Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3032997, 18 pages
https://doi.org/10.1155/2022/3032997

Research Article

An Ant Colony Optimization-Based Routing Algorithm for Load Balancing in LEO Satellite Networks

Xia Deng,[1] Shouyuan Zeng,[1] Le Chang,[2] Yan Wang,[1] Xu Wu,[3] Junbin Liang,[3] Jiangtao Ou,[4] and Chengyuan Fan[4]

[1] *School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China*
[2] *School of Automation, Guangdong University of Technology, Guangzhou 510006, China*
[3] *School of Computer and Electronics Information, Guangxi University, Nanning 530004, China*
[4] *AI Sensing Technology, Chancheng District, Foshan 528000, China*

Correspondence should be addressed to Le Chang; lechang@gdut.edu.cn

Received 26 January 2022; Revised 3 March 2022; Accepted 16 March 2022; Published 23 April 2022

Academic Editor: Xingwang Li

Satellite networks can provide a wider service range and lower delay than traditional terrestrial optical fiber networks. However, due to the bursty characteristic of the Internet traffic and the distributive feature of satellite links, traffic-intensive areas often suffer from link congestion while links in other areas are underutilized, i.e., the traffic imbalance problem in LEO satellite networks. In this paper, an ant colony optimization routing algorithm with window reduction for LEO satellite networks, ACORA-WR, is proposed to achieve load balancing. ACORA-WR limits the movement of the ant colony to a specific range and comprehensively considers the path distance, transmission direction, and link load to find a path with low delay and overhead. Simulation results verify that the proposed ACORA-WR scheme demonstrates high data delivery ratio and network throughput, while ensuring low average delay and network transmission overhead.

1. Introduction

With the development of on-board processing and intersatellite link technology, satellite communication networks have become a reality and attracted great attention. Low Earth Orbit (LEO) satellites are often used to form satellite networks, due to the relatively low operational height, low communication delay with ground, low link loss, and low transmission power. Iridium, Globalstar, Teledesic, Starlink, and other satellite networks are all built on LEO satellites.

Compared with terrestrial 4G/5G cellular networks, the biggest advantage of LEO satellite networks is the ability of providing seamless Internet services to the whole world [1–5]. LEO satellites can cover remote areas that are not covered by 4G/5G cellular networks and are of great value in ocean operations, scientific and technological broadband, aeronautical broadband, and disaster emergency communication. As shown in Figure 1, LEO satellites extend 4G/5G cellular networks to oceans, air, and other remote areas, which allows people to transfer data from urban areas to airplanes, cruises, and other vehicles in remote areas. However, because the land area accounts for less than 30% of the total area of the earth and the population is concentrated in large cities, the satellite data volume regarding populated areas is significantly greater than other places, so the distribution of data flow is unbalanced, i.e., the traffic imbalance problem.

The routing algorithms in LEO networks must be carefully designed to meet high-quality and large-capacity business under limited satellite resources and imbalanced traffic distribution. Compared with traditional terrestrial networks [6–8], the load balancing design for satellite communication networks is more complex. Terrestrial wired networks have

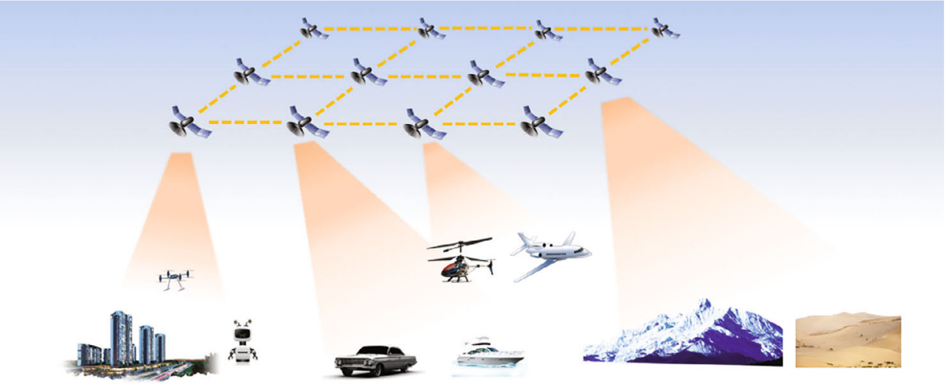


Figure 1: Application of LEO satellite networks in the global Internet.

relatively fixed topology, while the topology of the satellite communication networks is highly dynamic. The continuous movement of satellites leads to frequent change of the connections between satellites, which poses a great challenge to balancing the traffic load in such satellite networks.

Existing centralized load balancing routing schemes need to obtain path information for the entire network before calculating the optimal path, which consumes great network resources [9–11]. In contrast, distributed load balancing schemes can provide fast reaction to traffic changes [12–17]. In such category, the ant colony algorithm is a class of simulated evolutionary algorithms good at finding the optimal path under dynamic network environments [18]. Therefore, in this paper, we aim at the traffic imbalance problem in LEO satellite networks and propose an ant colony optimization routing algorithm with window reduction (ACORA-WR). Our contributions are summarized as follows.

(i) First, we have proposed an ant colony-based routing algorithm to tackle the load balancing problem in LEO satellite networks. We design congestion-avoiding heuristic information and combine it with pheromone to improve the ability of ants to enhance the local search ability and avoid stagnation. The pheromone update rule is also optimized based on path length and buffer status to improve the convergence speed to the global optimal solution

(ii) Second, we design a window-reduction mechanism to restrict the explore range of ants, which helps improve search efficiency and reduce the complexity

(iii) Last but not the least important, we implement ACORA-WR on NS2 and conduct extensive simulations to evaluate its performance. The results show that ACORA-WR achieves desirable performance on data delivery ratio and network throughput, while keeping the delay and overhead low

The remainder of this paper is organized as follows. We review the related works in Section 2. In Section 3, we explain the satellite network architecture. In Section 4, we describe the new routing algorithm. Section 5 presents the performance results of our ACORA-WR algorithm, and Section 6 summarizes the paper.

2. Related Works

Existing load-balancing routing strategies for LEO satellite networks can be mainly classified into two categories: centralized and distributed strategies.

Centralized strategies usually use a global optimization algorithm to optimize the system traffic allocation and compute the routing table. Li et al. proposed a mechanism to quantitatively estimate the link state and dynamically adjusted the weight of queuing delay [9]. The occupancy rate of each queue was divided into multiple levels, and the queuing delay was determined according to the corresponding level. HGL [10] proposed by Liu et al. combined the global and local strategy to optimally allocate IoT traffic flows. Wang et al. proposed a load balancing scheme based on Stackelberg game algorithm to make more effective use of the satellite storage space [11].

Distributed schemes are also developed because centralized strategies are difficult to implement on satellites. Na et al. proposed a distributed routing strategy for LEO satellite networks based on machine learning [12]. Taleb et al. proposed the Explicit Load Balancing (ELB) scheme which predicted the current transmission congestion by explicitly exchanging the queue usage on adjacent satellites [13]. Song et al. extended the idea of ELB and brought forward the traffic light-based routing (TLR) mechanism [14]. Papapetrou et al. proposed Location-Assisted On-demand Routing (LAOR) [15], where the satellite actively searched for a path to the destination satellite after receiving a user communication request. Karapantazis et al. extended the idea of LAOR and proposed Multiservice On demand Routing (MOR) [16]. Rao and Wang proposed Agent-based Load Balancing Routing (ALBR) [17], which used mobile agents to explore link states to randomly selected destinations.

Currently, ant colony algorithms are widely used in networks, such as avoiding the interflow and intraflow interference or balancing the loads in backbone networks [19], calculating the best route for vehicles in vehicle networks [20, 21], designing dynamic source routing algorithm meeting high QoS requirements in ad hoc networks [22], balancing energy efficiency for WSNs and IPv6 networks [23, 24], and solving the scheduling (SBS) problem of satellite broadcasting between satellites and ground stations [25]. Wang et al. proposed a load balancing scheme based on the ant colony algorithm (LBRA-CP) [18]. In LBRA-CP, the satellites sought out the congestion areas and shared the information of their positions to predict congestion, and the ant colony algorithm was used to find an optimal path for every connection request. However, ant colony algorithms still need to tackle two challenges. First, they may result in local optimal solutions. Second, the search range may be too wide, making the search efficiency lower with the increase of the number of satellites.

In this paper, we consider the total cost of the path and the size of satellite buffer and also set the upper limit of pheromone to avoid the ant colony algorithm to get the local optimal solution. Meanwhile, we propose a window reduction mechanism to increase the search speed of the ant colony, which can make full use of intersatellite links to improve data delivery ratio and throughput, while keeping a low transmission overhead and delay.

3. System Model

3.1. Satellite Networks. We model a LEO satellite network as a graph $G = (V, E)$, where V and E represent the set of satellite nodes and directed ISLs (intersatellite links), respectively. In the context of such network, we use the term “node” and “satellite” interchangeably. The types of ISLs include intraplane ISLs and interplane ISLs. Intraplane ISLs refer to the links between satellites in the same orbit, and interplane ISLs refer to those between satellites in different orbits. The intraplane ISLs remain constant with fixed length. Satellites move predictably and regularly with a constant period. The topology of satellite networks is treated as constant at an instant or within a small time duration following [26]. As shown in Figure 2, each satellite typically has two intraplane ISLs and two interplane ISLs. An intraplane ISL is used to connect a satellite with its predecessor or successor satellites in the same orbit, and interplane ISLs are used to connect two adjacent satellites in two adjacent orbits. We represent the logical address of a satellite using $N_{i,j}$, where i is the orbit number and j is satellite number within its orbit.

3.2. End-to-End Path Cost. The ISL link cost is based on the end-to-end delay, which is the summation of queuing delay and propagation delay [17]. The cost of the link between satellite i and j, i.e., on transmission link (i, j), is thus calculated as

$$\mathrm{Cost}(i, j)_t = \mathrm{PDelay}(i, j)_t + \mathrm{QDelay}(i, j)_t, \tag{1}$$

where the propagation delay $\mathrm{PDelay}(i, j)_t$ is calculated as

$$\mathrm{PDelay}(i, j)_t = \frac{d(i, j)_t}{c}. \tag{2}$$

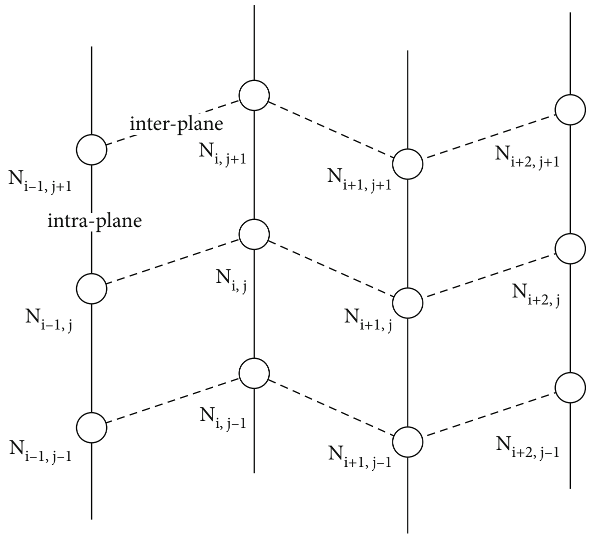


Figure 2: LEO satellite network topology.

The propagation delay $\mathrm{PDelay}(i, j)_t$ is the physical distance $d(i, j)$ of satellite link (i, j) at time t divided by velocity c, the optical transmission speed in vacuum, i.e., 3×10^8 m/s. The queuing delay $\mathrm{QDelay}(i, j)_t$ is calculated using the same method in [17]. Assuming that both the packet arrival and service pattern are Poisson processes, then the arrival and service intervals follow exponential distributions. We adopt the M/M/1 queuing model proposed in reference [27] and estimate the average number of packets in a queue using

$$\mathrm{Num}_q = \frac{\mu}{1 - \mu}, \tag{3}$$

where μ represents the mean ISL utilization following the calculation in [28].

μ_h is the mean ISL utilization when the h-th packet reaches the interface queue, calculated as

$$\mu_h = \mathrm{LinkState} + e^{-\Delta t_h} \times (\mu_{h-1} - \mathrm{LinkState}), \tag{4}$$

where LinkState is set to 0 if there are no packets in the interface queue or in transmission, and 1, otherwise. $e^{-\Delta t_h}$ represents the forgetting rate, and Δt_h is the time interval between the arrival of the h-th and $(h-1)$th packets in the queue. Therefore, the queuing delay can be estimated as

$$\mathrm{QDelay} = \mathrm{Num}_q \times \frac{S_{\mathrm{avg}}}{B}, \tag{5}$$

where S_{avg} is the average packet size in the queue and B is the link bandwidth.

$H(s, d)$ represents the set of all links from the source node s to the destination node d. The total cost from the source to the destination is

$$\mathrm{Cost}(s, d)_t = \sum_{\substack{i \in H(s,d) \\ j \in H(s,d)}} \mathrm{Cost}(i, j)_t. \tag{6}$$

We set $C(0 < C < 1)$ as the threshold and let $\varrho_{ij}(t)$ represent the waiting queue occupancy rate of link (i, j) at time t. If $\varrho_{ij}(t) > C$, $\mathrm{LinkAvailable}(i, j) = 0$ indicates that link (i, j) is congested and the ISL cannot be used. Otherwise, $\mathrm{LinkAvailable}(i, j) = 1$ indicates that ISL is used normally.

$$\mathrm{LinkAvailable}(i, j) = \begin{cases} 0 & \varrho_{ij}(t) > C \\ 1 & \text{otherwise} \end{cases}. \tag{7}$$

The goal of the end-to-end path cost model is to find the optimal path with the smallest cost for each packet from the source to destination under the above constraints. Therefore, the optimization model is formulated as

$$\begin{gathered} \text{Minimize} : \mathrm{Cost}(s, d)_t \\ \text{s.t.} \forall \mathrm{link}(i, j) \mathrm{LinkAvailable}(i, j) = 1 \\ (i, j) \in H(s, d) \end{gathered} \tag{8}$$

Our goal is to find the path with the lowest total cost from source to destination. Moreover, to ensure reliable data transmission, $(i, j) \in H(s, d)$ represents each ISL on $H(s, d)$ should be usable.

4. Ant Colony Optimization Routing Algorithm with Window Reduction

Ant colony algorithm is a probabilistic algorithm used to find the optimal path [29]. It has the characteristics of distributed calculation, positive information feedback, and heuristic search. Ant colony algorithm uses a control message (also known as ant) to collect path information for routing. The packet selects the next hop according to the probability formula. However, in the LEO satellite network environment, the ant colony algorithm may stagnate for a period of time, which makes the traffic congestion more serious. Therefore, we propose the congestion-avoiding heuristic information to alleviate this problem. On the other hand, due to the long distance and global coverage of satellite links (ISLs), in order to make the ant colony more efficient, we propose the window reduction mechanism to limit the search range of ants. In this section, we explain how we design an ant colony algorithm to fit the LEO scenarios.

4.1. Ant Colony Algorithm-Based Routing. We use data packets to simulate ants, which are divided into two kinds: forward ants and backward ants. The forward ants represent the messages from the source node to the destination node, which collect the path information from the source node to the destination node including end-to-end delay and number of hops. The backward ants represent the messages returned from the destination node to the source node, which change the routing information of each satellite node.

Table 1: The probability table structure of node i.

Flow	Adjacent node	Probability
f	j	P_{ijf}
f	k	P_{ikf}
g	l	P_{ilg}

Pheromone is the information left by ants when they pass through nodes. The forward ant calculates and determines the next hop using pheromone. When the backward ant passes through a node, it will update the pheromone of the node. With a larger number of ants passing through the node, the pheromone density will be greater. Intuitively, ants prefer to choose those nodes with greater pheromone density. The routing table of each node in the network is represented by a probability table, and the probability value is related to the pheromone density. Table 1 shows the structure of the probability table of node i, which includes three kinds of information: flow, adjacent node, and probability. The flow number is determined by source-destination pairs, and the adjacent node is the node directly adjacent to node i. When a data packet of flow f reaches satellite node i, the probability of selecting the next node j is P_{ijf}.

In the initial state, the pheromones on all paths are evenly distributed. As shown in Figure 3, a group of forward ants are sent from source S, and the forward ants are forwarded in the networks according to the pheromone. If the pheromone density is 0, it is forwarded randomly. The ants can remember the information of the path, including hops and delay, and finally reach the destination node receiver D. Obviously, the ant passing through the shortest path will reach the destination node at the fastest speed. If there are multiple identical ants, the destination node only receives the first arriving ant. When the ant arriving at the destination first reaches receiver D, receiver D will send the backward ant to the source point S, and the backward ant will return to S following the original path of the forward ant and modify the pheromone when passing through the node. The backward ant determines whether to increase or decrease the pheromone according to the communication quality of the whole path and the increment value of the pheromone according to the congestion degree of the node. As denoted by the yellow links in Figure 3, when the path is busy, the backward ant will reduce the pheromone of the node on the yellow path and thus reduce the probability of the forward ant choosing this path. At the same time, the backward ant passing through the blue path will increase the pheromone of this path and allow more data to be forwarded through the blue path. In addition, we set the red link as a congested link, which is temporarily unavailable. In this way, the iterative operation is carried out continuously to select the best path for the data packets.

4.1.1. Forward Ants. We let the source node periodically send out forward ants in the process of data transmission. A forward ant uses unicast to select the next hop node according to the state transition rules and records the travel

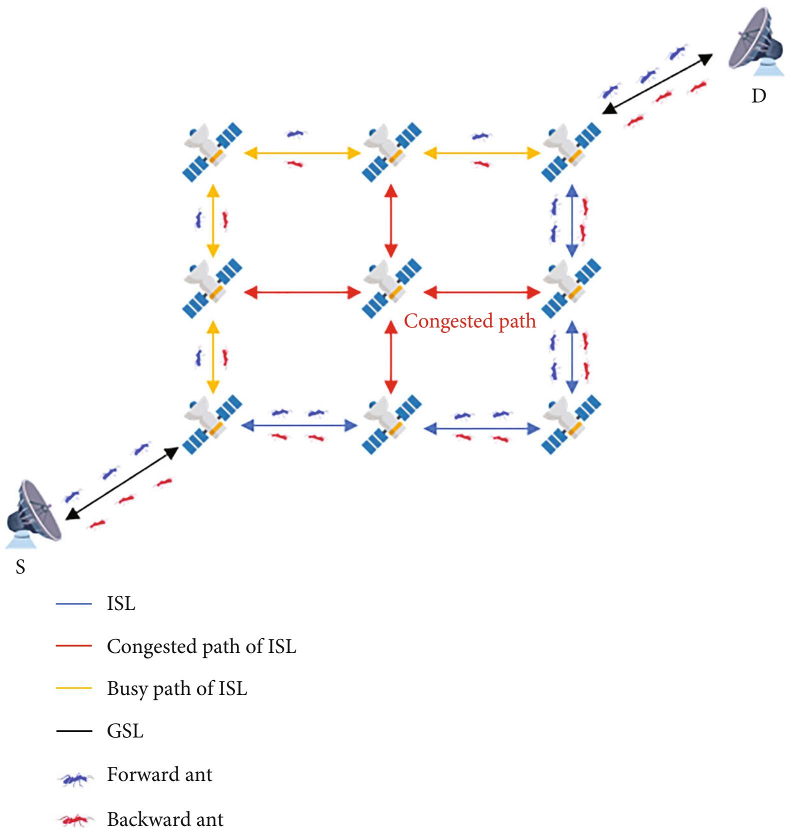


Figure 3: Ant colony algorithm in LEO satellite networks.

path and connection information along the way as the pheromone, so that the data passing through the satellite node can select the next hop node according to such pheromone. When it finally reaches the destination node, the forward ant dies and a backward ant is then generated. In addition, each forward ant also maintains a routing table independently to record the traversed satellite nodes, as to avoid loops in the forwarding process.

The state transition rule is the core rule of all ant colony algorithms; each ant selects the next node according to the state transition rule. Through the state transition rules, the algorithm can use the network link information to find the optimal path from the source node to the destination node.

When the forward ant is at satellite node i, the probability that the adjacent node j of node i is selected as the next hop is

$$P_{ijf} = \begin{cases} \dfrac{\left[p_{ijf}(t)\right]^{\alpha}\left[\eta_{ijf}(t)\right]^{\beta}}{\sum_{k\in K}\left[p_{ikf}(t)\right]^{\alpha}\left[\eta_{ikf}(t)\right]^{\beta}} & q < q_0 \text{and LinkAvailable}(i, j) = 1, \\ \dfrac{1}{A} & q \geq q_0 \text{ and LinkAvailable}(i, j) = 1, \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $p_{ijf}(t)$ and $\eta_{ijf}(t)$ are the pheromone density and congestion-avoiding heuristic information of link (i, j) with data flow f at time t, respectively, and α and β are the control parameters. q is a random number between 0 and 1. The value of q_0 determines the weights of exploring new paths and using prior knowledge. A is the number of candidate nodes for the next hop, and K is the set of adjacent nodes. If the link is available, $\text{LinkAvailable}(i, j) = 1$ and $q < q_0$, we calculate the probability according to the congestion-avoiding heuristic information and pheromone. If the link is available and $q \geq q_0$, we randomly select the next hop. If the link is unavailable, the probability value is set to 0, and the link is blocked.

The influence factors α and β represent the weights of the pheromone and congestion-avoiding heuristic information, respectively, where $\alpha + \beta = 1$. In order to accelerate the convergence speed in the initial stage and avoid rapid local convergence, we design the influence factor α at moment t as

$$\alpha(t) = \begin{cases} e^{-\sigma t} & \dfrac{\text{Num}_{\Delta v} \bullet S_{\text{avg}}}{\Delta v} > B \\ \varepsilon\left(1 + e^{-\sigma t}\right) & \text{otherwise} \end{cases}, \tag{10}$$

where ε and σ are constants and $\varepsilon, \sigma \epsilon (0, 1]$. t is the searching times. The larger the value of t is, the smaller the value of $\alpha(t)$ is. The weight of the pheromones decreases with the increase of the search times, while the congestion-avoiding heuristic information value β increases. $\text{Num}_{\Delta v}$ indicates the number of packets reaching the current node i in the past Δv time,

S_{avg} is the average packet size in the queue, and $\text{Num}_{\Delta\nu}\bullet S_{\text{avg}}/\Delta\nu$ indicates the bandwidth required for successful forwarding of $\text{Num}_{\Delta\nu}$ packets per second.

When the intersatellite link bandwidth B cannot support transmitting the data in time, we increase the weight of congestion-avoiding heuristic information to forward packets with idle ISLs to avoid adjacent links entering the congestion state quickly. When the congestion pressure of this node is light, we mainly make routing choices based on pheromone density to dynamically find the global optimal solution.

4.1.2. Congestion-Avoiding Heuristic Information. The pheromone only considers the transmission quality of the whole path from the source to the destination node but fails to handle local conditions in time. For example, after the ant colony algorithm runs for a period, there will be a stagnation phenomenon, i.e., all the individuals find the same solution and the search space converge, and no better solutions can be found. Stagnation may cause local node congestion. To tackle this problem, we propose the congestion-avoiding heuristic information, which estimates the busy degree of the link by considering the cache occupancy of the link. As the length of ISL changes from time to time, we also consider the length of the communication link. We design the congestion-avoiding heuristic information $\eta_{ijf}(t)$ of link (i, j) at moment t as

$$\eta_{ijf}(t) = \frac{1}{\mu d(i,j)_t + (1-\mu)\varrho_{ij}(t)}, \tag{11}$$

where $d(i,j)_t$ is the length of link (i,j) at moment t, $\varrho_{ij}(t)$ represents the waiting queue occupancy rate of link (i,j) at time t, and μ is the weight of the link length. If the length of link (i,j) is shorter or the cache occupancy is lower, the congestion-avoiding heuristic information value is greater. $\varrho_{ij}(t)$ is calculated as

$$\varrho_{ij}(t) = \frac{\delta_{ij}(t)}{Q}, \tag{12}$$

where Q is capacity of the buffer. $\delta_{ij}(t)$ is the space occupied by packets in the cache queue of current link (i,j) at time t.

4.1.3. Backward Ants. The backward ant returns to the source node from the destination according to the path recorded by the forward ant and updates the pheromone value of each node along the way according to the link information collected by the forward ant. With the increase of the number of backward ants passing through, the more pheromones accumulated, and the forward ants tend to choose the path with higher pheromone density. However, if the number of ant colonies is large and each individual ant starts to find the path from the source to the destination, the pheromone difference on each path will be small, and it is possible to obtain the stable solution after a certain number of iterations. However, an ant colony system algorithm is prone to stagnation, i.e., the stable solution is not the global optimal solution, and ants lose the ability to explore new paths. To solve these problems, we optimize the pheromone update rules to tackle the problems of slow convergence and untimely information update. We use p to represent the density of pheromones and design the pheromone update as

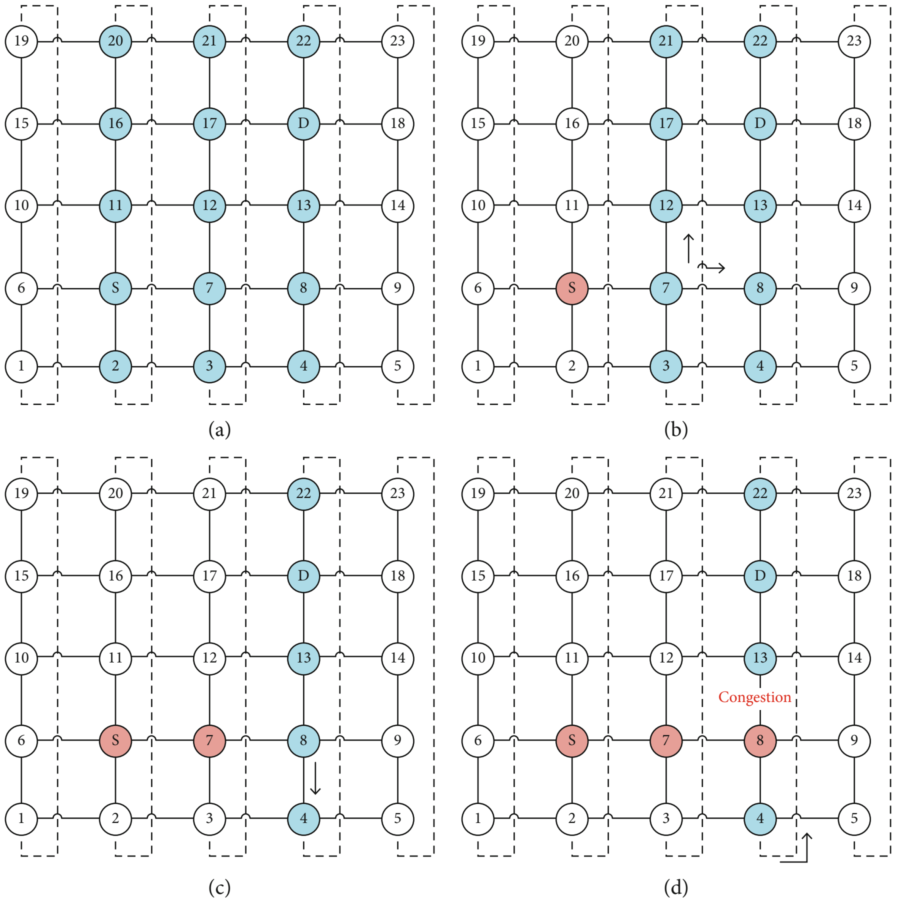
$$p_{ijf}(t+1) = \begin{cases} (1-\rho)p_{ijf}(t) + \sum_{k=1}^{m} \Delta\tau_{ijf}^{k}(t) & p_{ijf}(t+1) < \Gamma \\ \Gamma & \text{otherwise} \end{cases}, \tag{13}$$

where m is the number of ants at each iteration and $\rho \in (0, 1)$ is the pheromone evaporation coefficient. The main purpose of pheromone evaporation is to avoid stagnation. Each ant will experience pheromone update, and the best ant will place more pheromones on the node with the best solution. It will lead ants to explore the optimal solution in subsequent iterations. In order to prevent the infinite increment of pheromone, let Γ represent the upper limit of the pheromone density of link (i, j). The pheromone increment $\Delta\tau$ is related to the total cost of the current path and the congestion of the current link, and $\Delta\tau$ is calculated as

$$\Delta\tau_{ijf}^{k}(t) = \begin{cases} \varphi \dfrac{(1-\varrho_{ij}(t))}{\text{Cost}_{sdf}} & \text{Cost}_{sdf} \le \text{Cost}D_f \\ 0 & \text{otherwise} \end{cases}, \tag{14}$$

where Cost_{sdf} is the total cost of the data flow f from the source node s to the destination d and $\text{Cost}D_f$ is the maximum tolerated cost. If the path length explored by the forward ant exceeds $\text{Cost}D_f$, there will be a high delay in transferring data on this path, so we set the pheromone increment $\Delta\tau$ to 0. ϱ represents an immediate quantitative value related to the queue waiting time, and the pheromone increment $\Delta\tau$ is negatively correlated with the value of ϱ. We use φ to correct pheromone increment $\Delta\tau$, i.e., the relative relationship between the degree of link congestion and the cost of link.

4.2. Window Reduction. Ant colony algorithms use forward ants to explore new paths heuristically. A greater number of forward ants can find more available paths but also increase the transmission overhead. To reduce such overhead, we thus propose a window reduction mechanism to limit the routes to a specific range. It specifies a rectangular range according to the current satellite and the logical address of the destination satellite. With the movement of the forward ant, the size of the restricted area, referred to as the "window," is shrinking, and this process resembles reducing the size of the window. Each satellite node has a unique number, and the logical address can be calculated using this number. If there are m orbits in the constellation system and n satellites in each orbit, we use $<i, j>$ to represent the j-th satellite in the i-th orbit and $N_{i,j}$ to represent the logical address of the satellite, where i and j are calculated as



(a) (b) (c) (d)

Figure 4: Example of path selection for window reduction mechanism.

$$\begin{cases} i = \dfrac{N_{i,j}}{n} \\ j = N_{i,j} \bmod n \end{cases}. \qquad (15)$$

In satellite networks, the four adjacent satellites of satellite $<i_c, j_c>$ are $<(i_c + 1) \bmod \mathrm{m}, j_c>$, $<(i_c - 1) \bmod \mathrm{m}, j_c>$, $<(i_c, (j_c + 1) \bmod \mathrm{n}>$, and $<(i_c, (j_c - 1) \bmod \mathrm{n}>$. $<i_e, j_e>$ is the destination when packets are forwarded to the destination node:

$$\begin{cases} i_c = i_e, \\ j_c = j_e. \end{cases} \qquad (16)$$

Then, $<i_e, j_e>$ sends the data to the ground destination node directly.

If the current satellite node is not the destination satellite, we define $\mathrm{Direction}_X$ as the x-axis relative distance between the specified node and the destination satellite. The x-axis relative distance is calculated using the orbit number of the logical address as

$$\mathrm{Direction}_X(i_e - i) = \begin{cases} m - |i_e - i| |i_e - i| > \dfrac{m}{2} \\ |i_e - i| \text{ otherwise} \end{cases}. \qquad (17)$$

In the data forwarding process, the window reduction mechanism requires that the next-hop satellite node K must meet the following conditions:

$$\mathrm{Direction}_X(i_e - i_k) \le \mathrm{Direction}_X(i_e - i_c). \qquad (18)$$

The orbits of the current satellite and the destination satellite divide the satellite networks into two ranges. Equations (17) and (18) mean that the search area of an ant colony is restricted to the smaller one between two ranges. If $\mathrm{Direction}_X(i_e - i) = 0$, it means that the current satellite and the destination satellite are in one orbit, and the packet will only be forwarded on that orbit.

The purpose of the window reduction mechanism is to limit the routing to the range formed by the current node and the destination satellite. As shown in Figure 4(a), a group of data is forwarded from satellite S to satellite D. The window reduction mechanism allows ants to be forwarded to all blue nodes. The adjacent satellites of S are 2, 6, 7, and 11, and satellite nodes 2, 7, and 11 meet the conditions of the window reduction mechanism, so nodes 2, 7, and 11 are added to the candidate set, K. When a packet is about to be forwarded, only the nodes in set K can be selected. Similarly, as shown in Figure 4(b), when the data reaches satellite 7, satellites 3, 8, and 12 are added to set K according to the above rules. Figure 4(c) shows that when

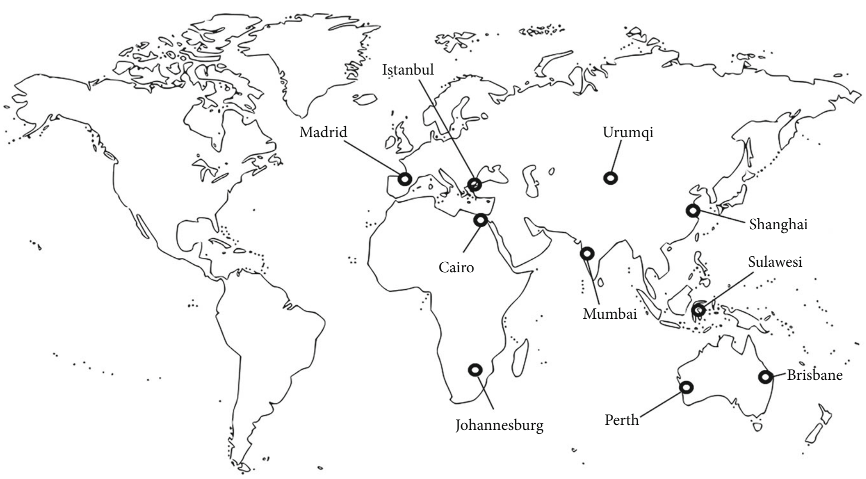


Figure 5: Distribution of ground destination cities.

the data reaches satellite 8, a satellite from satellites 4 and 13 will be selected as the next hop since satellite 8 is in the same orbit as the destination satellite D. Passing satellite 13 will make it faster to reach destination satellite D, so more ants will choose node 13 to forward. With the increase of packets passing through link $(8, 13)$, according to Equation (7), when $\varrho_{ij}(t) > C$, link$(8, 13)$ is not available. Satellite 4 is selected to forward the data. As shown in Figure 4(d), then the data of satellite 4 is forwarded to satellite 22 and finally to the destination satellite. The window reduction mechanism uses the x-axis coordinates of the satellite nodes to limit the range of forward ants. The size of this range is determined by the x-axis coordinates of the current and destination satellite nodes. We combine the optimized ant colony algorithm with the window reduction mechanism above to reduce the additional overhead caused by ants exploring new paths.

ACORA-WR is a distributed algorithm in which each satellite node updates its routing table independently according to Equation (13). The computational complexity at node is proportional to the number of neighbors. For example, in Iridium, there are no more than 4 satellite nodes adjacent to each satellite. In Starlink, there are about 4 adjacent satellite nodes, and in other LEO networks, there are usually no more than 8 neighbors. In addition, ground nodes are not counted in ACORA-WR. If the ground node is the destination, the satellite will forward the data to it directly. Therefore, the number of the neighbors of each node is a constant, so the computational complexity is O(1). Each satellite node stores only the pheromones, cache occupancy, and distance of its neighbors, so the storage complexity is O(1), too.

5. Results and Discussion

5.1. Simulation Setup. We use the NS2 simulation platform to simulate an Iridium-like system and evaluate the performance of our proposed ACORA-WR routing algorithm. The Iridium satellite network consists of 66 satellites with 11 satellites in each orbit. Each satellite has four intersatellite ISLs, including intraplane ISLs and interplane ISLs. We set the bandwidth of each ISL to 2 Mbps. The uplink and downlink bandwidth of the satellite is 2 Mbps. Other parameters of Iridium networks are as follows: altitude = 780 km and inclination = 86.4°. We use constant traffic with a fixed packet size of 512 B. The simulation time is 100 s. There are seven flows distributed among the cities shown in Figure 5, which are distributed between 50 degrees north and 50 degrees south latitudes. The flows, i.e., source-destination pairs, are shown in Table 2. We compare our ACORA-WR with three popular routing protocols, LBRA-CP [18], SPR, and LCRA [30].

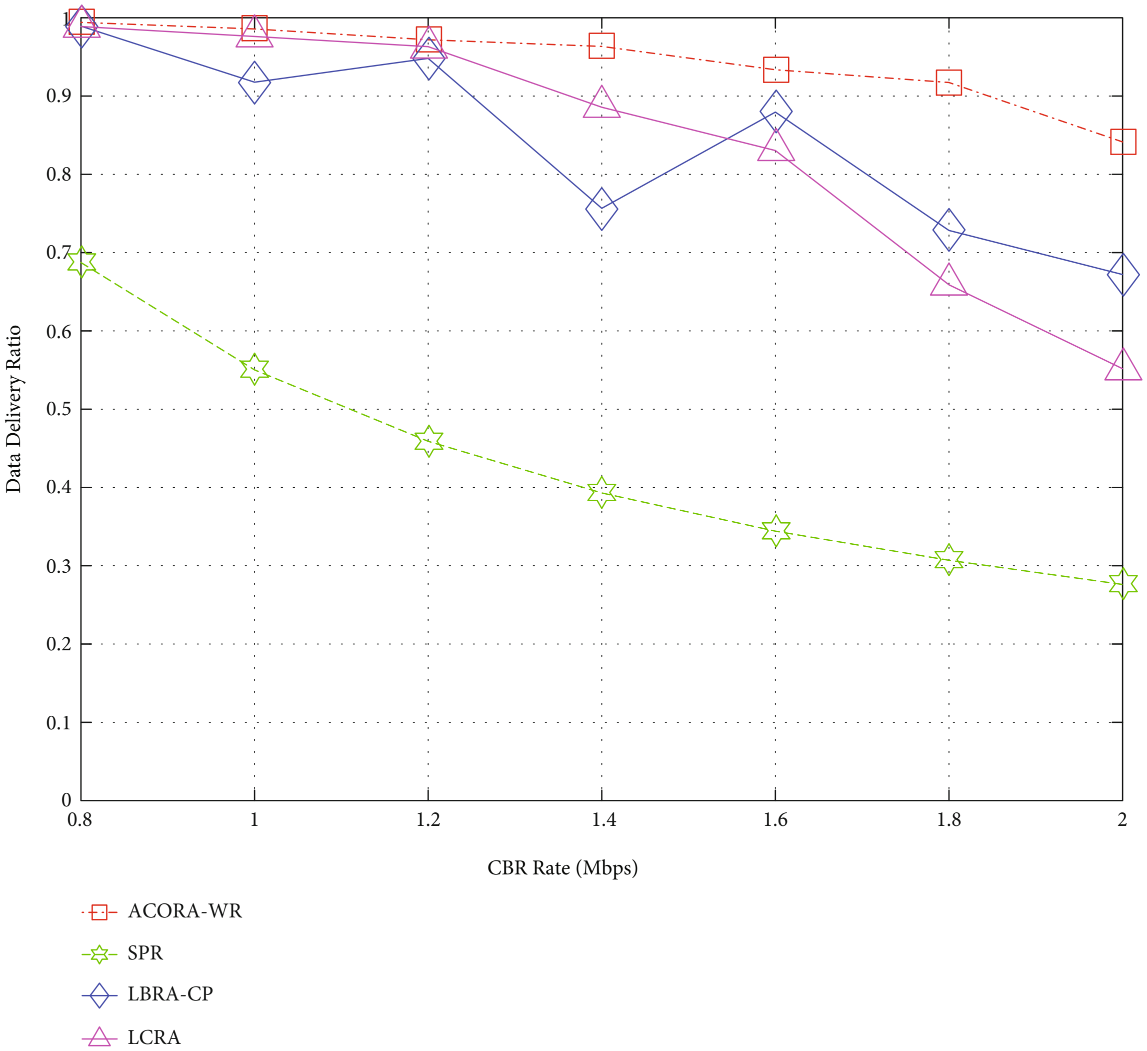
Table 2: Sending and receiving cities of simulated data flow.

Source	Destination
Madrid	Brisbane
Istanbul	Sulawesi
Mumbai	Perth
Urumqi	Johannesburg
Shanghai	Cairo
Johannesburg	Shanghai
Perth	Istanbul

(1) LBRA-CP is a load-balanced routing algorithm based on congestion prediction for LEO satellite networks. LBRA-CP sets up hot spot areas to avoid congestion by transferring traffic from hot spot areas to nonhot spot areas

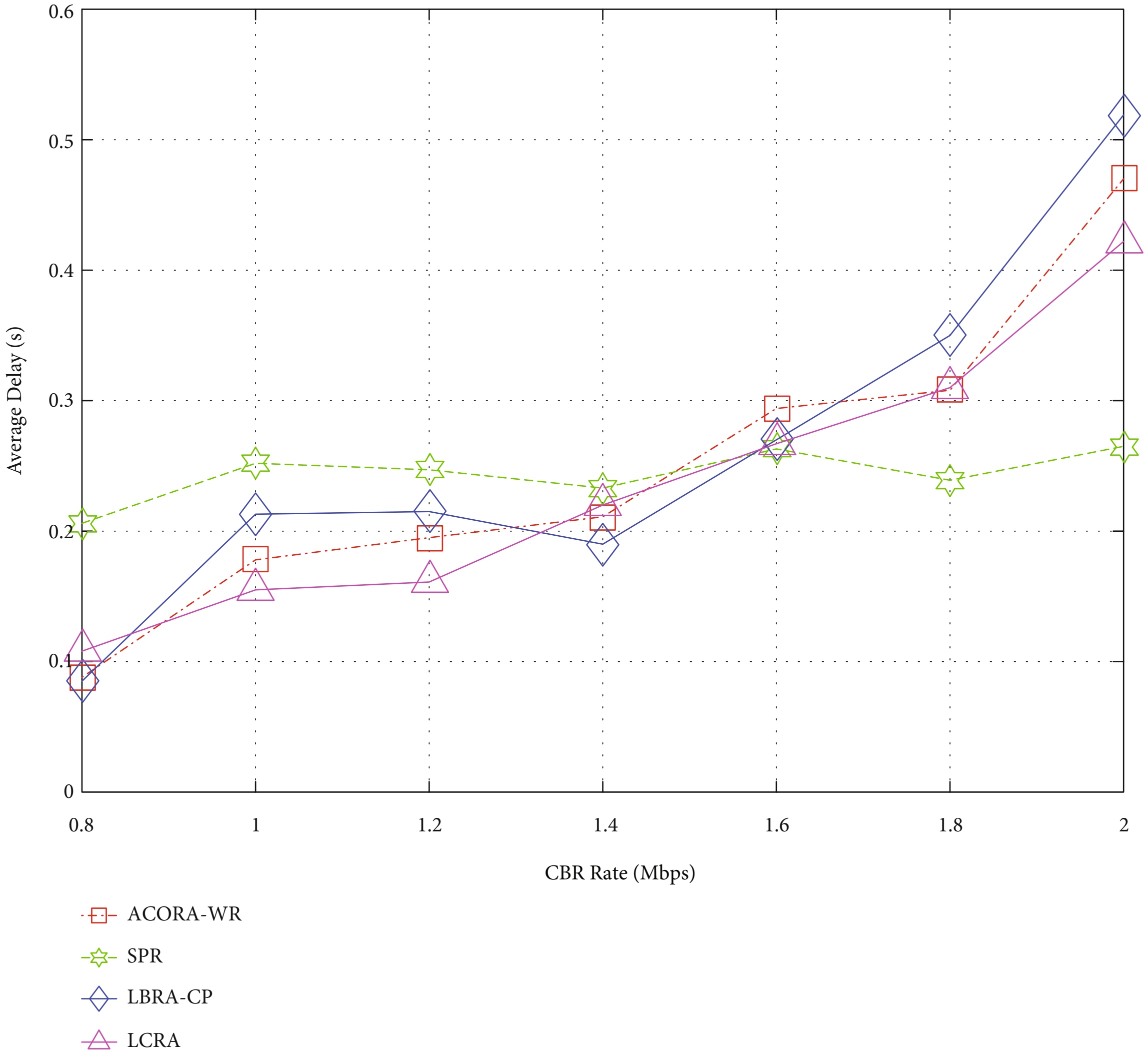
(2) SPR is the shortest path routing protocol based on Dijkstra

(3) LCRA is the cost-balanced routing protocol in LEO satellite networks. It selects the shortest path to the destination and uses the congestion information of neighboring nodes to alleviate the network congestion. It has low computational complexity and can effectively reduce network overhead



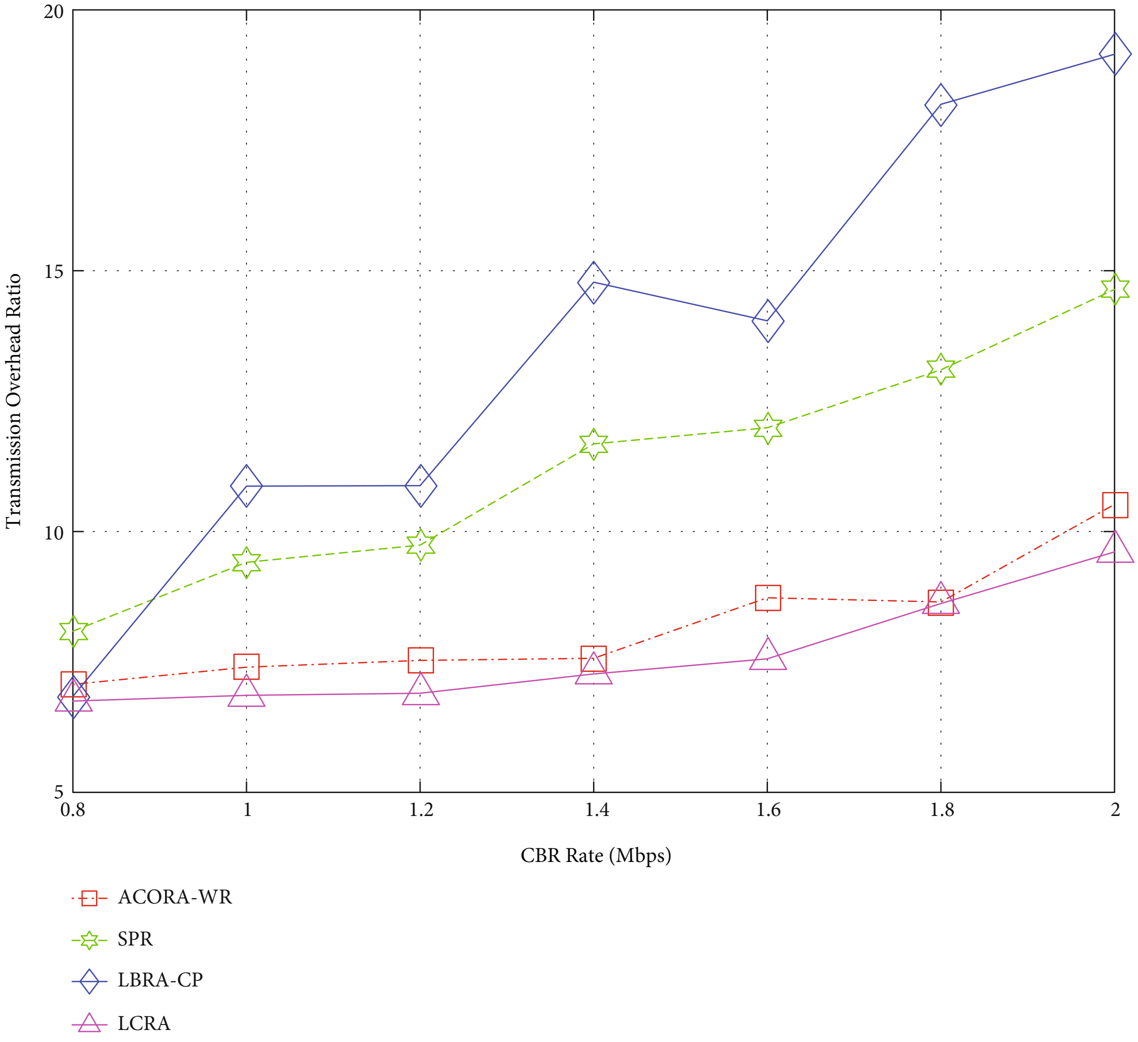
(a) Data delivery ratio

Figure 6: Continued.



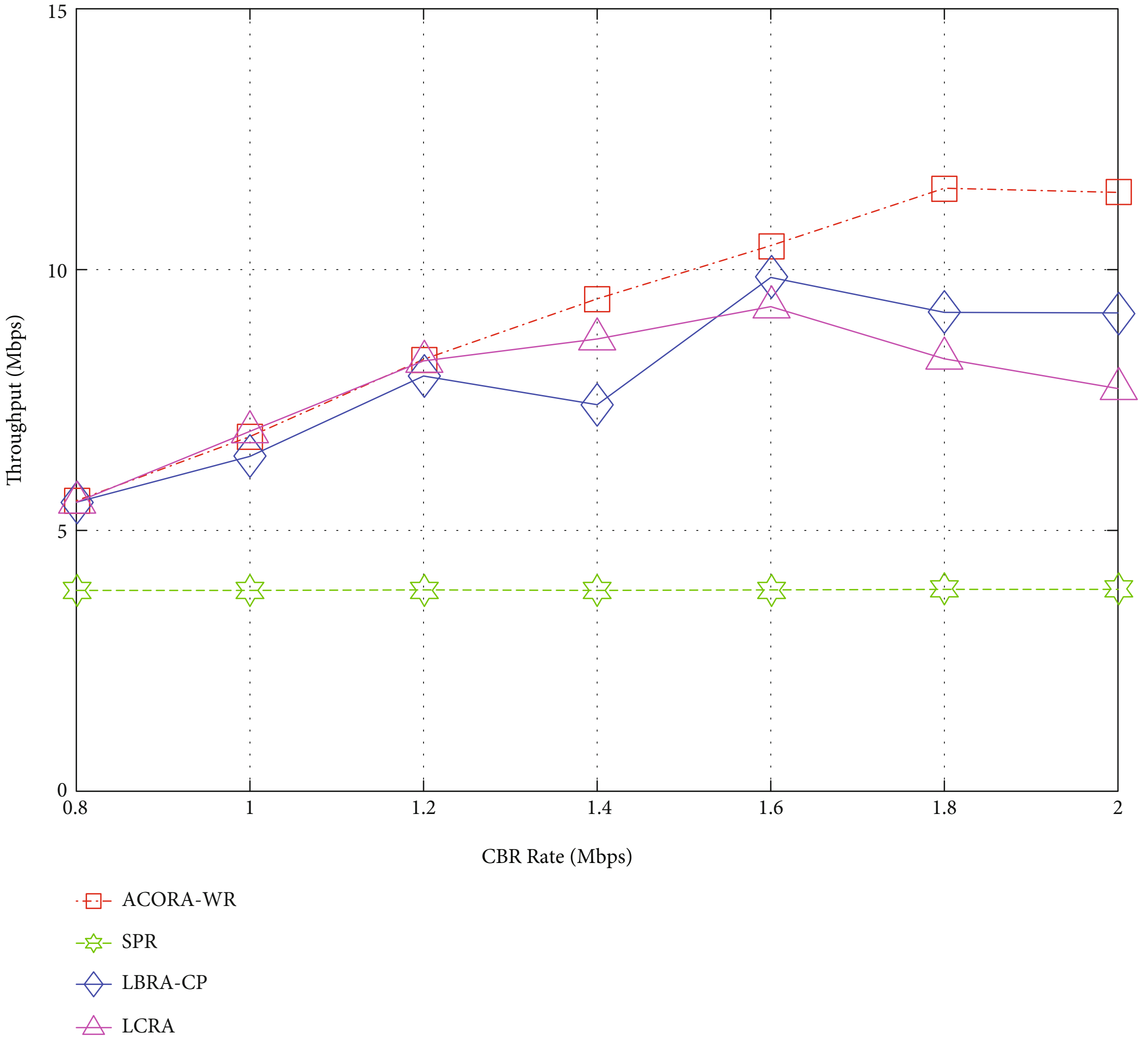
(b) Average delay

Figure 6: Continued.



(c) Transmission overhead ratio

Figure 6: Continued.



(d) Throughput

Figure 6: The performance of ACORA-WR with varying CBR rate.

The performance is evaluated using following metrics:

(1) *Data Delivery Ratio*. The ratio of the number of messages arrived at destinations to the number of messages expected to arrive at destinations

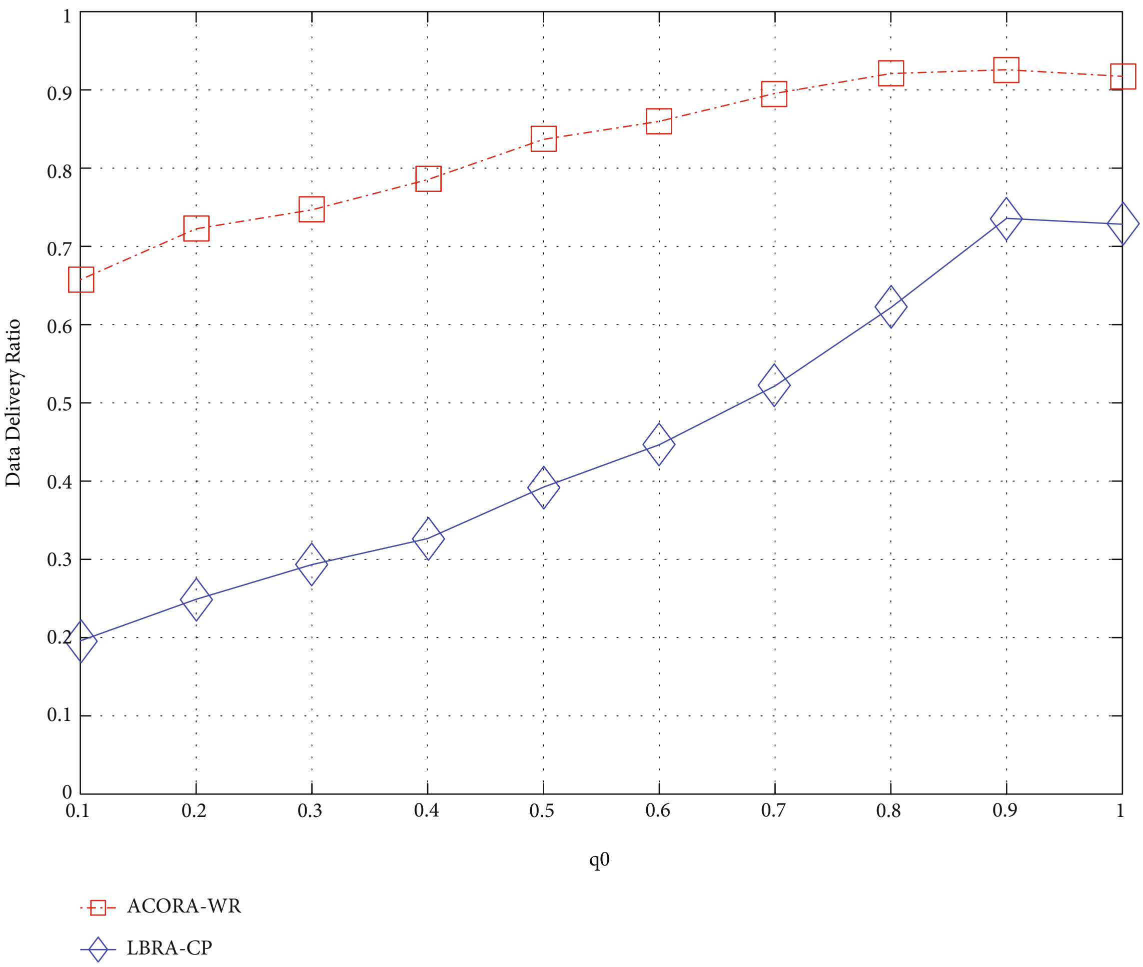
(2) *Average Delay*. The average time spent by all messages from the sources to the destinations

(3) *Transmission Overhead Ratio*. The total number of relayed messages divided by the number of messages arriving at the destination

(4) *Throughput*. The total successful message delivery rates on the destinations

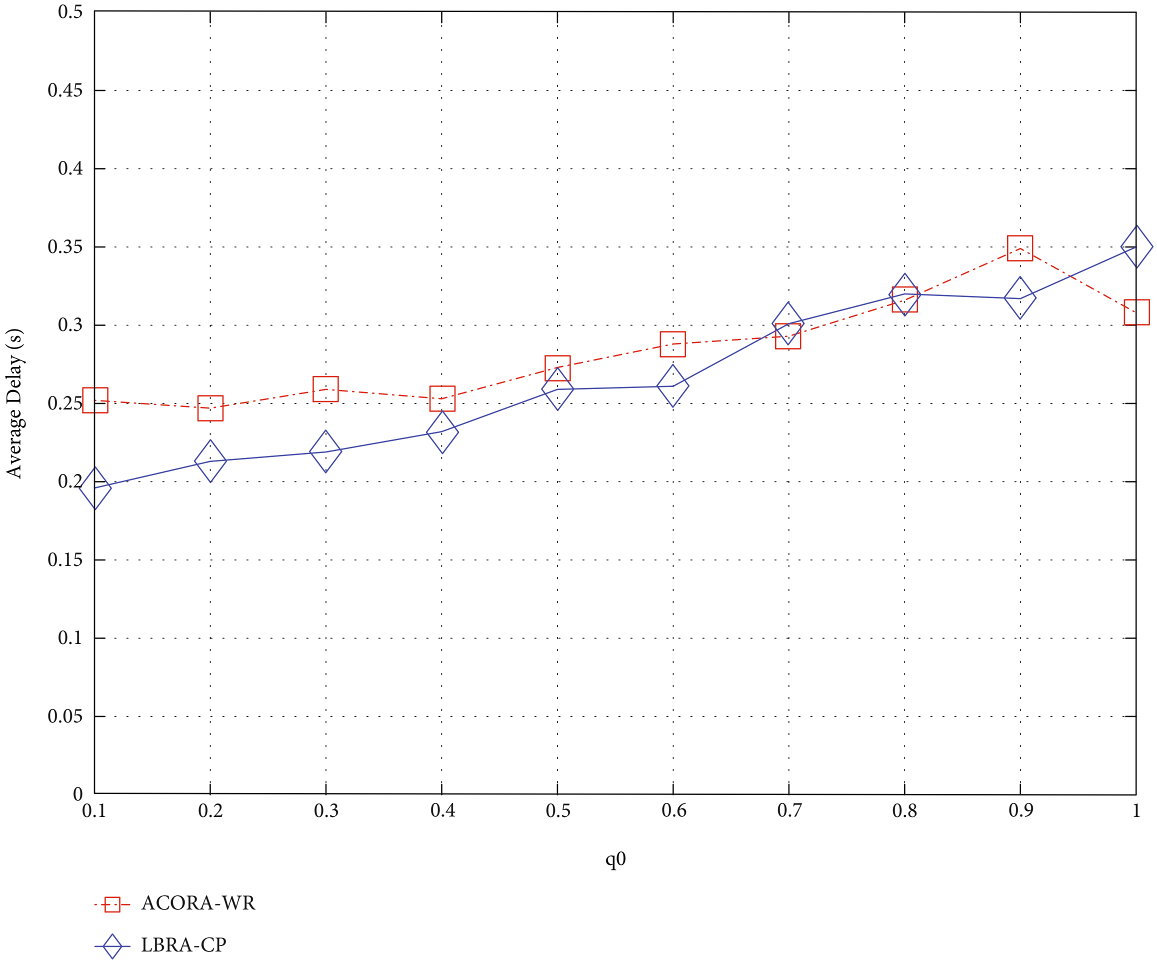
5.2. The Performance of ACORA-WR with Varying CBR Rate. To evaluate the effects of the transmission rate of CBR (constant bit rate) on ACORA-WR performance, we increase the total CBR generation rate in the networks from 0.8 Mbps to 2 Mbps. The buffer size of the node is 50 packets. The total simulation time is 100 s. For comparison fairness, we set the q_0 of ACORA-WR and LBRA-CP to be the same.

In Figure 6(a), as the CBR rate increases, the data delivery ratio in all protocols decreases due to the heavy transmission load. ACORA-WR shows the highest data delivery ratio and traffic balancing capability in the range of 0.8 Mbps to 2 Mbps of CBR rate. The shortest path algorithm SPR has the worst performance, and the average data delivery ratio of ACORA-WR is nearly twice as high as that of SPR. The LCRA algorithm considers the congestion information of adjacent nodes and chooses the shortest congestion path to the destination. However, LCRA only considers the congestion situation of adjacent nodes in routing, so it cannot optimize the routes from the global perspective. Due to the limitation of the transmission range, LCRA cannot provide enough paths to dynamically adapt to the traffic load. In contrast, ACORA-WR considers both global and local aspects through the ant colony algorithm. Compared with LCRA, the average data delivery ratio of ACORA-WR



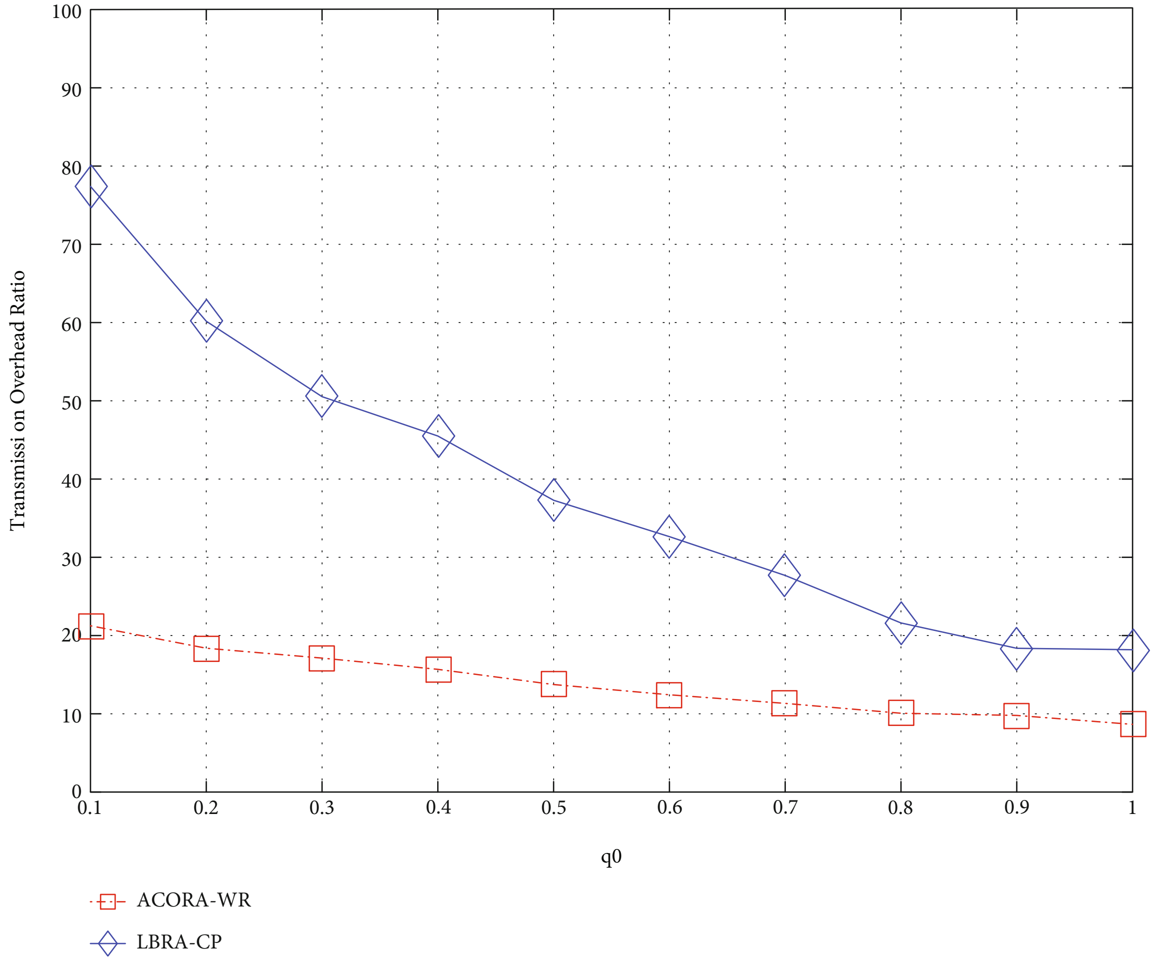
(a) Data delivery ratio

Figure 7: Continued.



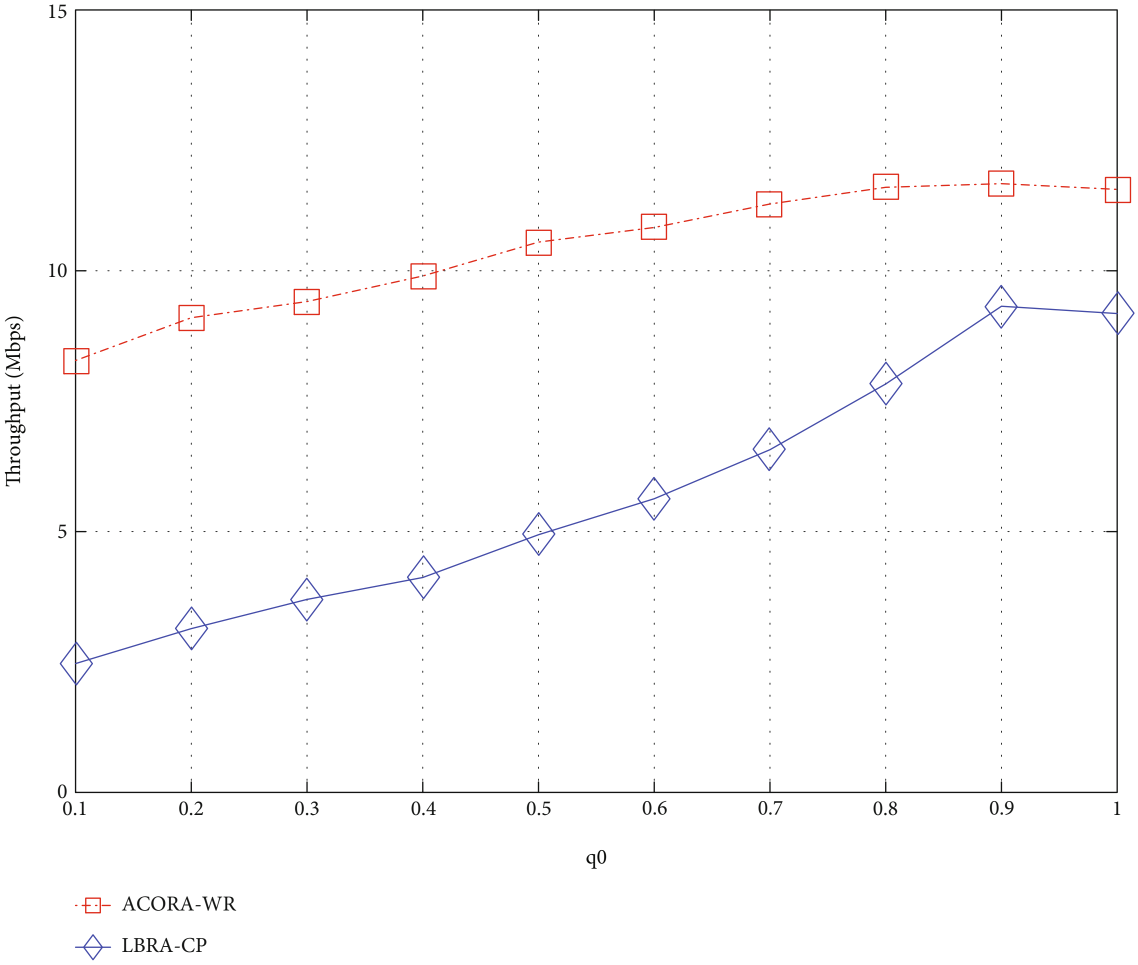
(b) Average delay

Figure 7: Continued.



(c) Transmission overhead ratio

FIGURE 7: Continued.



(d) Throughput

Figure 7: The performance of ACORA-WR with varying q_0.

increases by about 10%, and the LCRA performance decreases significantly when the CBR rate is at 2 Mbps. The data delivery ratio of ACORA-WR is nearly 50% higher than LCRA when the CBR rate is at 2 Mbps. LBRA-CP is also based on ant colony algorithms and combines congestion prediction to find the best route; however, its overhead is higher. ACORA-WR can speed up ant colony search and modify routing table in time. The experimental results show that the data delivery ratio of ACORA-WR is about 10% higher than LBRA-CP.

From Figure 6(b), we can see that the performance of our ACORA-WR is between LCRA and LBRA-CP in terms of the average delay. LCRA has the best average delay because the length of the path chosen by LCRA is generally smaller than ACORA-WR and LBRA-CP. However, a large amount of data is discarded while waiting in queues, and the percentage of packets discarded increases with the increase of the CBR rate. Similarly, SPR has a good average delay, but they always choose the shortest path. When the load is heavy, many packets will be discarded. LBRA-CP and ACORA-WR require higher average delay to ensure data delivery. The results show that the average delay of ACORA-WR is better than that of LBRA-CP in the range of 0.8 Mbps to 2 Mbps CBR rate.

In terms of transmission overhead ratio, as shown in Figure 6(c), LCRA has the lowest transmission overhead ratio, and ACORA-WR is slightly higher than LCRA because both LCRA and ACORA-WR limit the range of data transmission, but ACORA-WR can choose more flexible paths than LCRA. However, compared with LBRA-CP, the average transmission overhead of ACORA-WR is about 40% lower than that of LBRA-CP. This is because LBRA-CP has a broader ant search range and a slower ant colony convergence rate than ACORA-WR. Some data choose longer and longer paths, which results in a higher transmission overhead ratio.

In Figure 6(d), throughput of all protocols increases with the increase of CBR rate. ACORA-WR always maintains the highest throughput, SPR has the lowest throughput, LCRA has slightly higher throughput than LBRA-CP when CBR rate is 0.8 Mbps to 1.4 Mbps, and LBRA-CP overwhelms LCRA when CBR rate reaches 1.6 Mbps. When CBR rates range from 1.8 Mbps to 2 Mbps, the throughput of ACORA-WR increases by about 25% compared with LBRA-CP and 40% compared with LCRA.

In summary, the experimental results show that ACORA-WR can maintain a high data delivery ratio and network throughput. It can balance the traffic under heavy network load while maintaining a small average delay and overhead.

5.3. The Performance of ACORA-WR with Varying the Value of q_0. The value of q_0 determines the relative importance of exploring new paths and using prior knowledge. According to Equation (7), the probability values are calculated using congestion-avoiding heuristic information and pheromone when $q < q_0$. Routing forwarding is chosen randomly when the link is available and $q \geq q_0$. q is a random number, q_0 is a fixed value, and q_0 determines the probability ratio of the two rules to choose a route. The larger the value of q_0 is, the smaller the probability of forward ants will explore randomly, and the lower the ability of finding a new route. The smaller the value of q_0 is, the slower the convergence speed of the ant colony algorithm will have, but it will result in higher network overhead and delay. To evaluate the impact of the value of q_0 on the performance of two kinds of ant colony algorithm ACORA-WR and LBRA-CP, we change the value of q_0 in the networks from 0.1 to 1 and set the CBR generation rate to 1.8 Mbps. The buffer size of the node is 50 packets.

In Figure 7(a), both ACORA-WR and LBRA-CP have significantly improved data delivery rates as the value of q_0 increases, and both ACORA-WR and LBRA-CP show the best performance when the value of q_0 is 0.9. When the value of q_0 is low, most ants in ACORA-WR and LBRA-CP randomly select routes in the networks, but due to the narrowing of the window, ACORA-WR data is easier to find destinations, so even when the value of q_0 is low, it still shows a significantly better data delivery rate. When the value of q_0 is in the range of 0.1 to 0.8, the data delivery rate of ACORA-WR is about 90% higher than that of LBRA-CP.

In Figure 7(b), with the increase of q_0, the average delay of ACORA-WR differs slightly from that of LBRA-CP. This is because we take into account the delay of the entire path from the sources to the destinations of the data flows in the pheromone overlay process and adjust the traffic allocation of the link locally with congestion-avoiding heuristic information to ensure a lower average delay while increasing the data delivery rate.

In Figure 7(c), LBRA-CP has a very high transmission overhead ratio when the value of q_0 is small, because most of the data is searched randomly and unrestrictedly, resulting in longer paths being selected and higher transmission overhead ratio. With the constraint of the window reduction mechanism, even if most ants randomly select routes, they can still successfully forward data to their destinations with a low transmission overhead ratio.

Figure 7(d) shows the changes in network throughput of ACORA-WR and LBRA-CP as the value of q_0 increases. Overall, the network throughput of ACORA-WR and LBRA-CP increases with the value of q_0. The networks throughput of ACORA-WR is significantly better than that of LBRA-CP when the value of q_0 ranges from 0.1 to 0.8. When q_0 is 0.9, both algorithms have the best performance. With faster convergence, LBRA-CP throughput tends to approach ACORA-WR gradually, but overall, ACORA-WR is still better than LBRA-CP.

6. Conclusions and Future Work

In this paper, we aimed at balancing traffic load in LEO satellite networks and proposed an improved ant colony routing algorithm with window reduction. When packets are forwarded between intersatellite links, congestion-avoiding heuristic information is added to the state transition rules, and forward ants detect the route through our improved state transition rules periodically, as to tackle the problem that ant colony algorithms fail to adjust the local information in time. Moreover, we proposed a window reduction mechanism to limit the transmission direction of data and remove redundant paths, which accelerates the convergence speed of ant colony and reduce network transmission overhead. Finally, the new routing algorithm is implemented on the network simulation platform NS2. The experimental results show that compared with other distributed algorithms in LEO satellite networks, the proposed ACORA-WR scheme demonstrates higher data delivery ratio and network throughput. In conclusion, ACORA-WR can effectively balance the network traffic load and improve the data delivery ratio of satellite networks.

In our future work, we will consider energy consumption in load balancing algorithms. Satellites mainly receive energy from the sun, which is not guaranteed anywhere as they move. Load balancing for LEO satellite networks under these cases requires further study.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (Grant Nos. 61702127 and 62062006), the Science and Technology Program of Guangzhou (Grant No. 201804010461), the China Scholarship Council (Grant Nos. 201908440064 and 201908440085), the Hundred Young Talents Plan Project of Guangdong University of Technology (Grant No. 220413618), and the Guangxi Innovation-driven Development Major Project Platform (No. Guike AA20302002).

References

[1] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.

[2] N. N. Dao, Q. V. Pham, N. H. Tu et al., "Survey on aerial radio access networks: toward a comprehensive 6G access infrastructure," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1193–1225, 2021.

[3] B. Shang, Y. Yi, and L. Liu, "Computing over space-air-ground integrated networks: challenges and opportunities," *IEEE Network*, vol. 35, no. 4, pp. 302–309, 2021.

[4] Q. Wang, X. Li, Y. Liu, L. T. Alex, S. A. Khowaja, and V. G. Menon, "UAV-enabled non-orthogonal multiple access networks for ground-air-ground communications," *IEEE Transactions on Green Communications and Networking*, p. 1, 2022.

[5] X. Deng, J. Shao, L. Chang, and J. Liang, "A blockchain-based authentication protocol using cryptocurrency technology in LEO satellite networks," *Electronics*, vol. 10, no. 24, p. 3151, 2021.

[6] X. Li, Y. Zheng, W. U. Khan et al., "Physical layer security of cognitive ambient backscatter communications for green Internet-of-Things," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1066–1076, 2021.

[7] Z. Xie, Z. Chu, V. G. Menon, S. Mumtaz, and J. Zhang, "Exploiting benefits of IRS in wireless powered NOMA networks," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 1, pp. 175–186, 2022.

[8] G. Li, H. Liu, G. Huang, X. Li, B. Raj, and F. Kara, "Effective capacity analysis of reconfigurable intelligent surfaces aided NOMA network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, 16 pages, 2021.

[9] X. Li, F. Tang, L. Chen, and J. Li, "A state-aware and load-balanced routing model for LEO satellite networks," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, Singapore, Dec. 2017.

[10] Z. Liu, J. Li, Y. Wang, X. Li, and S. Chen, "HGL: a hybrid global-local load balancing routing scheme for the Internet of Things through satellite networks," *International Journal of Distributed Sensor Networks*, vol. 13, 16 pages, 2017.

[11] E. Wang, H. Li, and S. Zhang, "Load balancing based on cache resource allocation in satellite networks," *IEEE Access*, vol. 7, pp. 56864–56879, 2019.

[12] Z. Na, Z. Pan, X. Liu, Z. Deng, Z. Gao, and Q. Guo, "Distributed routing strategy based on machine learning for LEO satellite network," *Wireless Communications and Mobile Computing*, vol. 2018, 10 pages, 2018.

[13] T. Taleb, D. Mashimo, A. Jamalipour, N. Kato, and Y. Nemoto, "Explicit load balancing technique for NGEO satellite IP networks with on-board processing capabilities," *IEEE/ACM Transactions on Networking*, vol. 17, no. 1, pp. 281–293, 2009.

[14] G. Song, M. Chao, B. Yang, and Y. Zheng, "TLR: a traffic-light-based intelligent routing strategy for NGEO satellite IP networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 6, pp. 3380–3393, 2014.

[15] E. Papapetrou, S. Karapantazis, and F. N. Pavlidou, "Distributed on-demand routing for LEO satellite systems," *Computer Networks*, vol. 51, no. 15, pp. 4356–4376, 2007.

[16] S. Karapantazis, E. Papapetrou, and F. N. Pavlidou, "Multiservice on-demand routing in LEO satellite networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 1, pp. 107–112, 2009.

[17] Y. Rao and R. C. Wang, "Agent-based load balancing routing for LEO satellite networks," *Computer Networks*, vol. 54, no. 17, pp. 3187–3195, 2010.

[18] H. Wang, G. Wen, N. Liu, J. Zhang, and Y. Tao, "A load balanced routing algorithm based on congestion prediction for LEO satellite networks," *Cluster Computing*, vol. 22, no. S4, pp. 8025–8033, 2019.

[19] F. Bokhari and G. Zaruba, "Interference-aware routing using ant colony optimization in wireless mesh networks," in *Wireless Communications and Networksing Conference*, Budapest, Hungary, April 2009.

[20] M. H. Eiza, T. Owens, Q. Ni, and Q. Shi, "Situation-aware QoS routing algorithm for vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5520–5535, 2015.

[21] X. Xiang, Y. Tian, X. Zhang, J. Xiao, and Y. Jin, "A pairwise proximity learning-based ant colony algorithm for dynamic vehicle routing problems," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP(99), pp. 1–12, 2021.

[22] S. Chatterjee and S. Das, "Ant colony optimization based enhanced dynamic source routing algorithm for mobile adhoc network," *Information Sciences.*, vol. 295, pp. 67–90, 2015.

[23] M. Rathee, S. Kumar, A. H. Gandomi, K. Dilip, B. Balusamy, and R. Patan, "Ant colony optimization based quality of service aware energy balancing secure routing algorithm for wireless sensor networks," *IEEE Transactions on Engineering Management*, vol. 68, no. 1, pp. 170–182, 2021.

[24] W. Ni, Z. Xu, J. Zou, Z. Wan, and X. Zhao, "Neural network optimal routing algorithm based on genetic ant colony in IPv6 environment," *Computational Intelligence and Neuroscience*, vol. 2021, 3115713 pages, 2021.

[25] S. Kilic and O. Ozkan, "Ant colony optimization approach for satellite broadcast scheduling problem," in *2017 8th International Conference on Recent Advances in Space Technologies*, Istanbul, Turkey, June 2017.

[26] H. S. Chang, B. W. Kim, C. G. Lee et al., "FSA-based link assignment and routing in low-earth orbit satellite networks," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 3, pp. 1037–1048, 1998.

[27] D. Bertsekas and R. Gallager, *Data networks*, Prentice-Hall, Inc, 1992.

[28] L. Bai Jianjun and X.,. L. Zexin, "Compact explicit multi-path routing for LEO satellite networks, in," in *Workshop on High Performance Switching and Routing*, pp. 386–390, Hong Kong, China, May 2005.

[29] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *Biosystems*, vol. 43, no. 2, pp. 73–81, 1997.

[30] X. Liu, X. Yan, Z. Jiang, C. Li, and Y. Yang, "A low-complexity routing algorithm based on load balancing for LEO satellite networks," in *2015 IEEE 82nd Vehicular Technology Conference*, Boston, MA, USA, Sept. 2015.