# Inter-Satellite Link Re-Planning Algorithm under Link Failures of LEO Satellite Constellations

Kaixin Chen†, Yongyi Ran*, Shaohua Xia†, Jiangtao Luo*, Shuangwu Chen‡
School of Communications and Information Engineering,
Chongqing University of Posts and Telecommunications, Chongqing, China*†
University of Science and Technology of China, Hefei, China‡
†{S220131002, S220101168}@stu.cqupt.edu.cn, *{ranyy, luojt}@cqupt.edu.cn, ‡{chensw}@ustc.edu.cn

2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall) | 979-8-3315-1778-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/VTC2024-Fall63153.2024.10757838

***Abstract*—The Low Earth Orbit (LEO) satellite constellation has been recognized as an important component of the future 6G network. However, due to the high-speed movement of LEO satellites and the potential for link failures, achieving optimal satellite communication performance with a static inter-satellite links (ISLs) scheme is challenging. To solve this problem, this paper proposes an ISL re-planning algorithm with considering link failures based on multi-agent deep reinforcement learning (named ReISL). In ReISL, a multi-objective optimization problem is formulated to maximize the system capacity while minimizing the link switching costs. Then, multi-agent deep reinforcement learning is employed to derive the optimal ISL re-planning schemes, where each satellite utilizes Double Deep Q-Network (DDQN). Finally, extensive experiments are carried out and the results demonstrate that our proposed algorithm ReISL can outperform the baseline algorithms.**

***Index Terms*—LEO satellite, link failure, inter-satellite link re-planning, multi-agent deep reinforcement learning.**

## I. Introduction

The Low Earth Orbit (LEO) satellite constellation is an emerging and promising technology to provide broadband communications, low latency services, and global coverage for ground users [1]. To provide efficient communication among users in different satellite coverage areas without the relay of any ground station, inter-satellite links (ISLs) are usually established between satellites. However, due to the complexity and uncertainty of the space environment, ISLs may fail due to satellite hardware damage or natural phenomena. These failures can be categorized into two types according to their duration: temporary link failures and persistent link failures. Temporary link failures are mainly caused by interference in the transmission medium (eg, solar activity) and last for a short duration; persistent link failures are mainly caused by satellite hardware failures (eg, transponder failures) and last for a long duration. Temporary link failures usually recover quickly with low impact on the system. In contrast, persistent link failures are difficult to repair, leading to a reduction in the number of available communication paths and a degradation in the quality of communications, which reduce system performance and efficiency. In addition, they lead to a waste of satellite resources. Therefore, how to re-plan ISLs to quickly adapt to link failures and optimize the network topology is a key field for future research.

However, it is challenging to solve the above problem due to the dynamic environmental states of the LEO satellite constellation. First, link failures are random, which are unpredictable and of variable duration. How to timely re-plan ISLs according to the link failure state is one of the keys to solve the problem. Second, a satellite has more than one candidate satellite, and there exists competition and cooperation between satellites within the same orbit. Competition with other satellites is to improve the quality of their inter-plane ISLs, while cooperation is to maximize the overall constellation performance. It is essential to achieve a good trade-off between competition and cooperation for the satellites in the constellation. Third, during link re-planning, when different satellites in the same orbit make decisions to the same candidate satellite, there will be a "decision conflict", resulting in some of the satellites being unable to establish ISLs, which leads to resource wastage.

Most of the existing ISL planning algorithms do not consider the impact of link failures on system performance. The basic ISL planning algorithms are heuristic [2],[3], which derive the scheme based on the partial information of the LEO constellation by greedy, genetic, and other methods, and are easy to result in local optimality. Some other researches are based on the Integer linear programming (ILP) algorithms [4],[5] and the Mixed-Integer Programming (MIP) algorithms [6],[7]. However, the complexity of the ILP and MIP algorithms grows exponentially as the constellation size increases. A Multi-agent Deep Reinforcement Learning (MADRL)-based ISL planning algorithm is proposed in [8]. However, this algorithm cannot completely solve the decision conflict problem. A satellite link model with three fixed ISLs and one dynamic ISL is designed in [9], which uses MADRL to reduce communication, storage, and computation costs. Nonetheless, the algorithm is not flexible enough and the system performance enhancement is limited.

To solve the above problem, we propose a MADRL-based ISL re-planning algorithm with considering link failures, named ReISL algorithm. In this algorithm, each agent makes actions based on its local observations and is trained with observations and actions of all the agents. Our main contributions are summarized as follows:

- We propose a MADRL-based distributed ISL re-planning algorithm with considering link failure information, which aims to quickly adapt to changes in the LEO satellite constellation.
- We model the ISL re-planning problem as a Markov

979-8-3315-1778-6/24/$31.00 ©2024 IEEE

Decision Process (MDP) and solve it using a Double Deep Q Network (DDQN) to achieve a good trade-off between satellite competition and cooperation in the constellation.

- We employ polling decision-making among agents within the same orbit plane and independent decision-making among agents across different orbit planes in each decision-making process in order to avoid decision conflicts.
- Extensive experiments are carried out and the results show that the ReISL algorithm can significantly improve the system capacity and reduce the switching cost.

## II. RELATED WORK

In this section, we review the recent research on the ISL planning algorithms of the LEO satellite constellation. Most research algorithms about ISL planning are focused on heuristic algorithms [2], [3], ILP algorithm [4], [5], MIP algorithm [6], [7] and deep reinforcement learning [8], [9], as described below.

Tu *et al.* [2] proposed an inter-plane ISL planning algorithm based on the greedy algorithm to jointly optimize the ISL switching energy consumption and data transmission energy consumption. Li *et al.* [3] defined a representation of satellite system design based on tree structure, and designed an ISL planning method based on genetic algorithm. The above methods are based on heuristic algorithms, which are easy to result in local optimality. Yan *et al.* [4] modeled the network of ISLs with Finite State Automaton (FSA) and solved the ISL planning problem based on ILP. Yan *et al.* [5] formulated the ISL topology design problem as an ILP problem, and proposed a more effective heuristic algorithm based on the maximum weight matching algorithm. Both [6] and [7] formulated the ISL planning and resource allocation problem as a MIP problem and solved it using the Lagrangian method. However, with the increase of constellation size, the complexity of ILP and MIP algorithms increases exponentially. To solve the high complexity problem, Pi *et al.* [8] proposed an approach based on the MADRL to train the algorithm orbit by orbit. However, this algorithm cannot completely solve the decision conflict problem. Wang *et al.*[9] designed a satellite link model with three fixed ISLs and one dynamic ISL and solved it with MADRL, but the algorithm is not flexible enough and the system performance improvement is limited. Additionally, the above methods do not consider the impact of link failures on system performance, which cannot adapt to link failures and re-plan ISLs in time.

## III. SYSTEM MODEL AND PROBLEM FORMULATION

### A. System Architecture

As shown in Fig. 1, we consider an inclined orbit constellation consisting of $N$ satellites evenly distributed across $M$ orbital planes. For the satellite $i$, we define its Cartesian coordinates as $(x_i, y_i, z_i)$ and the orbital plane it is located in as $m_i$. The constellation can be represented as an undirected

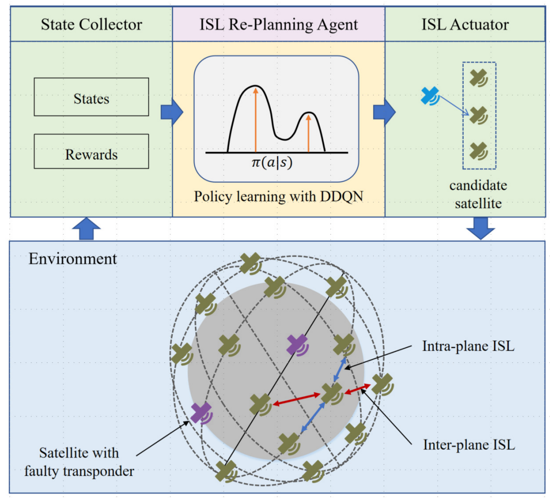



Fig. 1: LEO satellite constellation topology and decision network

graph $g = (V, E)$, where $V$ denotes the set of vertices (satellites), $E$ denotes the set of edges (ISLs). And the link between satellite $i$ and satellite $j$ is denoted as $e_{ij}$ .

Each satellite can establish two types of ISLs: intra-plane ISLs and inter-plane ISLs. Intra-plane ISLs denote links established by satellites within the same orbital plane, whereas inter-plane ISLs denote links established by satellites within adjacent orbital planes. We assume that all satellites actively decide to establish ISLs with satellites in the next orbital plane, defining this side as the positive side, and passively accept ISLs from satellites in the previous orbital plane, defining this side as the negative side.

Each satellite is considered as an ISL re-planning agent with a deterministic policy network and a value network. The state collector obtains the states and rewards by interacting with other satellites in the environment. The ISL re-planning agent makes decisions based on the state information collected by the state collector, and the ISL actuator establishes inter-plane ISLs with the corresponding satellites according to the instructions from the ISL re-planning agent.

### B. Communication Model

To describe the potential communication relationships between satellites, we introduce the concept of "candidate satellite pairs". The conditions for satellite $i$ and satellite $j$ to form a candidate satellite pair include multiple aspects. First, the Euclidean distance between satellite $i$ and satellite $j$ is less than the line-of-sight distance, expressed as

$$d_{ij} < d_{\text{horiz}}, \tag{1}$$

where $d_{ij}$ denotes the Euclidean distance between the satellites, denoted as

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2}, \tag{2}$$

and $d_{\text{horiz}}$ represents the line-of-sight distance between satellites, denoted as

$$d_{\text{horiz}} = 2\left[h\left(h + 2R_E\right)\right]^{1/2}, \tag{3}$$

where $R_E$ is the radius of the Earth. Second, satellite $i$ and satellite $j$ are not in the same orbital plane. ISLs within the same orbital plane are stable and do not require planning.

Based on the above analysis, the set of candidate satellite pairs for the inclined orbit constellation can be expressed as

$$Y = \left\{ ij : \left| m_i - m_j \right| \notin \{0\} \ \ \& \ \ d_{ij} < d_{\text{horiz}} \right\}. \tag{4}$$

After the link is established, the signal power received by the optical receiver is

$$P_r = P_t + G_t + G_r - \left[ L_p + \sum L_e + S_f \right], \tag{5}$$

where $P_r$ is the received signal power, $P_t$ is the transmitted power, $G_t$ is the transmitted antenna gain, $G_r$ is the received antenna gain, $L_p$ is the free-space path loss, $\sum L_e$ is the sum of other losses, and $S_f$ is the compensation factor. Assuming that ISLs are established adopting PIN photoelectric detector, the SNR can be expressed as

$$\frac{S}{N} = \frac{\xi P_r}{\Delta f h v}, \tag{6}$$

where $\xi$ is the quantum efficiency, $v$ is the frequency of incident light, $h$ is Planck constant, and $\Delta f$ is the unilateral broadband. The Bit Error Rate (BER) is denoted as $P_e$, when $P_e = 10^{-9}$ and $\xi = 1$, the link capacity can be expressed as

$$r_b = 2\Delta f = \frac{p_r}{18 h v}. \tag{7}$$

### C. Link Switching Cost Model

In this paper, we estimate the link switching cost based on the antenna steering angle. The antenna steering angle for switching $e_{i,j_1}$ to $e_{i,j_2}$ is

$$\theta_i^{j_1,j_2} = \arccos\left( \frac{\left(d_{ij_1}\right)^2 + \left(d_{ij_2}\right)^2 - \left(d_{j_1 j_2}\right)^2}{2 \cdot d_{ij_1} \cdot d_{ij_2}} \right). \tag{8}$$

To calculate the antenna steering costs of inter-plane ISLs, we define the average antenna steering angle $\hat{\theta}_i$ for each satellite $i$. For the $t$-th time slot, $\hat{\theta}_{i,t}$ is defined as [8]

$$\hat{\theta}_{i,t} = \frac{\sum_{j_1 \neq j_2 \in Y_{i,t}^+} \theta_i^{j_1,j_2} + \sum_{j_1 \neq j_2 \in Y_{i,t}^-} \theta_i^{j_1,j_2}}{\binom{N_{i,t}^+}{2} + \binom{N_{i,t}^-}{2}}, \tag{9}$$

where $Y_{i,t}^+$ and $Y_{i,t}^-$ represent the set of satellites $j$ that satisfy the condition $ij \in Y_t$ in the positive and negative side planes of the satellite $i$, respectively, $Y_t$ is the set of candidate satellite pairs for the $t$-th time slot. And $N_{i,t}^+$, $N_{i,t}^-$ represent the number of candidate satellite pairs in sets $Y_{i,t}^+$ and $Y_{i,t}^-$.

We define the antenna steering angle caused by establishing the inter-plane ISL $e_{ij}$ in the $t$-th time slot as

$$\theta_{e_{ij},t} = \begin{cases} 0 & , e_{ij} \in E_{t-1} \\ \hat{\theta}_{i,t} + \hat{\theta}_{j,t} & , e_{ij} \notin E_{t-1} \end{cases}, \tag{10}$$

where $E_t$ denotes the set of ISLs at the $t$-th time slot.

### D. Problem Formulation

The optimization objective of this paper is to increase the system capacity while reducing the link switching costs. For the $t$-th time slot, we define the utility function $\varphi(t)$ as

$$\varphi(t) = \sum_{e_{ij}} \alpha_1 r_{e_{ij},t} - \sum_{e_{ij}} \alpha_2 \theta_{e_{ij},t}, \tag{11}$$

where $\alpha_1$ and $\alpha_2$ are weight factors, $r_{e_{ij},t}$ denotes link capacity of $e_{ij}$. We assume that a satellite period has $N_d$ time slots. Therefore, the optimization problem can be formulated as maximizing the utility of the satellite network as follows

$$\begin{aligned} &\max \sum_{t=1}^{N_d} \varphi(t) \\ &s.t. \quad \begin{cases} ij \in Y_t \\ e_{ij} \in E_t \\ \alpha_1, \alpha_2 \end{cases}. \end{aligned} \tag{12}$$

## IV. The Proposed ReISL Algorithm

In this section, we introduce the proposed ReISL algorithm, which formulates the optimization problem as a model-free MDP and utilizes the deep reinforcement learning method to optimize the decision-making process.

### A. Elements of MDP

In the LEO satellite constellation, each satellite is considered as an agent. Within the MDP framework, agents do not rely on explicit dynamic models of the environment, but learn the optimal strategy through interaction with the environment. The key elements of MDP can be defined as follows.

**State Space**. For agent $i$, the state space $\mathbf{S}_i$ is defined as $\mathbf{S}_i = \left\{ \mathbf{D}_i, \mathbf{C}_i, \mathbf{F}_i^s, \mathbf{F}_i^r, \mathbf{I}_i \right\}$. Setting the number of candidate satellites for each satellite to $n$, the closest $n$ satellites in the next orbit will be candidates for satellite $i$. $\mathbf{D}_i = \langle D_{i,1}, \ldots, D_{i,n} \rangle$ denotes the set of normalized distances to candidate satellites; $\mathbf{C}_i = \langle C_{i,1}, \ldots, C_{i,n} \rangle$ indicates the set of states whether candidate satellites are occupied by decisions of other satellites or not, $C_{i,j} = 1$ indicates that the $j$-th candidate satellite is occupied, otherwise $C_{i,j} = 0$; $\mathbf{F}_i^s = \langle F_{i,1}^s, \ldots, F_{i,n}^s \rangle$ represents the set of positive-side transponder failure states of neighboring satellites in the same orbit, which can make them cooperate better, and the number of neighbors depends on the number of candidate satellites, $F_{i,j}^s = 1$ indicates that the positive-side transponder of the $j$-th neighbor satellite is faulty, otherwise $F_{i,j}^s = 0$; $\mathbf{F}_i^r = \langle F_{i,1}^r, \ldots, F_{i,n}^r \rangle$ represents the set of negative-side transponder failure states of candidate satellites in the next orbit, $F_{i,j}^r = 1$ represents the negative-side transponder of the $j$-th candidate satellite is faulty, otherwise $F_{i,j}^r = 0$; $\mathbf{I}_i = \langle I_{i,1}, \ldots, I_{i,n}, I_{i,0} \rangle$ denotes the connection state of satellite $i$ in the previous time slot, represented by a one-hot encoding, with $I_{i,j} = 1$ when the satellite $i$ is connected to the $j$-th candidate satellite, and $I_{i,0} = 1$ when the satellite $i$ is not connected to any candidate satellite.

**Action Space**. In this paper, we consider ISL re-planning as a discrete decision-making process. The action space $\mathbf{A}_i$ of agent $i$ is defined as $\mathbf{A}_i = \{v_0, v_1, \ldots, v_n\}$, where $v_0$ indicates the satellite does not establish an ISL with the satellite in the next orbit, and $v_j (1 \leq j \leq n)$ indicates the satellite selects the $j$-th candidate satellite to establish ISL.

**Reward**. We define the reward function for agent $i$ as $\sum_{k=1}^{N_{m_i}} r_k + \beta r_i$, where $\sum_{k=1}^{N_{m_i}} r_k$ represents the sum of satellite rewards in the orbital plane $m_i$, $N_{m_i}$ denotes the total number of satellites in the orbital plane $m_i$, $r_i$ represents the reward for the agent $i$, and $\beta$ represents the reward weight for the agent $i$. $r_i = \alpha_1 R_i - \alpha_2 \theta_i$, where $R_i$ represents the link capacity of satellite $i$, and $\theta_i$ represents the link switch cost of satellite $i$.

During each decision-making process, agents in the same orbital plane take turns to make decisions using a polling method and communicate the results to the next decision-making satellite to avoid decision conflicts. Decision-making for satellites in different orbital planes is independent of each other.

### *B. The Proposed ReISL Algorithm*

This paper uses the DDQN algorithm to optimize the decision-making process. The core framework of DDQN consists of an online network $Q$ with parameter $\omega$ and a target network $Q'$ with parameter $\omega'$, where the online network $Q$ selects optimal actions and the target network $Q'$ evaluates these actions. When the algorithm executes an action and observes a new state and reward, this information is stored as an experience tuple $(s, a, r, s')$ and added to the experience replay buffer. In the DDQN algorithm, exploration and exploitation are balanced by using the $\epsilon$-greedy strategy. At each time step, the algorithm selects a random action with probability $\epsilon$ and selects the currently estimated optimal action with probability $1 - \epsilon$.

In DDQN, the target value $y$ is calculated through the target network $Q'$. For each experience tuple $(s, a, r, s')$, the target value $y$ is defined as

$$y = r(s, a) + \gamma \cdot \max\left(Q'\left(s', a', \omega'\right)\right), \tag{13}$$

where $r(s, a)$ is the current reward, $\gamma$ is the discount factor, and $\max\left(Q'\left(s', a', \omega'\right)\right)$ is the maximum predicted value of the target network $Q'$ at the next state $s'$ and the next action $a'$.

The loss function $L(\omega)$ is defined based on the temporal difference error, measuring the discrepancy between the predicted values of the online network $Q$ and the target value $y$. The loss function $L(\omega)$ is defined as

$$L(\omega) = (y - Q(s, a, \omega))^2 \tag{14}$$

The parameter $\omega$ of the online network is updated by gradient descent algorithm, and the formula is

$$\omega_{t+1} \leftarrow \omega_t - \alpha \cdot \nabla_{\omega_t} L(\omega_t) \tag{15}$$

where $\alpha$ is the learning rate, $\nabla_{\omega_t} L(\omega_t)$ is the loss function gradient of online network parameter $\omega_t$. During the updating process, the algorithm calculates the gradient of the loss function with respect to the online network parameters and adjusts the parameters to optimize agent action selection.

The pseudo-code of training ReISL algorithm is shown in Algorithm 1, where initializations and training processes are the same for each agent.

**Algorithm 1** Training process of ReISL

**for** each agent $i = 1$ to $N$ **do**
  Initialize the online network $\omega_i$ and the target network parameters $\omega_i^{'}$ randomly
  Initialize ReplayBuffer
**end for**
**for** each episode **do**
  **for** each agent $i = 1$ to $N$ **do**
    Use $\epsilon$-greedy policy to select action $a$
    Establish an ISL with the corresponding target satellite
    Observe reward $r$ and next state $s'$
    Store tuple $(s, a, r, s')$ in ReplayBuffer
  **end for**
  **if** $t \mod T_{\text{update}} == 0$ **then**
    **for** each agent $i = 1$ to $N$ **do**
      Update the target network parameters: $\omega_i^{'} = \omega_i$
    **end for**
  **end if**
**end for**
**for** each agent $i = 1$ to $N$ **do**
  Sample a random batch $(s_b, a_b, r_b, s_b^{'})$ from ReplayBuffer for training
  Set $y_b = r(s_b, a_b) + \gamma \cdot \max\left(Q'\left(s_b', a_b', \omega'\right)\right)$
  Calculate loss according to Equation (14)
  Update the online network parameters according to Equation (15)
**end for**

## V. Experiment and Analysis

### *A. Simulation Setup*

In this paper, we use Python 3.9.15 and Pytorch 1.10.0 to build a simulation platform and conduct simulation experiments to verify the feasibility and effectiveness of the proposed algorithm. We model the distribution of ground services based on the distribution data of residential areas, and map ground services to inter-satellite traffic by calculating the coverage relationship between satellites and the ground based on the current constellation position. The constellation parameters are shown in Table I.

TABLE I: SATELLITE CONSTELLATION PARAMETERS

| Parameter | Value |
|---|---|
| Number of satellites | 120 |
| Number of orbital planes | 10 |
| Altitude of orbital planes | 1200 Km |
| Inclination of orbital planes | 56 deg |

In our experiments, we use the satellite transponder failure probability to simulate the satellite link failure. We mainly consider the persistent link failure and set the failure duration as 5 to 50 minutes. We set the weighting factors to $\alpha_1 = 1, \alpha_2 = 0.5$ during the training phase, and the total traffic demand is set to 90Gbps during the evaluation phase. We compared the performance of each algorithm as the satellite transponder failure probability varied from 0% to 3%. Other simulation parameters are provided in Table II.

TABLE II: PARAMETER SETTINGS FOR EVALUATION

| Parameter | Value |
|---|---|
| Simulation step | 5 min |
| Number of candidate satellites | 5 |
| Bit error rate tolerance | $10^{-9}$ |
| Maximum link bandwidth | 1.5 GHz |
| Failure duration | 5 ~ 50 min |
| Learning rate | 0.001 |
| Discount rate | 0.5 |
| Target network update cycle | 100 |
| Initial exploration rate | 0.9 |
| Exploration rate attenuation | 0.999 |
| Minimum exploration rate | 0.01 |
| Experience pool size | $2400 \cdot N_{sat}$ |
| Size of the Mini-batch | 1024 |

### B. Performance Metrics and Baseline Algorithms

**Performance Metrics:** 1) Global throughput: The sum of the traffic transmitted by all ISLs. 2) Packet loss rate: The proportion of unsuccessfully transmitted traffic in the network to the total network traffic. 3) Average ISL capacity: The average capacity of all inter-plane ISLs. 4) Link connection ratio: The ratio of successfully established inter-plane ISLs to the maximum inter-plane ISLs. 5) Link switching ratio: The ratio of switched ISLs to the total ISLs. 6) Average switching cost: The average angle of antenna steering caused by all link switching.

**Baseline Algorithms:** 1) GEO: Divide the latitude of a satellite constellation into multiple logical regions and establish inter-plane ISLs for satellites belonging to the same logical region. 2) GIEM: A dynamic inter-plane ISL planning algorithm based on the greedy algorithm [10]. 3) GMM: The extension of the GIEM algorithm aims at maintaining inter-plane ISLs as much as possible [10]. 4) Fixed-ISL: The traditional static link connection algorithm that establishes inter-plane ISLs between satellites on adjacent orbits with the same orbital slot index.

### C. Experiment Results and Analysis

1) Convergence analysis: Fig. 2 represents the reward convergence diagram. During the early stages of training, the decisions made by the satellites are suboptimal due to the randomly generated parameters of the neural network. However, the algorithm basically reaches convergence after 7000 episodes, as a result of the neural network's learning and parameter updates.

2) Algorithm comparison analysis: As shown in Fig. 3, when there is no satellite transponder failure, the performance of other algorithms is almost the same except GEO algorithm. This is because the GEO algorithm establishes ISLs for satellites belonging to the same logical region, and its performance is limited by region segmentation and the number of co-orbital satellites. However, as the failure probability increases, advantages of the ReISL algorithm gradually become apparent.

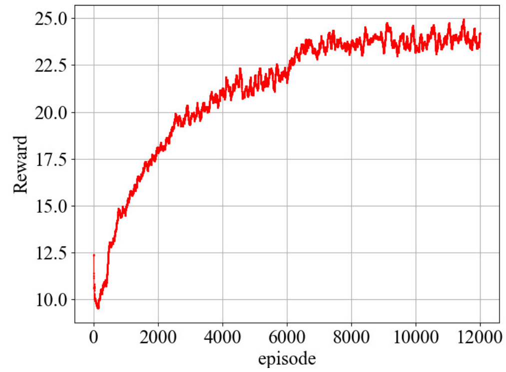



Fig. 2: Agents average reward after 12000 episodes

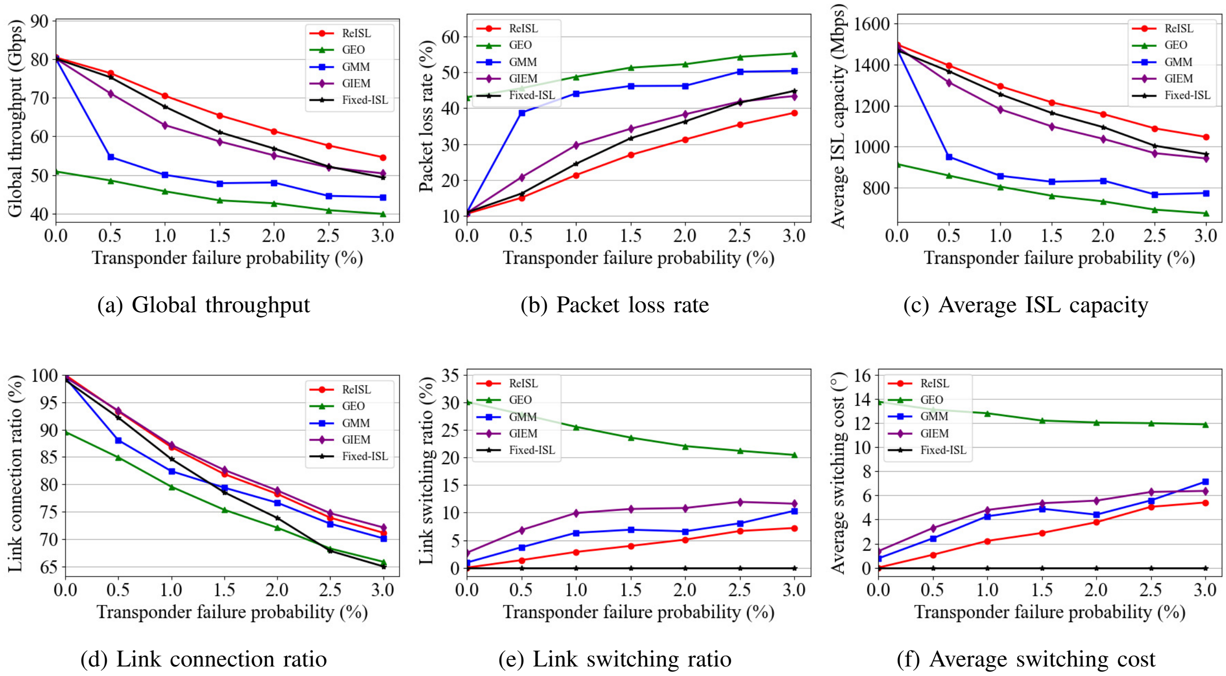
Fig. 3(a) shows that the throughput of satellite network decreases with the increase of the failure probability. However, compared with baseline algorithms, ReISL algorithm shows better throughput performance. Compared to other ISL planning algorithms, the throughput of the ReISL algorithm is improved up to 58.09%, and compared to the Fixed-ISL algorithm, it is improved up to 10.73%. Fig. 3(b) shows that the packet loss rate of ReISL algorithm is much lower than that of baseline algorithms, because the ReISL algorithm tends to select links that maximize the constellation system capacity under link failures. Fig. 3(c) shows that the average link capacity decreases with the increase of transponder failure probability. This is because link failures prevent some satellites from establishing links with the most suitable candidate satellites. Fig. 3(d) illustrates that the link connectivity ratio of the ReISL algorithm is higher than that of the GEO, GMM, and Fixed-ISL algorithms, but slightly lower than that of the GIEM algorithm, due to that GIEM algorithm does not consider switching cost. Fig. 3(e) and Fig. 3(f) respectively show the variations of the link switching ratio and the average link switching cost with increasing transponder failure probability. The Fixed-ISL algorithm does not perform link switching, so its switching ratio and switching cost are both 0. Compared to other ISL planning algorithms, the ReISL algorithm maintains the lowest switching ratio and switching cost.

## VI. Conclusion

This paper investigated the ISL re-planning of LEO constellation under link failure, to improve the system capacity of satellite constellation and reduce the link switching costs. We model the optimization objective as model-free MDP and use the ReISL algorithm to obtain the optimal decision. The experimental results show that compared with baseline algorithms, our proposed ReISL algorithm achieves better performance.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. U23A20275, 62171072, 62101525), and the Natural Science Foundation of Chongqing (No. cstc2021jcyj-msxmX0586).




(a) Global throughput
(b) Packet loss rate
(c) Average ISL capacity
(d) Link connection ratio
(e) Link switching ratio
(f) Average switching cost

Fig. 3: Comparison of results for different algorithms

## References


[1] X. Fang, W. Feng, T. Wei, Y. Chen, N. Ge, and C.-X. Wang, "5g embraces satellites for 6g ubiquitous iot: Basic models for integrated satellite terrestrial networks," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14 399–14 417, 2021.

[2] Z. Tu, H. Zhou, K. Li, M. Li, and A. Tian, "An energy-efficient topology design and ddos attacks mitigation for green software-defined satellite network," *IEEE Access*, vol. 8, pp. 211 434–211 450, 2020.

[3] J. Li, Y. Dong, M. Xu, and H. Li, "Genetic programming method for satellite system topology and parameter optimization," *International Journal of Aerospace Engineering*, vol. 2020, pp. 1–14, 2020.

[4] Z. Yan, G. Gu, K. Zhao *et al.*, "Integer linear programming based topology design for gnsss with inter-satellite links," *IEEE Wireless Communications Letters*, vol. 10, no. 2, pp. 286–290, 2020.

[5] Z. Yan, K. Zhao, W. Li, C. Kang, J. Zheng, H. Yang, and S. Du, "Topology design for gnsss under polling mechanism considering both inter-satellite links and ground-satellite links," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 2084–2097, 2021.

[6] R. Wang, R. Ma, G. Liu, W. Kang, W. Meng, and L. Chang, "Joint link adaption and resource allocation for satellite networks with network coding," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 12, pp. 15 882–15 898, 2023.

[7] R. Wang, W. Zhu, R. Ma, and et al., "Inter-satellite link scheduling and power allocation method for satellite networks," *Wireless Networks*, 2023.

[8] J. Pi, Y. Ran, H. Wang, Y. Zhao, R. Zhao, and J. Luo, "Dynamic planning of inter-plane inter-satellite links in leo satellite networks," in *ICC 2022-IEEE International Conference on Communications*, pp. 3070–3075. IEEE, 2022.

[9] G. Wang, F. Yang, J. Song, and Z. Han, "Optimization for dynamic laser inter-satellite link scheduling with routing: A multi-agent deep reinforcement learning approach," *IEEE Transactions on Communications*, pp. 1–1, 2023.

[10] I. Leyva-Mayorga, B. Soret, and P. Popovski, "Inter-plane inter-satellite connectivity in dense leo constellations," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3430–3443, 2021.