



## Article

# A Routing Strategy Based Genetic Algorithm Assisted by Ground Access Optimization for LEO Satellite Constellations

Peiying Zhang <sup>1</sup>, Chong Lv <sup>1</sup>, Guanjun Xu <sup>2,\*</sup>, Haoyu Wang <sup>3</sup> and Lizhuang Tan <sup>4,5,\*</sup>  
and Kostromitin Konstantin Igorevich <sup>6</sup>

- <sup>1</sup> Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China; zhangpeiying@upc.edu.cn (P.Z.); S22070061@s.upc.edu.cn (C.L.)
- <sup>2</sup> Space Information Research Institute, Hangzhou Dianzi University, Hangzhou 310018, China
- <sup>3</sup> School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; hywang2003@sjtu.edu.cn
- <sup>4</sup> Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250013, China
- <sup>5</sup> Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250013, China
- <sup>6</sup> Department of Physics of Nanoscale Systems, South Ural State University, 454080 Chelyabinsk, Russia; kostromitinki@susu.ru
- \* Correspondence: gjxu@ee.ecnu.edu.cn (G.X.); tanlzh@sdas.org (L.T.)

**Abstract:** Large-scale low Earth orbit satellite networks (LSNs) have been attracting increasing attention in recent years. These systems offer advantages such as low latency, high bandwidth communication, and all terrain coverage. However, the main challenges faced by LSNs is the calculation and maintenance of routing strategies. This is primarily due to the large scale and dynamic network topology of LSN constellations. As the number of satellites in constellations continues to rise, the feasibility of the centralized routing strategy, which calculates all shortest routes between every satellite, becomes increasingly limited by space and time constraints. This approach is also not suitable for the Walker Delta formation, which is becoming more popular for giant constellations. In order to find an effective routing strategy, this paper defines the satellite routing problem as a mixed linear integer programming problem (MILP), proposes a routing strategy based on a genetic algorithm (GA), and comprehensively considers the efficiency of source or destination ground stations to access satellite constellations. The routing strategy integrates ground station ingress and exit policies and inter-satellite packet forwarding policies and reduces the cost of routing decisions. The experimental results show that, compared with the traditional satellite routing algorithm, the proposed routing strategy has better link capacity utilization, a lower round trip communication time, and an improved traffic reception rate.

**Keywords:** LEO satellite network; routing strategy; multi-commodity flow; heuristic algorithm



**Citation:** Zhang, P.; Lv, C.; Xu, G.; Wang, H.; Tan, L.; Igorevich, K.K. A Routing Strategy Based Genetic Algorithm Assisted by Ground Access Optimization for LEO Satellite Constellations. *Electronics* **2023**, *12*, 4762. <https://doi.org/10.3390/electronics12234762>

Academic Editor: Athanasios D. Panagopoulos

Received: 15 September 2023

Revised: 20 November 2023

Accepted: 20 November 2023

Published: 24 November 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In recent years, 5G technology has led to the exploration of integrating ground networks and satellite communication networks [1]. Satellite networks can provide global coverage and high-speed transmission, which can improve the performance of ground networks. However, the integration of 5G and satellite networks also has many challenges, such as spectrum, architecture, and resource challenges. With the research of 6G, satellite networks have become more attractive, as 6G aims to achieve higher performance than 5G, such as 1 Tbps rates and 0.1 ms latency [2]. To meet these requirements, satellite networks need to be developed and improved in aspects such as number, orbit, antenna, frequency, and communication. Therefore, the integration of 5G and satellite networks could offer a

promising network architecture that can provide better communication and more scenarios for users. With the development of 6G, satellite networks will also experience new changes and opportunities, bringing new possibilities for the future communication industry [3].

There are three categories to classify satellite networks, which are based on the orbit altitude: low Earth orbit (LEO) satellite networks, medium Earth orbit (MEO) satellite networks, and geostationary Earth orbit (GEO) satellite networks [4]. Recently, LEO satellite constellations have emerged as a significant technology capable of enhancing communication. Satellite networks have the ability to compete with ground networks owing to the higher speed of inter-satellite links (ISL), which surpass the capabilities of optical fibers used in ground networks by reaching the speed of light in a vacuum [5]. In addition, operating at an altitude of 2000 km above the Earth's surface, LEO satellites can significantly reduce the round-trip time (RTT) and waiting time, contributing to their advantage of low latency. However, the dynamic nature of LEO constellations presents new challenges that need to be overcome.

The next-generation Internet, based on terrestrial networks and mega-satellite networks, is rapidly growing as an up-and-coming future network with the development of commercial satellites such as Starlink, OneWeb, and Telesat [6]. LSNs offer key advantages over terrestrial network solutions, including global network access, low deployment costs, and 24/7 availability. Moreover, LSNs provide lower communication latency and higher data rates compared to GEO and MEO satellite communication systems [7]. The advantages of LSNs make them a crucial infrastructure in the development of the next generation of space–air–ground integrated networks. Given the LSN's ability to support a range of applications, including telecommunication, remote sensing, location, and navigation, it is essential to enhance the performance of LSN facilities in response to their dynamic topology.

The issue of routing strategy and ground station geographic planning is inevitable [8] in LSNs; the former controls the transmission of traffic and the latter improves LSN performance by ensuring satellite coverage [9]. LSNs exhibit different behavior compared to terrestrial networks due to several factors. Firstly, the mobility of satellites causes frequent variations in the network topology. Secondly, the unstable relative position and severe space environment can potentially degrade the link performance. Thirdly, satellites in LSNs have a limited bandwidth and memory and need to minimize the communication overhead as much as possible. Lastly, the scale of LSNs makes it relatively challenging to maintain traffic forwarding [10]. Therefore, routing algorithms suitable for LSNs need to prioritize latency performance and packet reception success rates [11].

Traditional networks have undergone extensive research with regard to routing optimization [12–14]. However, the algorithms developed for these networks cannot be directly used for LSNs due to their unique features. Therefore, this paper focuses on studying the routing optimization problem specifically in LSNs, taking into consideration their distinct features. The main contributions of this paper are as follows:

1. The low-orbit satellite network model is defined as an unsplittable multi-commodity flow problem (UMCF) [15], which is then modeled as an MILP problem.
2. We propose a routing strategy called GAGR, which is based on GA assisted by ground access optimization; it incorporates GA to obtain an approximate optimal solution.
3. Before sending a flow from both the source and destination ground stations, we select three nearby satellites as candidate satellites, replacing the traditional strategy of selecting the nearest satellite.
4. We compare the performance of the GAGR routing strategy, virtual-topology-based shortest path (VTSR) routing strategy [16], and Floyd–Warshall (FW) [17] routing strategy under the UDP protocol and TCP protocol through experiments, and the experimental results demonstrate that our GAGR routing algorithm exhibits improved performance in terms of a reduced RTT, increased flow reception rate, and lower ISL occupancy rate.

Given the extensive use of abbreviations in this paper, we provide Abbreviations section containing their definitions for reference.

The rest of this article is organized as follows. Section 2 introduces the related work on satellite network routing. Section 3 introduces the network model and problem definition. In Section 4, we introduce the GA solution algorithm for UMCF and discuss the reasons for introducing three candidate satellites for ground stations. Section 5 describes the experiments and results. We summarize the paper in Section 6.

## 2. Related Works

### 2.1. Dynamic Satellite Routing

The performance of traffic scheduling is directly influenced by the network quality [18]. To address this, dynamic routing strategies have been proposed. Some approaches involve continuous packet exchange to obtain the most up-to-date network state description. For instance, building upon the optimized link state routing protocol, Ruiz et al. [19] proposed a method that senses the network state and detects any changes using specially designed packets. However, the complexity of networks may lead to a flood of detection packets. To tackle this issue, Li et al. [20] proposed a new protocol for adaptively selecting the optimal routing path in dynamically changing environments, using local and global pheromones based on the ant colony algorithm for QoS routing selection. Nevertheless, this method lacks an overall definition of the network and a comprehensive state tracking strategy.

Meanwhile, in order to improve satellite networks, some scholars have started utilizing software-defined network (SDN) technology. This technology allows for the separation of the data layer and control layer within the SDN architecture [21]. The primary function of the SDN controller entity is to concentrate the control logic and update the forwarding rules of data layer devices in response to network resource changes. Bi et al. [22] proposed an SDN-based composite architecture for a space fusion network, which integrates satellite backbone networks, ground mobile networks, and internet backbone networks for real-time signaling exchange, completing the transmission, storage, and processing of massive data and achieving the efficient utilization of network resources. Qi et al. [23] explained the benefits of integrating SDNs into satellite networks and proposed a multipath routing strategy based on an SDN, which used a combination of delay, bandwidth, and node load as the calculation factors for the link transmission costs.

### 2.2. Constellation State Adaptive Routing

Traffic scheduling needs to be adaptive due to dynamic satellite motion. Some studies [24,25] have implemented traffic scheduling and resource management using SDN controllers. However, the use of a centralized controller leads to increased latency and decreased accuracy with network scale and topology changes. Therefore, there is a need for more flexible algorithms that can handle dynamic topologies. To address the limitations of centralized controllers, Zhang et al. [26] have proposed a multipath routing strategy based on a hidden Markov model. This routing strategy predicts the future network connection state of a mobile node by analyzing its historical connection state. However, the limited resources of satellites, such as communications and packet processing capacity, pose challenges [27]. Therefore, reasonable traffic guidance is necessary to retard network congestion. Tang et al. [28] have developed a dynamical traffic scheduling strategy to implement multipath finding. This strategy calculates paths by estimating the topology and distributes traffic.

Fu et al. [29] proposed a routing approach that considers a mixed potential field based on depth and energy to make routing decisions. This approach shows good performance in stable networks. However, it needs a long convergence time when the network undergoes changes because its routing decision parameters rely on information from neighboring nodes. Liu et al. [30] proposed a routing algorithm that combines a static topology with a real-time link state. Based on the periodicity of satellite networks and the predictability of topology changes, priority is given to calculating static topology sequences. Once the link state changes, the routing forwarding table is recalculated using a breadth-first search

algorithm. This updated table is then broadcast across the entire network. Wang et al. [31] proposed a routing algorithm for LEO satellite networks based on random geometry. As an effective mathematical tool, random geometry is very suitable for the system-level analysis of satellite network topologies. This algorithm can effectively ensure that data packets are always forwarded in the established reference direction within a reliable range. Liu et al. [32] proposed a dynamic routing algorithm based on mobile ad hoc networks. In the article, the satellite network is modeled as a multi-hop wireless network cluster to avoid frequent topology changes causing broadcast storms. The authors combined the advantages of static routing and dynamic routing, making this algorithm highly autonomous and functionally compatible in satellite networks.

### 2.3. Virtual Topology Satellite Routing

This type of routing algorithm solves the dynamic topology problem of the constellation by discretizing the satellite network structure in time. In the virtual topology routing algorithm [33], the network topology at the beginning of each time interval is taken as the constant static topology during this period. By applying the shortest path algorithm on the ground, the forwarding table for each satellite is calculated. This strategy has low complexity. However, it is not ideal for unexpected situations in the network, such as link failures and cache congestion, which cannot be responded to and handled in a timely manner. It is suitable for satellite networks with high stability and a uniform load flow in space segment satellite loads and inter-satellite links [34]. Jia et al. [16] proposed the integration of the DFS and the Dijkstra algorithms into an SDN and virtual topology. The DFS algorithm is utilized to identify specialized nodes, while the Dijkstra algorithm is employed to handle the maximum traffic load.

Routing techniques based on virtual nodes have become one of the most popular research areas in satellite networks. By mapping actual satellite nodes onto static virtual nodes, such techniques shield the effects of the dynamic characteristics of satellite networks on routing. However, this approach might also lead to the congestion of links in high-latitude regions and increased queuing delays. To address this issue, Page et al. [35] proposed a distributed probabilistic congestion control method. It utilized periodic link state exchanges between satellites to obtain the real-time congestion information of adjacent nodes. Based on the congestion status, it probabilistically selected the next hop to balance the load across links in all directions. However, the link state information exchange in this method could experience delays, resulting in the outdated congestion status acquisition of nodes, which would reduce the accuracy of routing decisions.

## 3. Problem Definition and Network Model

### 3.1. Problem Definition

We define the satellite routing problem as an UMCF problem; it involves a graph with undirected edges and capacities assigned to those edges, and each satellite has a maximum of four neighboring satellite nodes that communicate with each other through ISL. The number of neighboring nodes for each satellite depends on the number of visible nodes for that satellite in the current time step. Additionally, there is a set of commodities, each with a source node, a destination node, and a demand value. In order to describe the satellite routing problem as an UMCF problem, we treat the traffic forwarding requests made by each ground station as commodities, which include three attributes: source ground station, destination ground station, and traffic demand. The goal is to find a unique path for each commodity to route their traffic demand from the source node to the destination node, without exceeding the capacity limit on the link when passing through the path. This problem has been extensively studied in the field of network optimization.

The UMCF at hand is known to be NP-hard, which means that finding optimal solutions for instances with a large number of nodes and commodities is extremely challenging. However, there are heuristics and approximation algorithms available that can still provide high-quality solutions for such instances. In this paper, we will focus on the GA.

### 3.2. Network Model

To model an LEO satellite network, we utilize undirected graph  $G(N, E)$ , where  $N$  represents the set of satellites and  $E$  represents the set of all possible links connecting the satellites. Additionally, we define a set of commodities, denoted as  $M(o_i, d_i, D_i)_{i \in I}$ , characterized by their source, destination, and demand. We summarize the main symbols and their explanations in Abbreviations section.

In order to ensure that each commodity can only choose one path to forward traffic in the end,

$$t_{ei} \in \{0, 1\}, \quad \forall i \in I, \quad \forall e \in E. \quad (1)$$

which  $t_{ei}$  indicates whether there is traffic on link  $e$  for commodity  $i$ . In order to ensure that the traffic can pass through the inter satellite link normally, we set the optimization objective as a constraint:

$$\min_{t_{ei}, \Delta_e} \sum_{e \in E} \Delta_e. \quad (2)$$

where  $\Delta_e$  represents the overflow on link  $e$ . To guarantee that the incoming and outgoing traffic of each commodity flowing through each satellite node is the same, we have the following constraint:

$$\sum_{e \in E^+(n)} t_{ei} = \sum_{e \in E^-(n)} t_{ei}, \quad \forall i \in I, \quad \forall n \in N. \quad (3)$$

where  $E^+(n)$  represents the incoming traffic of node  $n$ , and  $E^-(n)$  represents the outgoing traffic of node  $n$ . In order to ensure reasonable traffic overflow when traffic passes through link  $e$ , we have the following constraint:

$$\sum_{i \in I} t_{ei} D_i - c_e \leq \Delta_e, \quad \forall e \in E. \quad (4)$$

## 4. The Proposed Routing Strategy

In this section, we will introduce our inter-satellite routing algorithm and the first and last hop selection methods for ground stations to access and exit the constellation. Our inter-satellite routing algorithm is a heuristic method based on GA, which can effectively solve the UMCF problem and assign a unique path for each commodity. Our first and last hop selection methods for ground stations are simple methods based on the greedy strategy, which can select the optimal satellite node as the first and last hop according to the source and destination of the commodity. We will describe these two methods in detail in Sections 4.1 and 4.2, respectively. Before introducing the GA to our problem, we will introduce linear relaxation for MILP, the constraint in our problem.  $t_{ei} \in \{0, 1\}$  represents that each commodity can only select a single fixed path to send traffic. This is an NP-hard problem. By replacing this constraint with  $t_{ei} \in [0, 1]$ , we obtain the linear relaxation form of the problem. In the linear relaxation problem, each commodity is allowed to split its traffic among multiple paths. This linear relaxation problem can reduce the complexity in solving the original problem.

### 4.1. Routing Algorithm

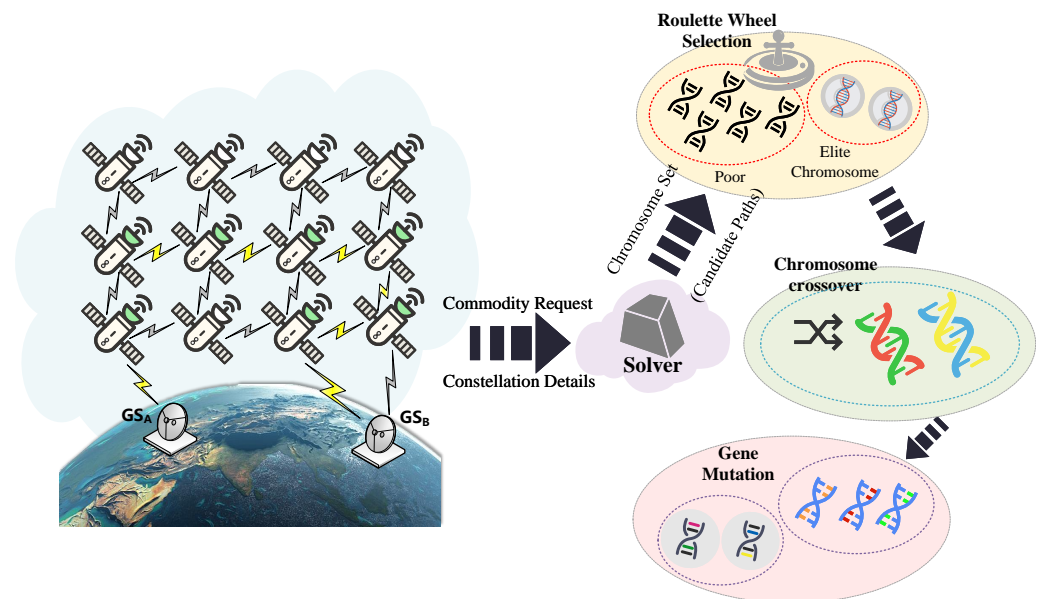
The GA offers a practical approach to performing a global search through multiple local searches. This algorithm relies on operations such as selection, crossover, and mutation to optimize the solution. In each loop, the GA generates a set of chromosomes, which represent candidate solutions for the optimization problem. The goal is to use the fitness value of each chromosome as a standard to find an approximate optimal solution over several generations.

The algorithm process is shown in Figure 1. In each generation, the roulette wheel selection method is used to select elite chromosomes. The crossover operation from elite chromosomes then selects the optimal chromosomes as parents. In this operation, the genes



of the two parent chromosomes are randomly exchanged with a predefined probability, giving rise to next-generation chromosomes. To maintain diversity and feasibility in the set of chromosomes, the genes in the next generation of chromosomes undergo mutation based on the following three principles:

1. Mutation operates on a randomly selected individual gene, i.e., an individual ISL;
2. The mutated gene is only allowed to be replaced with one of the three alternative ISLs on the same path;
3. We ensure that this hop after mutation is feasible.



**Figure 1.** GAGR routing strategy.

Once the crossover and mutation operations are completed, the next-generation chromosomes undergo evaluation based on the fitness value and problem constraints. This evaluation determines the fitness values of the chromosomes and prepares them for the next loop of operations. This process continues until the optimal chromosome remains unchanged for a specified number of generations. Table 1 summarizes the GA terminology that we will use in the algorithm.

The specific execution process of the routing algorithm is displayed in Algorithm 1, and we can see that it is generally executed alternately between two steps:

1. Solve the linear relaxation problem, where each product can have multiple paths that can forward the flow;
2. Utilize GA to ensure the uniqueness of the solution after linear relaxation, so that each commodity has a unique path.

The algorithm continues to execute until all commodities have a unique traffic forwarding path. Due to the fact that solving relaxation is easier than solving the satellite routing problem, our algorithm can achieve faster convergence times. Referring to the worst-fit memory allocation concept, we rank the commodities in descending order of demand. Studies have demonstrated the significant influence of this order on the heuristic's solution quality. After obtaining a linear relaxation solution, each commodity has several candidate paths.

**Table 1.** GA terminology.

GA Term	Definition
Generations	The number of iterations
Fitness	The objective function to measure the quality of candidate paths
Chromosome	The candidate path
Gene	The inter-satellite link on the candidate path
Population	The number of candidate paths in each loop
Elites	The best fitness value path in each iteration $e$

The GA treats the problem's solution set as a population and improves the solution quality through selection, crossover, and mutation operations, ultimately obtaining an approximate optimal solution. Therefore, it serves as a significant method in solving the aforementioned problem.

**Algorithm 1:** The GA heuristic

---

```

input :  $G = (N, E, c)$  a satellite network graph,
         $M = (o_i, d_i, D_i)_{i \in I}$  a set of commodities
output: a set of optimal paths that covers all commodities
1  tolerance = 5 ;
2  Sort the commodities by decreasing flow demand;
3  foreach commodity  $i$  do
4      if Implementation is required then
5          commodity  $i$ 's candidate paths set  $P$ 
6          = Solve_Linear_Relaxation_in_Gurobi( $G, M$ ) ;
7          Sort the  $P$  in decreasing order of fitness( $p$ ) ;
8          select elite chromosomes and add to new population  $P_{new}$ ;
9          foreach candidate path  $p$  in  $P$  do
10             /* Parent Selection Operation */
11             Using the Roulette Wheel to select two low fitness parents in  $P$ ;
12             /* Crossover Operation */
13              $P_{new} \leftarrow P_{new} + \mathbf{Crossover}(l, \text{parents})$  Algorithm 2;
14          end
15          foreach candidate path  $p$  in  $P$  do
16             /* Mutation Operation */
17              $P_{new} \leftarrow P_{new} + \mathbf{Mutation}(l, \text{chromosome})$  Algorithm 3;
18          end
19          sort population  $P_{new}$  in decreasing order;
20          if elite  $p$  same with previous generation then
21             tolerance  $\leftarrow$  tolerance − 1;
22             if tolerance = 0 then
23                 select best fitness path  $p$  as final;
24                 break;
25             end
26          end
27          initialize tolerance to initial tolerance
28      end
29 end

```

---

**Algorithm 2: Crossover**


---

**input** : paternal chromosome *parents*,  
 limitation on the number of crossover *l*  
**output**:  $P_{new}$  offspring chromosomes

```

1 while crossover attempts  $\geq l$  do
2   | apply loop crossover to selected parents;
3   | if offspring solutions are valid then
4   |   | add offspring to  $P_{new}$ ;
5   |   | break ;
6   | end
7   | crossover attempts  $\leftarrow$  crossover attempts  $- 1$ ;
8   | if crossover attempts = l then
9   |   | add parents to  $P_{new}$ ;
10  | end
11 end
12 return  $P_{new}$ ;

```

---

**Algorithm 3: Mutation**


---

**input** : offspring chromosome *chromosome*,  
*l* limitation on the number of mutations  
**output**: mutated offspring chromosomes  $P_{new}$

```

1 while mutation attempts  $\geq l$  do
2   | apply mutation to chromosome;
3   | if mutated solution is valid then
4   |   | add mutated chromosome to  $P_{new}$ ;
5   |   | break;
6   | end
7   | mutation attempts  $\leftarrow$  mutation attempts  $- 1$ ;
8   | if mutation attempts = l then
9   |   | add chromosome to  $P_{new}$ ;
10  | end
11 end
12 if no valid mutation then
13 |   | add original chromosome to  $P_{new}$ ;
14 end
15 return  $P_{new}$ ;

```

---

Firstly, we use a uniform-length integer-coded method to encode the chromosomes. The number of genes on each chromosome corresponds to the number of ISLs present on the candidate paths. The gene located at each position within the chromosome represents a specific ISL.

Secondly, the algorithm maintains high-quality genes by performing selection operations on the chromosomes. The fitness value serves as the basis for the selection process. We define the fitness equation as follows:

$$fitness = \begin{cases} 0, & \text{if cannot reach the destination.} \\ w_1 \times (min(rc_p)) + w_2 \times (min(d_p)), & \text{otherwise.} \end{cases} \quad (5)$$

where  $min(rc_p)$  represents the minimum remaining capacity among all candidate paths, and  $min(d_p)$  represents the minimum delay among all candidate paths. Meanwhile, the al-



gorithm employs the roulette wheel selection method for the chromosome selection operation. We define the probability that the candidate path  $p$  is selected, which is

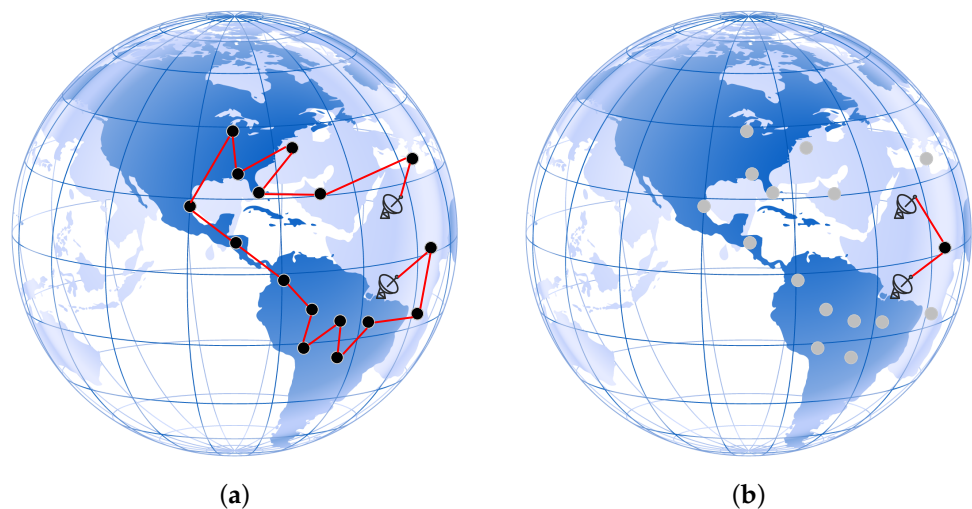
$$R(p) = \frac{fitness(p)}{\sum_{p=1}^{|P|} fitness(p)} \quad (6)$$

Based on Equation (6), the operation of roulette wheel selection prioritizes chromosomes with higher fitness values and eliminates unreachable chromosomes. In this way, the operation chooses chromosomes with better genes to be carried over to the next loop. Furthermore, our algorithm employs the best chromosome retention strategy. This strategy involves directly including the chromosome with the highest fitness value in the next population. Subsequently, the chromosome with the lowest fitness value is discarded to preserve the elite chromosomes.

Finally, we generate the next generation through crossover and mutation operations, which are important components of GA and help to maintain population diversity. For crossover operations, we employ the loop crossover method. This method involves selecting two low fitness chromosomes from the parents and performing a crossover operation on randomly selected genes at the same position on their chromosomes, resulting in the generation of new chromosomes. For mutation, we use single point mutation, i.e., when a candidate path is selected for the mutation operation, the chromosome is a candidate path, selecting one link on this path and randomly changing to one of the other three links on the forwarding node of this link.

#### 4.2. Ground Station Access

When a GS accesses the constellation, the satellite selection is typically based on the minimum distance. If multiple satellites are visible and accessible to a GS at a given moment, the one with the closest distance is directly chosen for data transmission. The satellites within a constellation are interconnected in a grid pattern, which allows them to maintain connectivity with several satellites consistently. Nevertheless, if satellites are moving in reverse directions, it is impossible to establish stable communication, even if they are neighbors. This means that two GSs that are geographically close may need to use a longer satellite path for communication. On the contrary, considering several adjacent satellites to the GS, different paths may be found, such as the source and destination satellites being in the same direction of motion. We present these two possible scenarios in Figure 2.



**Figure 2.** The impact of different access constellation strategies for GS. (a) Select nearest satellite to GS; (b) select one of the nearest satellites to GS.

To enhance the performance of existing routing algorithms, we propose conducting additional computations on the allowed incoming and outgoing hops within the constellation. The specific process of the algorithm is shown in Algorithm 4. The specified three nearest candidate satellites to all ground stations are stored in the memory at each time step. By comparing the three nearest satellites to the GS for all considered pairs, this process defines two scenarios as follows.

- (i) If the source GS and destination GS are located within the range of the three nearest candidate satellites, only one satellite is needed to establish the connection. In such cases, we can optimize the routing by directly utilizing this satellite node for forwarding.
- (ii) If two of the candidate satellites from two ground stations have direct inter-satellite links (ISLs), it means that these candidate satellites are neighboring each other in the constellation. In such scenarios, we can designate this pair of satellites as the source node and destination node.

---

**Algorithm 4:** GS Access/Exit Algorithm

---

**input :**  $G = (N, E)$ ,  $M = (o_i, d_i, D_i)_{i \in I}$ , time step,  
a visible satellite set for each GS in the current time step;  
**output:** routing table for the given time step;

- 1 Obtain the 3 nearest visible satellites for each GS;
- 2 **foreach** pair of considered GSs in the set of commodities **do**
- 3     compare their 3 nearest satellites;
- 4     **if** they have a common candidate satellite **then**
- 5         Generate a list of optimal first hop satellites and last hop satellites for the pair of GSs;
- 6     **end**
- 7     **if** they have candidate satellites directly connected through an ISL **then**
- 8         Generate a list of optimal first hop satellites and last hop satellites for the pair of GSs;
- 9     **end**
- 10    Select the nearest satellite separately and generate a list of the first and last hop satellites for the pair of GSs;
- 11 **end**
- 12 Calculate routing table through Algorithm 1;

---

## 5. Performance Evaluation

For LSNs, the communication delay is a key optimization indicator. Compared to the link capacity, reducing the delay is more important in improving the network performance. Therefore, we assigned weight  $w_1 = 0.3$  to the delay, and weight  $w_2 = 0.7$  to the remaining capacity in the simulation, giving more importance to the delay.

The performance of the proposed GAGR algorithm is assessed using several performance indicators, such as RTT, ISL utilization, the ratio of received/sent traffic, and the hop count, and compared to the VTSR algorithm [16] and FW routing algorithm [17]. VTSR is a DFS algorithm based on an SDN and virtual topology; the FW algorithm utilizes the idea of dynamic programming to calculate the shortest path between any two satellite nodes whenever the topology state changes.

To evaluate its performance in terms of the link capacity and flow distribution within the network, we utilize the UDP protocol and TCP protocol. We selected 100 pairs of cities from the set of the 100 most populous cities to represent 100 commodities, and these three routing algorithms were simulated for 60 s in the Telesat simulation constellation provided by Hypatia v1. Telesat's satellites use laser ISLs to communicate with each other. A radio up-link from the source GS to the ingress satellite, some laser ISLs, and a radio down-link from the egress satellite to the destination GS form an end-to-end path between two GSs.

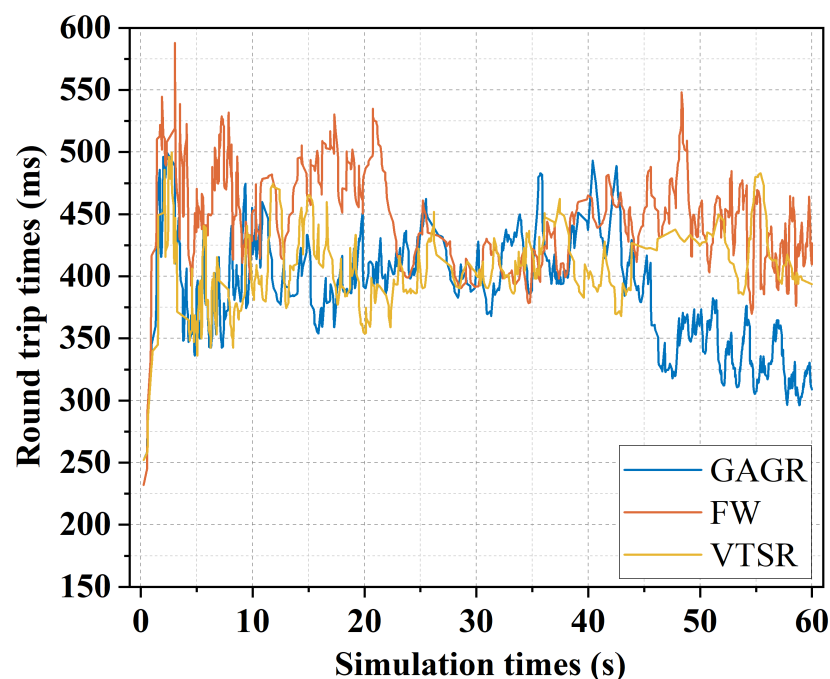
In order to simulate a case where the maximum capacity is insufficient, we set the flow size of each commodity to about 100 Mbit, which is much higher than the maximum capacity of the link, 32 Mbit/s. In this way, we can observe the behavior and bottlenecks of the link under a high load. Its other attributes are shown in Table 2.

The simulation environment is created using Hypatia on a personal computer configured as follows: Core i5-8300H 2.3 GHz CPU, 16GB RAM, and Ubuntu 20.04. Hypatia converts the continuous motion of satellites and the consequent changes in their paths into discrete processes. Although the delay on the path is constantly updated, the forwarding status is only recalculated at a fixed time step. Moreover, we use commercial solvers such as Gurobi 10.0.3 to solve the linear relaxation problems.

**Table 2.** Constellation parameters.

	Attribute	Value
TeleSat-1015	Alt. (km)	1015
	Inclin.	98.98
	N. Sat	351
	N. Orb	27
	N. Sat. Orb	13
	Cap. ISL. (Mbit/s)	32

Firstly, we calculated the RTTs of all commodities under the TCP protocol under the three routing algorithms, as shown in Figure 3. The simulation time is within [0, 45] s; our proposed GAGR algorithm has an average RTT of 404.02 ms, while the FW algorithm has an average RTT of 469.57 ms, and the VTSR algorithm has an average RTT of 453.93 ms. The average RTT ratio of the GAGR algorithm to the FW algorithm is 0.86, and the average RTT ratio to VTSR is 0.89. The average RTT of our proposed routing algorithm is better after 45 s; our proposed GAGR algorithm has an average round-trip time of 334.93 ms, while the FW algorithm has an average round-trip time of 482.01 ms, and the VTSR algorithm has an average round-trip time of 438.84 s. Compared with the FW algorithm, the average RTT ratio of GAGR is 0.69, and, compared with the VTSR algorithm, it is 0.76.



**Figure 3.** Round-trip times under TCP protocol.

In Figure 4, we show the average round-trip times of 100 commodities within the simulation time based on the UDP protocol and sort them in ascending order. During the entire simulation time, the average RTT of the GAGR algorithm is 378.51 ms, the average RTT of the FW algorithm is 472.64 ms, and the average RTT of the VTSR algorithm is 448.42 ms.

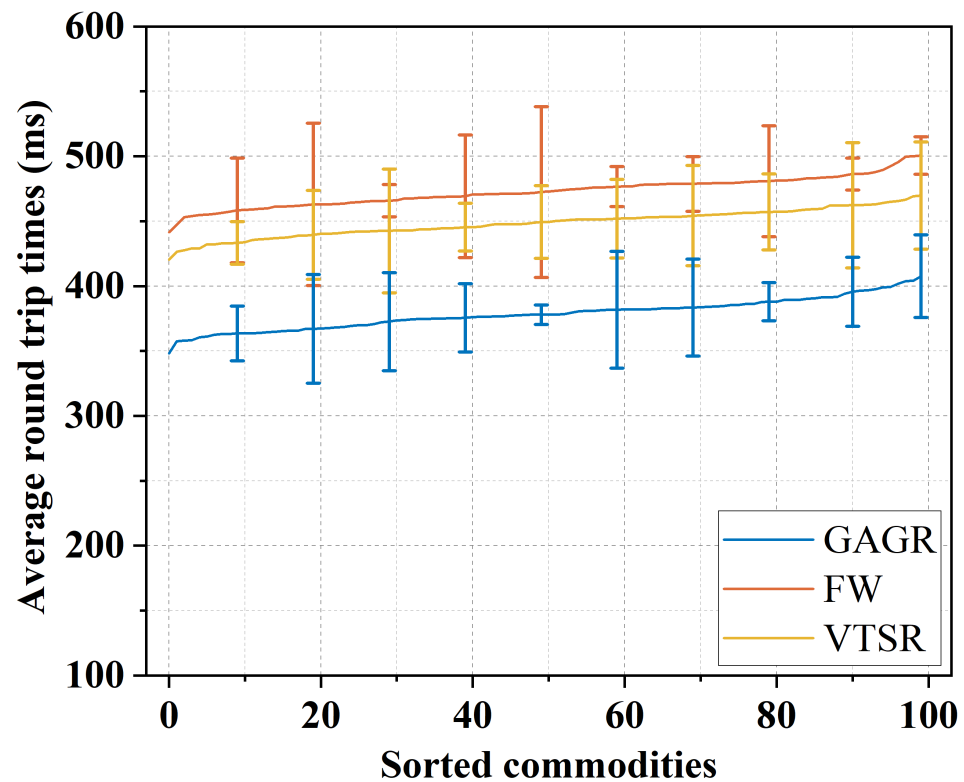


Figure 4. Average round-trip times under TCP protocol.

In Figure 5, we show the average round-trip times of 100 commodities within the simulation time based on the TCP protocol and sort them in ascending order. Among the first 87 commodities, the ratio of the average delay of the GAGR algorithm to the average delay of the FW algorithm is 0.8, and the ratio of the average delay of the VTSR algorithm is 0.84. However, both the FW algorithm and VTSR algorithm have 13 commodities that do not arrive at the ground station, resulting in packet loss, while the 100 commodities of the GAGR algorithm do not experience packet loss.

Secondly, we calculate the ISL utilization of all commodities under the TCP protocol under the three routing algorithms, as shown in Figure 6. We divide the simulation time into 10 s intervals, which can capture the dynamic changes in the network performance and the influence of our algorithm over time, showing the advantages of the GAGR algorithm in reducing the link utilization. Obviously, in all simulation intervals, the link utilization of the GAGR algorithm is lower than that of the FW algorithm and VTSR algorithm. The average link utilization of the FW algorithm is the highest, followed by that of the VTSR algorithm. Because the paths found by these two algorithms are not satisfactory, the number of forwarding hops is greater and more ISLs need to be passed. Similar to the situation under TCP, the monitoring link utilization under UDP is as shown in Figure 7. The link utilization of the GAGR algorithm is still the lowest among the three, because the approximate optimal solution is found in the optimization and iteration process of the candidate path, which reduces the number of forwarding hops.

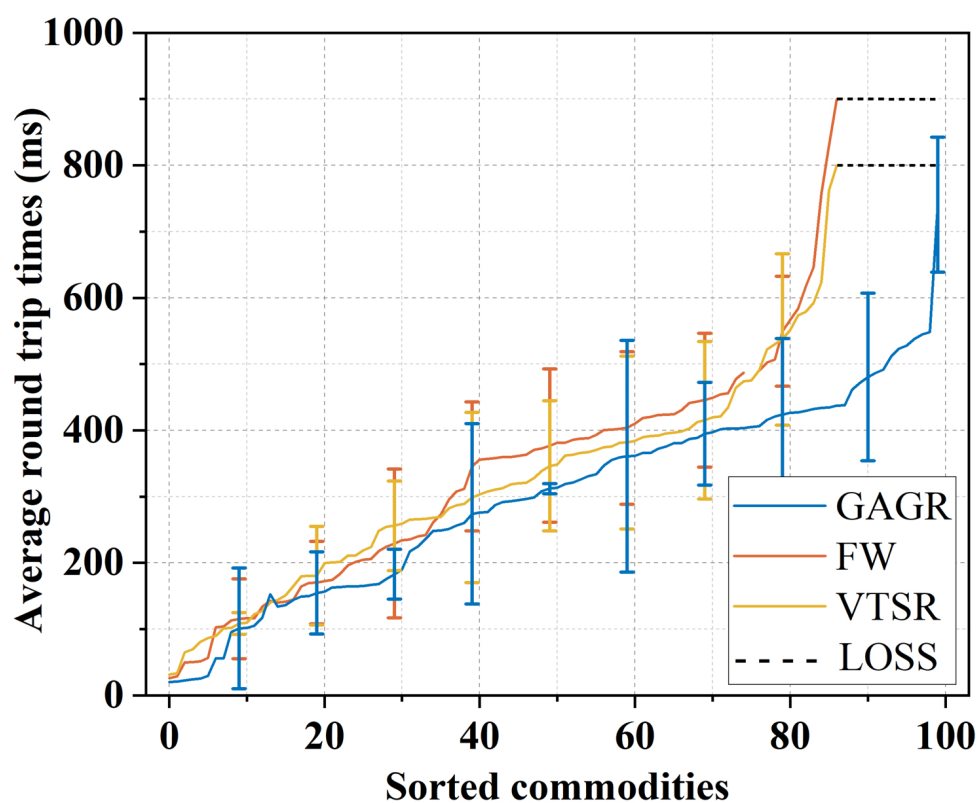


Figure 5. Average round trip times under UDP protocol.

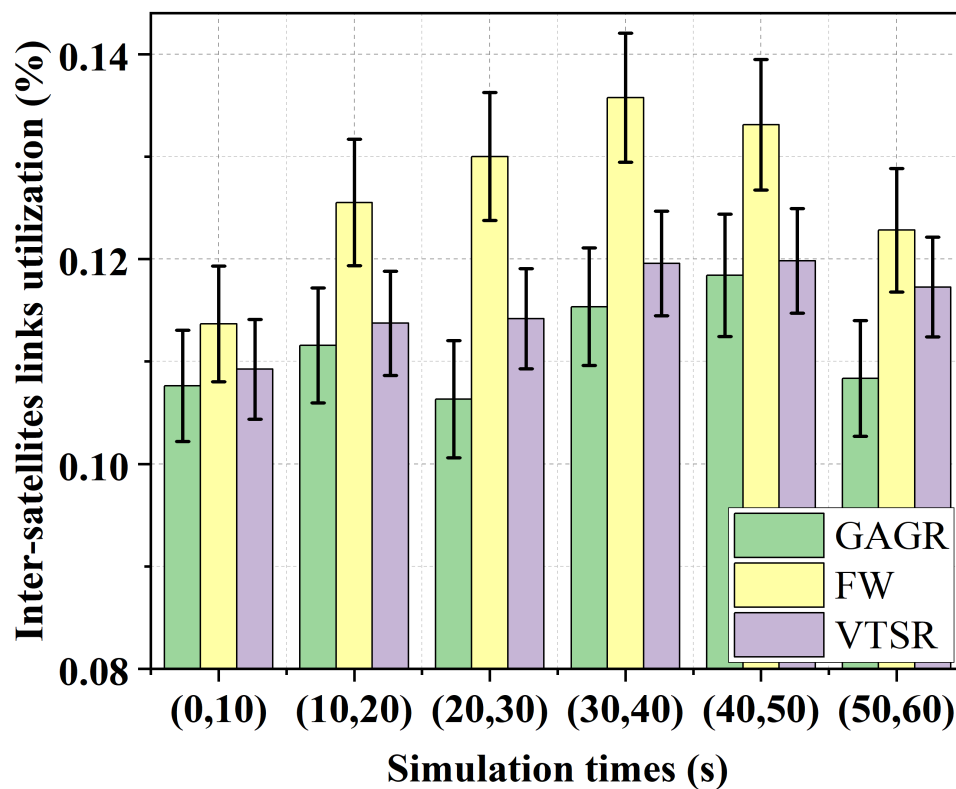


Figure 6. Inter-satellite link utilization under TCP protocol.

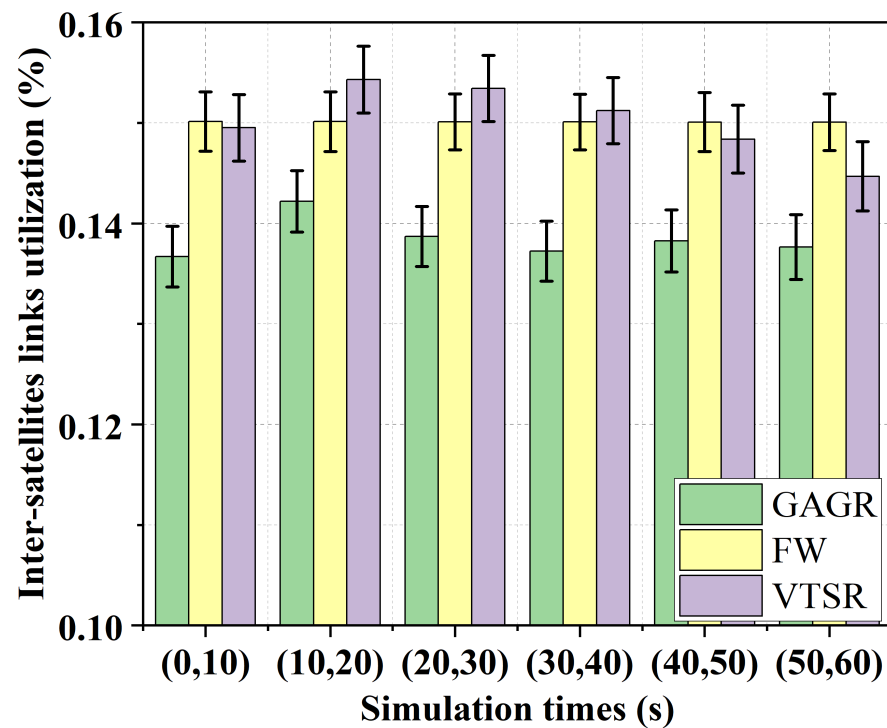


Figure 7. Inter-satellite link utilization under UDP protocol.

Furthermore, we sort each commodity based on its receive–send comparison according to the TCP protocol, as shown in Figure 8. Among the receiving performance of 100 commodities, the average receive–send ratio of our proposed GAGR algorithm is 0.49, with that of the FW algorithm being 0.38 and the VTSR algorithm being 0.42. In the UDP protocol, as shown in Figure 9, the average receive–send ratio of our proposed GAGR algorithm is 0.89, with that of the FW algorithm being 0.69 and the VTSR algorithm being 0.72. It can be seen that the GAGR algorithm has better throughput and provides more reliable network services.

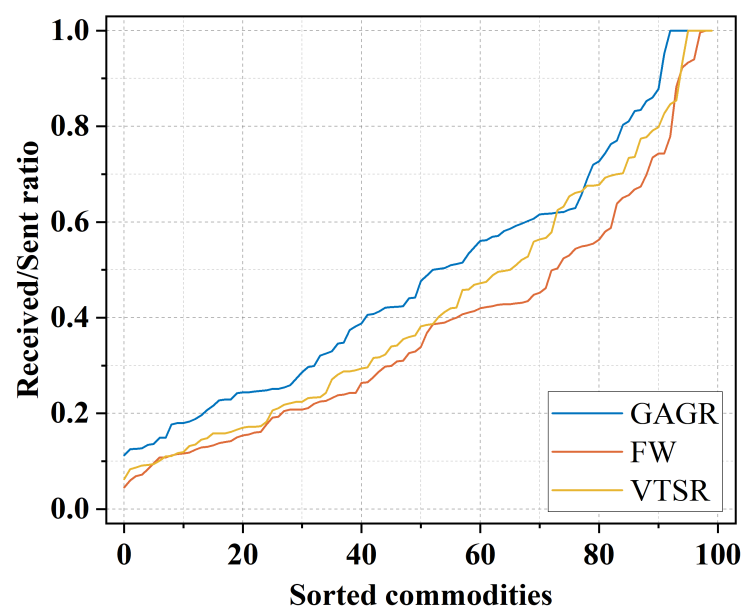


Figure 8. Received/sent traffic ratio under TCP protocol.



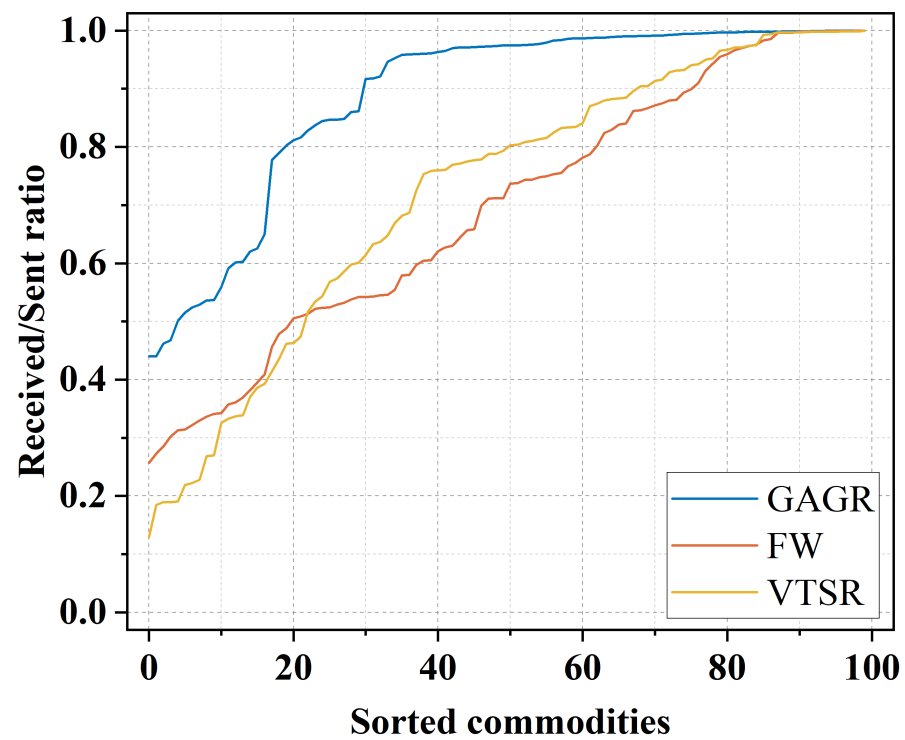


Figure 9. Received/sent traffic ratio under UDP protocol.

Finally, we compare the three routing strategies in terms of the number of hops for each commodity, as shown in Figure 10. FW requires the highest number of forwarding steps to complete a request, while GAGR has an average of 9.35 hops per commodity request and VTSR has an average of 9.44 hops per commodity request.

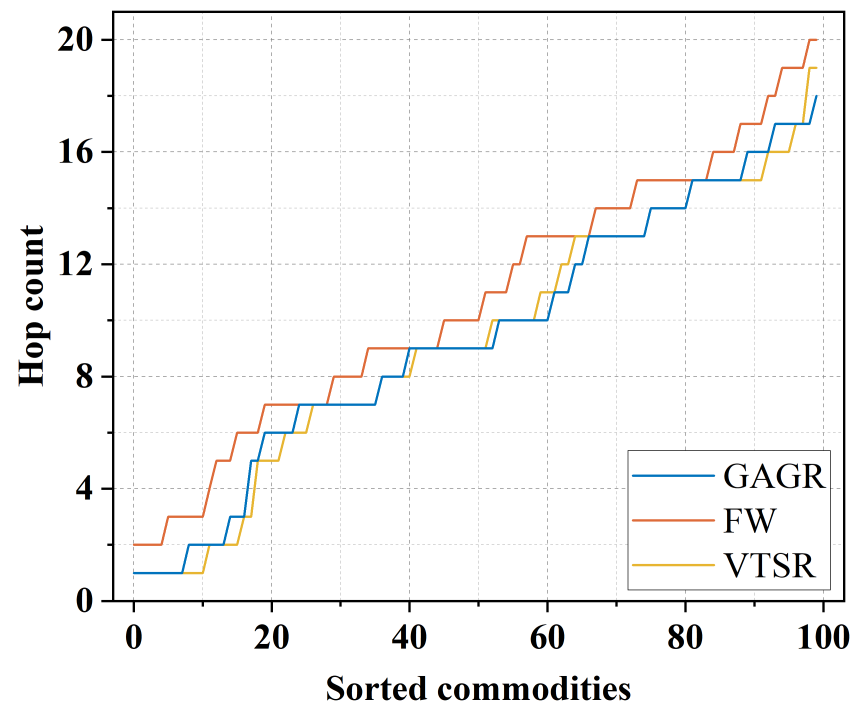


Figure 10. Hop count comparison.

## 6. Conclusions

In this paper, we propose a routing strategy based on GA assisted by ground access optimization for LEO satellite constellations. The proposed algorithm aims to optimize the routing strategy for LSN systems with ISLs, which can reduce the RTT and improve the packet reception rate. The experimental results show that the proposed algorithm outperforms the traditional satellite routing algorithm in terms of these metrics. Moreover, the proposed algorithm can adapt to different network topologies and traffic patterns, which makes it suitable for various LSN systems.

In summary, the proposed GAGR routing strategy provides a promising solution for the optimization of the routing strategy in LSN systems with ISLs. Although our proposed GAGR routing strategy demonstrates significant performance improvements in our experiments, it still inherently suffers from the drawbacks of heuristic algorithms, making it prone to falling into local optima. Future work will consider payload reconfiguration, introduce onboard processing techniques [36], incorporate machine learning techniques for predictive routing optimization to further improve the routing algorithm performance, strengthen the load balancing effect, increase the realism of experiments, and extend the proposal to other LEO satellite constellations.

**Author Contributions:** Conceptualization, P.Z. and G.X.; methodology, C.L. and H.W.; software, L.T.; validation, G.X., P.Z. and C.L.; formal analysis, H.W.; investigation, K.K.I.; resources, L.T. and P.Z.; data curation, K.K.I.; writing—original draft preparation, C.L. and G.X.; writing—review and editing, P.Z. and C.L.; visualization, H.W.; supervision, P.Z.; project administration, G.X.; funding acquisition, P.Z. and L.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is partially supported by the Natural Science Foundation of Shandong Province under Grants ZR2023LZH017, ZR2022LZH015, and 2023QF025; partially supported by the Industry-University Research Innovation Foundation of the Ministry of Education of China under Grant 2021FNA01001; partially supported by the Open Foundation of the State Key Laboratory of Integrated Service Networks (Xidian University) under Grant ISN23-09; partially supported by the Integrated Innovation of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grant 2023PX057; partially supported by the Talent Project of Qilu University of Technology (Shandong Academy of Sciences) under Grant 2023RCKY141; and partially supported by the RSF project under Grant 22-71-10095.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

Notation	Definition
LSN	Large-scale low Earth orbit satellite network
MILP	Mixed linear integer programming
RTT	Round-trip time
UMCF	Unsplittable multi-commodity flow
GA	Genetic algorithm
VTSR	Virtual-topology-based shortest path
FW	Floyd–Warshall
SDN	Software-defined network
QoS	Quality of service
GS	Ground station
ISL	Inter-satellite links
$G(N, E)$	The overall network topology
$N$	The set of satellites and GSs
$E$	The set of ISL

$M(o_i, d_i, D_i)$	The set of defined commodities
$o_i$	The source node of the commodity $i$
$d_i$	The destination node of the commodity $i$
$D_i$	The demand of the commodity $i$
$c_e$	The capacity on link $e$
$\Delta_e$	The overflow on link $e$
$rc_p$	The remaining capacity on the candidate paths $p$
$d_p$	The delay on the candidate paths $p$
$P$	The set of candidate paths for a commodity

## References

- Wang, C.; Zhang, P.; Kumar, N.; Liu, L.; Yang, T. GCWCN: 6G-based global coverage wireless communication network architecture. *IEEE Netw.* **2022**, *37*, 218–223. [\[CrossRef\]](#)
- Lin, X.; Hofström, B.; Wang, Y.P.E.; Masini, G.; Maattanen, H.L.; Rydén, H.; Sedin, J.; Stattin, M.; Liberg, O.; Euler, S.; et al. 5G new radio evolution meets satellite communications: Opportunities, challenges, and solutions. In *5G and Beyond: Fundamentals and Standards*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 517–531.
- Xu, G.; Zhang, Q.; Song, Z.; Ai, B. Relay-Assisted Deep Space Optical Communication System over Coronal Fading Channels. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, 1–16. [\[CrossRef\]](#)
- Huang, Y.; Jiang, X.; Chen, S.; Yang, F.; Yang, J. Pheromone incentivized intelligent multipath traffic scheduling approach for leo satellite networks. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 5889–5902. [\[CrossRef\]](#)
- Huang, Y.; Shufan, W.; Zeyu, K.; Zhongcheng, M.; Huang, H.; Xiaofeng, W.; Tang, A.J.; Cheng, X. Reinforcement learning based dynamic distributed routing scheme for mega LEO satellite networks. *Chin. J. Aeronaut.* **2023**, *36*, 284–291. [\[CrossRef\]](#)
- Hua, Q.; Hongqiang, W.; Jihong, Z.; Yongyue, Y. A Multi-dimensional Resource Allocation Algorithm Based on Task Splitting and Adjustment in Satellite Networks. In Proceedings of the 2022 3rd Information Communication Technologies Conference (ICTC), Nanjing, China, 6–8 May 2022; pp. 66–72.
- Jiang, C.; Zhu, X. Reinforcement learning based capacity management in multi-layer satellite networks. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 4685–4699. [\[CrossRef\]](#)
- Li, C.; He, W.; Yao, H.; Mai, T.; Wang, J.; Guo, S. Knowledge graph aided network representation and routing algorithm for LEO satellite networks. *IEEE Trans. Veh. Technol.* **2022**, *72*, 5195–5207. [\[CrossRef\]](#)
- Baeza, V.M.; Ortiz, F.; Lagunas, E.; Abdu, T.S.; Chatzinotas, S. Gateway Station Geographical Planning for Emerging Non-Geostationary Satellites Constellations. *IEEE Netw.* **2023**, *1*. [\[CrossRef\]](#)
- Leyva-Mayorga, I.; Soret, B.; Popovski, P. Inter-plane inter-satellite connectivity in dense LEO constellations. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 3430–3443. [\[CrossRef\]](#)
- Wang, P.; Zhang, J.; Zhang, X.; Yan, Z.; Evans, B.G.; Wang, W. Convergence of satellite and terrestrial networks: A comprehensive survey. *IEEE Access* **2019**, *8*, 5550–5588. [\[CrossRef\]](#)
- Ma, Y.; Peng, W.; Yu, W.; Su, J.; Wu, C.; Zhao, G. A distributed routing algorithm for LEO satellite networks. In Proceedings of the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Melbourne, VIC, Australia, 16–18 July 2013; pp. 1367–1371.
- Chen, Q.; Zheng, K.; Ouyang, F.; Gan, X.; Xu, Y.; Tian, X. A shortest path routing algorithm based on virtual coordinate in nels. In Proceedings of the 2016 8th International Conference on Wireless Communications & Signal Processing (WCSP), Yangzhou, China, 13–15 October 2016; pp. 1–5.
- Gounder, V.V.; Prakash, R.; Abu-Amara, H. Routing in LEO-based satellite networks. In Proceedings of the 1999 IEEE Emerging Technologies Symposium, Wireless Communications and Systems (IEEE Cat. No. 99EX297), Richardson, TX, USA, 12–13 April 1999; pp. 22.1–22.6.
- Lamothe, F.; Rachelson, E.; Haït, A.; Baudoin, C.; Dupé, J.B. Dynamic unsplittable flows with path-change penalties: New formulations and solution schemes for large instances. *Comput. Oper. Res.* **2023**, *152*, 106154. [\[CrossRef\]](#)
- Jia, M.; Zhu, S.; Wang, L.; Guo, Q.; Wang, H.; Liu, Z. Routing algorithm with virtual topology toward to huge numbers of LEO mobile satellite network based on SDN. *Mob. Netw. Appl.* **2018**, *23*, 285–300. [\[CrossRef\]](#)
- Kassing, S.; Bhattacharjee, D.; Águas, A.B.; Saethre, J.E.; Singla, A. Exploring the “Internet from space” with Hypatia. In Proceedings of the ACM Internet Measurement Conference, Virtual, Pittsburgh, PA, USA, 27–29 October 2020.
- Qu, L.; Xu, G.; Zeng, Z.; Zhang, N.; Zhang, Q. UAV-assisted RF/FSO relay system for space-air-ground integrated network: A performance analysis. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 6211–6225. [\[CrossRef\]](#)
- Ruiz-De-Azúa, J.A.; Camps, A.; Augé, A.C. Benefits of using mobile ad-hoc network protocols in federated satellite systems for polar satellite missions. *IEEE Access* **2018**, *6*, 56356–56367. [\[CrossRef\]](#)
- Li, G.; Boukhatem, L.; Wu, J. Adaptive quality-of-service-based routing for vehicular ad hoc networks with ant colony optimization. *IEEE Trans. Veh. Technol.* **2016**, *66*, 3249–3264. [\[CrossRef\]](#)
- Jiang, W. Software defined satellite networks: A survey. In *Digital Communications and Networks*; Elsevier: Amsterdam, The Netherlands, 2023.

22. Bi, Y.; Han, G.; Xu, S.; Wang, X.; Lin, C.; Yu, Z.; Sun, P. Software defined space-terrestrial integrated networks: Architecture, challenges, and solutions. *IEEE Netw.* **2019**, *33*, 22–28. [[CrossRef](#)]
23. Qi, H.; Guo, Y.; Hou, D.; Xing, Z.; Ren, W.; Cong, L.; Di, X. SDN-based dynamic multi-path routing strategy for satellite networks. *Future Gener. Comput. Syst.* **2022**, *133*, 254–265. [[CrossRef](#)]
24. Du, J.; Jiang, C.; Zhang, H.; Ren, Y.; Guizani, M. Auction design and analysis for SDN-based traffic offloading in hybrid satellite-terrestrial networks. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2202–2217. [[CrossRef](#)]
25. Ren, X.; Aujla, G.S.; Jindal, A.; Batth, R.S.; Zhang, P. Adaptive recovery mechanism for SDN controllers in Edge-Cloud supported FinTech applications. *IEEE Internet Things J.* **2021**, *10*, 2112–2120. [[CrossRef](#)]
26. Zhang, T.; Zhao, S.; Cheng, B. Multipath routing and MPTCP-based data delivery over manets. *IEEE Access* **2020**, *8*, 32652–32673. [[CrossRef](#)]
27. Xu, G.; Zhang, Q. Mixed RF/FSO deep space communication system under solar scintillation effect. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 3237–3251. [[CrossRef](#)]
28. Tang, F.; Zhang, H.; Yang, L.T. Multipath cooperative routing with efficient acknowledgement for LEO satellite networks. *IEEE Trans. Mob. Comput.* **2018**, *18*, 179–192. [[CrossRef](#)]
29. Fu, X.; Yang, Y.; Postolache, O. Sustainable multipath routing protocol for multi-sink wireless sensor networks in harsh environments. *IEEE Trans. Sustain. Comput.* **2020**, *6*, 168–181. [[CrossRef](#)]
30. Liu, Y.; Liu, C. Distributed dynamic routing algorithm for satellite constellation. In Proceedings of the 2018 10th International Conference on Communication Software and Networks (ICCSN), Chengdu, China, 6–9 July 2018; pp. 300–304.
31. Wang, R.; Kishk, M.A.; Alouini, M.S. Stochastic geometry-based low latency routing in massive LEO satellite networks. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3881–3894. [[CrossRef](#)]
32. Liu, Z.; Li, J.; Wang, Y.; Li, X.; Chen, S. HGL: A hybrid global-local load balancing routing scheme for the Internet of Things through satellite networks. *Int. J. Distrib. Sens. Netw.* **2017**, *13*, 1550147717692586. [[CrossRef](#)]
33. Tan, H.; Zhu, L. A novel routing algorithm based on virtual topology snapshot in LEO satellite networks. In Proceedings of the 2014 IEEE 17th International Conference on Computational Science and Engineering, Chengdu, China, 19–21 December 2014; pp. 357–361.
34. Zhang, P.; Zhang, Y.; Kumar, N.; Guizani, M. Dynamic sfc embedding algorithm assisted by federated learning in space-air-ground integrated network resource allocation scenario. *IEEE Internet Things J.* **2022**, *10*, 9308–9318. [[CrossRef](#)]
35. Page, P.S.; Bhargao, K.S.; Baviskar, H.V.; Kasbekar, G.S. Distributed Probabilistic Congestion Control in LEO Satellite Networks. In Proceedings of the 2023 15th International Conference on COMMunication Systems NETWORKS (COMSNETS), Bangalore, India, 3–8 January 2023; pp. 335–339.
36. Ortiz, F.; Monzon Baeza, V.; Garcés-Socarras, L.M.; Vásquez-Peralvo, J.A.; Gonzalez, J.L.; Fontanesi, G.; Lagunas, E.; Querol, J.; Chatzinotas, S. Onboard processing in satellite communications using ai accelerators. *Aerospace* **2023**, *10*, 101. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.