightforward way to fulfill average path performance constraints is to use crude Monte Carlo simulation (CMC) as follows:

$$\min \gamma_t \text{ s.t. } \frac{1}{N} \sum_{k=1}^{N} \prod_{i,j \in \pi_k} H(\pi_k, \gamma_t) > \rho$$

(3)

$$H(\pi_k, \gamma_t) = e^{-\frac{L(\pi_k)}{\gamma_t}}$$

(4)

where $H(\pi_k, \gamma_t)$ is the Boltzmann performance function returning the quality of the path $\pi_k$. For a fixed control temperature $\gamma_t$, the larger the $H(\pi_k, \gamma_t)$ the higher the quality of the path $\pi_k$. The minimum solution for $\gamma_t$ will result in a certain amplification (controlled by the search focus $\rho$) of high quality paths and a minimum average of all path qualities in the current batch of $N$ paths.

However, in formula (3), one needs to find all paths to implement CMC, which is highly inefficient in a distributed satellite network. Therefore, referring to [10], formula (3) is rewritten via the autoregressive counterpart as:

$$\min \gamma_t \text{ s.t. } h_t(\gamma_t) > \rho$$

(5)

where

$$h_t(\gamma_t) = \beta h_{t-1}(\gamma_t) + (1 - \beta) H(\pi_t, \gamma_t)$$

$$\approx \frac{1 - \beta}{1 - \beta^t} \sum_{k=1}^{t} \beta^{t-k} H(\pi_k, \gamma_t)$$

where $h_0(\gamma) = 0$, and $0 < \beta \leq 1$ controls the memory of the path history. Parameter $\beta$ is similar to the vaporization rate in ACO [11]. In this way, the CE method can be implemented by performing updates after finding every new path. According to [6] the control temperature $\gamma_t$ can be expressed as:

$$\gamma_t = \frac{b_0 - L(\pi_t) e^{-L(\pi_t)/\gamma_{t-1}}}{(1 + \frac{L(\pi_t)}{\gamma_{t-1}}) e^{-L(\pi_t)/\gamma_{t-1}} + a_t - \rho}$$

(6)

where $a_0 = b_0 = 0$ and $\gamma_0 = -\frac{L(\pi_0)}{\ln \rho}$, and $a_t = \beta(a_{t-1} + (1 + \frac{L(\pi_t)}{\gamma_t}) e^{-L(\pi_t)/\gamma_t})$,

$$b_t = \beta(b_{t-1} + L(\pi_t) e^{-L(\pi_t)/\gamma_t})$$

With $\gamma_t$ and $H(\pi_k, \gamma_t)$, the probability matrix $P_{t,ij}$ can be updated to minimize the Kullback-Leibler distance as follows:

$$P_{t+1} = \text{argmax} \frac{1}{N} \sum_{k=1}^{N} H(\pi_k, \gamma_t) \sum_{i,j \in \pi_k} \log P_{t,ij}$$

(7)

with $P_0$ uniformly distributed as:

$$p_{ij,0} = \begin{cases} \frac{1}{n-1} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

(8)

and where $n$ is the size of matrix $P$, i.e. the number of nodes in the network. Similarly as in [4] we can solve (7) as:

$$p_{ij,t} = \frac{T_{i,j,t} \cdot I(i \notin U)}{\sum_{i,j \in E \setminus U} T_{i,j,t}}$$

(9)

where $E$ is the entire links set for node $i$. The symbol $U$ is the tabu list, and $T_{i,j,t}$ is the pheromone value deposited on the link $(i,j)$ at visit $t$ expressed as:

$$T_{i,j,t} = \sum_{k=1}^{t-1} I((i,j) \in \pi_k) \beta \sum_{m=k+1}^{t} I((i,j) \in \pi_m) H(\pi_k, \gamma_t)$$

$$+ I((i,j) \in \pi_t) H(\pi_t, \gamma_t)$$

(10)
where $l \neq j$ and $H(L(\pi_k), \gamma_t) = e^{-L(\pi_k)/\gamma_t}$. $I(x)$ is a directive function, as:

$$I(x) = \begin{cases} 
1 & x \text{ is true} \\
0 & \text{otherwise} 
\end{cases} \quad (11)$$

### B. Adaptive Ant Generation Rate

In CEARS, a relatively high ant generation rate is necessary to maintain up-to-date statistics of the network status. This might create significant overhead in terms of routing traffic, thus increasing the cost and possibly having a negative impact on the overall network performance. In satellite networks, cost saving is a consideration of even greater importance than in Earth-based networks. To save computation and bandwidth cost, the ant generation rate is adaptively changed according to the routing situation.

At the beginning, all forward ants visit the next hop node randomly, and the backward ant follow their own forward paths reversely and update the pheromone values. In this case, the backward rate is lower than the forward rate. When one or several optimal paths are found, the backward rate become close to the forward rate. Hence, the relationship between forward rate and backward rate can be used to decide the ant generation rate, as follows.

$$r_g(k) = \begin{cases} 
r_{\text{rgmin}}, & r_g(k) \geq \max_r \\
\frac{r_g(k)}{r_f(k)}, & \text{else} 
\end{cases}$$

where $\max_r$, $r_{\text{rgmin}}$, and $r_{\text{rgmax}}$ are fixed parameters. $r_f(k)$, $r_b(k)$, and $r_g(k)$ are respectively forward rate, backward rate, and generation rate at visit $k$.

In order to play down the impact of the bursty traffic on the forward and backward ants rates, an exponentially weighted moving average (EWMA) low-pass filter is designed to estimate the rates as,

$$r_f(k) \leftarrow W_f r_f(k-1) + (1-W_f)r_g(k-1)$$

$$r_b(k) \leftarrow W_b r_b(k-1) + (1-W_b)\frac{n_b}{T_k}$$

where $W_f$ is the time constant for the low-pass filter, $r_f(0) = 0$, $n_b$ is the number of backward ants in time $[(k-1)T, kT]$.

### C. “Close to Destination” Neighbor Nodes Set

The “Close to Destination” neighbor nodes set is a set, in which the nodes are contained in the minimum-hop path. To determine the “close to destination” neighbor nodes set with minimum hop metrics, we use the locations relation between the current satellite $n_C$ and the destination satellite $n_D$ as showed the pseudo code in Fig. 5.

The CEARS algorithm is described as follows:

1) A uniformly distributed probability matrix is initialized according to formula (8). Three types of ants: destination-oriented ants, normal ants and explorer ants, are generated at the source node and start to search forward paths to the destination node.

2) For destination-oriented ants, according to the relative position of the current node and destination node by comparing their row and column indexes, a set of “closer-to-destination” neighbor nodes are generated. For other types of ants, the neighbor set includes all neighbor nodes.

3) For normal ants and destination-oriented ants, the next hop node to be visited is selected according to a probability distribution in their own neighbor sets. If no nodes are available in the set of “closer-to-destination” neighbor nodes for destination-oriented ants, the set with all neighbor nodes will be considered. For explorer ants, the next hop node to be visited is selected randomly.

4) If the selected node has been visited, it will be avoided.

5) If the forward ants does not arrive at the destination, go to (2).

6) If the forward ants arrived at the destination, the cost of the path is calculated according to formula (2), and the temperature is updated according to formula (6).

7) Backward ants are generated and return on the reserved path and update the pheromone values and probability distribution according to formulas (10) and (9) respec-
In the above algorithm, destination-oriented ants are used for improving the convergence speed of achieving optimal or suboptimal paths. Normal ants are used for finding the optimal paths in case destination-oriented ants fail to find them. Explorer ants are used for exploring all available paths. In the satellite mesh, normal ants and explorer ants walk globally, while destination-oriented ants walk aiming at finding a minimum hop count path which is potentially simpler, faster, but suboptimal. The backward ants update the pheromone values and probability distribution by the criterion of minimizing cost, which is chosen to be the same as delay in this paper. Even though there is difference between minimum hop count path and minimum delay path, on average the delay difference is quite small in the iridium-like topology network [12]. Therefore, destination-oriented ants also play a positive role in improving the speed of achieving a set of optimal or suboptimal paths. The CEARS algorithm is the same to CEAARS but without destination-oriented ants. Here, we analyze how destination-oriented ants in the above CEAARS algorithm try to find a set of minimum-hop paths. In Fig. 2, node 17 tries to connect to node 29 with assumption that all interplane ISLs are active.

**Case 1:** All links and nodes work. Destination-oriented ants will choose node 18 or 28 as the next hopping node, and two possible paths: 17-18-29 and 17-28-29 will be found, which are both minimum-hop paths.

**Case 2:** ISL 17-28 is inactive and node 18 does not work. Since the set of “closer-to-destination” neighbors is empty, destination-oriented ants will choose node 16 or 6 as the next hopping node. If they go to node 16, a path 17-16-27-28-29 will be found, which is the minimum-hop path. If they go to node 6, a path 17-6-7-8-19-30-29 will be found.

**Case 3:** ISLs 17-28, 18-29, 19-30 are inactive. Destination-oriented ants will choose node 18 as the next hopping node. Since ISL 18-29 is inactive, the set of “closer-to-destination” neighbors is empty, and they will choose node 19 or 7 as the next hopping node. As this going on, four possible paths: 17-18-19-8-20-31-30-29, 17-18-19-20-31-30-29, 17-18-7-8-19-20-31-30-29 and 17-18-7-6-5-16-27-28-29 will be found. The minimum-hop path is 17-16-27-28-29, which means destination-oriented ants fail to find the optimal path in this case.

**IV. SIMULATION**

For performance evaluation of the CEAARS, Simulation experiments on an iridium-like topology network described in section 2 are studied on Network Simulator 2 (ns-2), a discrete event simulator. The main goal of this work is to demonstrate the efficiency of CEAARS, and there are some simplified assumptions: the cost for all intraplane and interplane ISLs are the same, the delay for one hop is 15 ms, no processing delay, interplane ISLs in the polar region are inactive, and all interplane ISLs in one interplane become...
TABLE I
INTERPLANE ISL STATUS ON DIFFERENT SNAPSHOTS

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
<th>S9</th>
<th>S10</th>
<th>S11</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>I2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>I3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>I6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>I7</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I10</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>I11</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig. 9. Average end to end delay of CEAARS with adaptive ant generation rate (Scenario 1)

Fig. 10. Ratio between No. of paths through a node and No. of all paths generated by using CEARS with fixed ant generation rate (Scenario 1)

Fig. 11. Ratio between No. of paths through a node and No. of all paths generated by using CEAARS with adaptive ant generation rate (Scenario 1)

Fig. 12. Ratio between No. of paths through a node and No. of all paths generated by using CEAARS with adaptive ant generation rate (Scenario 1)
inactive simultaneously. To clearly show the simulation results, we, based on the status of interplane ISLs, divide the network into 11 snapshots, and one snapshot only last 50 seconds. Table I shows the interplane ISLs status, where \( S_i \) means the \( i \)-th snapshot, \( I_j \) means the \( j \)-th interplane, 0 at the \( i \)-th column and \( j \)-th row means interplane ISLs of the \( j \)-th interplane at the \( i \)-th snapshot is inactive, and 1 means active. In the simulation, \( \beta = 0.95 \), the ISLs’ bandwidth is 25Mb, the source node is 3, the destination node is 58, and 50 data packets with size of 1024 bytes per packet are generated per second at the source node.

Scenario 1 (Adaptivity to topology change): No nodes and links fail. For CEARS with fixed ant generation rate: 100 normal ants and 10 explorer ants are generated per second at the source node. For CEARS with adaptive ant generation rate, maximum rate is 200, and minimum rate is 20. For CEAARS with adaptive ant generation rate, maximum rate is 100, and minimum rate is 10.

Scenario 2 (Robustness to links or nodes failure): Nodes 13, 14 and 15 fail at \( t = 80 \)s. For CEARS with fixed ant generation rate: 50 normal ants and 10 explorer ants are generated per second at the source node. For CEAARS with fixed ant generation rate: 25 destination-oriented ants, 25 normal ants and 10 explorer ants are generated per second at the source node.

Scenario 3 (Relation between the number of overhead ants and performance): No nodes and links fail. For CEARS: 20 with fixed ant generation rate, normal ants and 4 explorer ants are generated per second at the source node. For CEAARS with fixed ant generation rate: 10 destination-oriented ants, 10 normal ants and 4 explorer ants are generated per second at the source node.

In Scenario 1, Fig. 7 shows the average end to end delay result of the CEARS with fixed ant generation rate. At the first 60 seconds, the delay is around 400ms which is quite high corresponding to a theoretical minimum delay at the first two snapshots. Then, a set of optimal or suboptimal paths is found, and the delay values are much closer to the theoretical
ones at each snapshot. Fig. 8 shows the average end to end delay result of the CEARS with adaptive ant generation rate. The routing converges after about 25 seconds. The accumulative number of ants is showed in Fig. 6. It shows that the CEARS with adaptive ant generation rate, compare to the fixed ant generation rate, can achieve better performance with less overhead ants. However, the convergence time is still long. Since the delay is quite high at the beginning of the simulation and the convergence time to optimal or suboptimal paths is relatively long, CEARS is not suitable for short-time connections. Fig. 9 shows the average end to end delay result of the CEARS. At the first 150 seconds, the delay is 75ms, which is exactly the theoretical minimum delay value at the first three snapshots. From 150th to 250th second, the delay is around 105ms, which is the theoretical minimum delay value at the 4th and 5th snapshots, and so on. Since the destination-oriented ants are capable of sensing the direction to the destination, the convergence speed is remarkably improved, and the set of optimal or suboptimal paths is maintained at the whole simulation period, which means the CEAARS is suitable for both short-time and long-time connections. Moreover, the statistics on the ratio between the number of paths through a node and the number of all paths are showed in Fig. 10, Fig. 11, and 12, where \((\text{Interplane No.} – 1) \times 11 + \text{Plane No.}\) is equal to the node number in Fig. 2. The height of the bar on a node shows the ratio value between the number of paths through this node and the number of all paths generated by using CEARS or CEAARS. The result in Fig. 12 confirms that CEAARS rarely routes to nodes in non-optimal paths.

In Scenario 2, Fig. 13 and Fig. 14 show the average end to end delay results of the CEARS and CEAARS respectively. The results show that, after nodes 13, 14 and 15 failing at \(t = 80s\), A set of optimal or suboptimal paths without these nodes, theoretically from 195ms to 165ms, is updated immediately in both CEARS and CEAARS, which shows the robustness of these two algorithms.

In Scenario 3, Fig. 15 shows the average end to end delay result of the CEARS. With the decrease of ants generation rate, the CEARS takes almost 250 seconds to converge. However, as shown in Fig. 16, CEAARS has similar performance on the average end to end delay. Therefore, CEAARS can maintain a good performance with much fewer overhead ants (management packets), and hence computing cost is significantly saved.

V. Conclusion

In this paper, we studied the optimal routing problem for a LEO satellite network architecture; based on the known cross-entropy (CE) routing method, we designed two LEO satellite network routing systems based on CEARS and CEAARS algorithms, and compared their simulation performance in three different scenarios. In simulation, CEAARS shows its adaptiveness and robustness fairly well in the topology change and failure recovery scenarios. Since CEAARS is capable of sensing the direction of destination, it remarkably improves the convergence speed. As CEAARS can find optimal or suboptimal paths fairly soon at the beginning of the simulations, it is suitable for both short-term and long-term connections. The simulation on CEAARS also shows a significant saving in the number of ants with similar data packet delay and service availability. Consequently, CEAARS can be implemented as an on-demand routing system, with low memory and computational burden to LEO satellites.

However, the assumption taken in the CEAARS algorithm is that all nodes in the network are roughly aware of the relative positions of other nodes, which limits its application to some random topology network without node location information. The test scenarios in the paper evaluate the algorithm to a limited extent, and further work will be conducted as follows. First, with properly designed sharing of the location information, CEAARS can be used in irregular constellations satellite networks, in which a node does not need relative position information of other nodes beforehand, but obtains this information via communication. Second, by incorporating more conditions into the cost constraint with different impact factors, such as energy consumption, bandwidth etc., CEAARS can be tested for robustness and adaptivity in more complex network scenarios. A comparison with a baseline protocol will be conducted for performance evaluation.

References