Dynamic optimization of intersatellite link assignment based on reinforcement learning

Weiwu Ren, Jialin Zhu, Hui Qi, Ligang Cong and Xiaoqiang Di

Abstract
Intersatellite links can reduce the dependence of satellite communication systems on ground networks, reduce the number of ground gateways, and reduce the complexity and investment of ground networks, which are important future trends in satellite development. Intersatellite links are dynamic over time, and different intersatellite topologies have a great impact on satellite network performance. To improve the overall performance of satellite networks, a satellite link assignment optimization algorithm based on reinforcement learning is proposed in this article. Different from the swarm intelligence method in principle, this algorithm models the combinatorial optimization problem of links as the optimal sequence decision problem of a series of link selection actions. Realistic constraints such as intersatellite visibility, network connectivity, and number of antenna beams are regarded as fully observable environmental factors. The agent selects the link according to the decision, and the selection action utility affects the next selection decision. After a finite number of iterations, the optimal link assignment scheme with minimum link delay is achieved. The simulation results show that in 8 or 12 satellite network systems, compared with the original topology, the topology calculated by this method has better network delay and smaller delay variance.

Keywords
Satellite link assignment, intersatellite link, intersatellite topology, reinforcement learning, network delay

Date received: 8 June 2021; accepted: 30 November 2021

Handling Editor: Peio Lopez Iturri

Introduction
In the 1990s, the LEO satellite constellation project was vigorously launched, such as the Iridium satellite project\(^1\) and the Globalstar project.\(^2\) These plans meet the needs of satellite relay calls. In recent years, with the reduction in satellite launch costs and the development of small satellite technology, an increasing number of countries and enterprises have begun to focus on the construction of LEO satellite constellations, such as the OneWeb\(^3\) and SpaceX\(^4\) programs in the United States and the Hongyun project\(^5\) in China. With the increasing number of LEO satellites, constellation networking has become necessary. The intersatellite link is the basis of constellation networking. At present, the design concepts of intersatellite links are different from each other. Even for satellite constellations under construction or in operation, the existing link design also has some problems. However, there is no doubt that the intersatellite link is very suitable for LEO constellations, which is not convenient for global distribution. It can enhance LEO constellations, improve the ability of independent network communication and expand the coverage of the communication system.
Intersatellite link assignment is the basis of constellation networking technology, and the communication performance of intersatellite links largely depends on the assignment of intersatellite links. The ground network topology generally does not change much over a long period of time, while the intersatellite link topology will change over time due to the influence of satellite motion. Therefore, it is necessary to conduct special research on intersatellite link assignment. To solve the problem of dynamic changes in link topology, Chang et al.6 divided the period of the LEO satellite system into equal time intervals by using the idea of finite automata. In each interval, the intersatellite visibility was not limited, and the topology of the intersatellite link remained unchanged. Follow-up studies have basically adopted this idea.

An effective intersatellite link assignment scheme is used to improve the overall or special performance of the network based on the service requirements. Focusing on this problem, many representative studies have been completed, which can be divided into three categories according to three different solutions:

1. The link assignment problem is constructed as an optimal combination problem. Real satellite conditions are mapped to various constraints. Performance indicators are set as optimization objectives. Various optimization algorithms have been applied to this model. To reduce the call blocking rate, Chang et al.6 used the simulated annealing algorithm to optimize the link assignment. Yan et al.7 optimized the satellite communication delay by the first improved and simulated annealing algorithm. Dong et al.8 optimized two objectives: network delay and position dilution of precision (PDOP), which not only improve the network performance of the navigation satellite system but also improve the navigation accuracy of the satellite. Liu et al.9 proposed a link assignment algorithm based on competitive decisions and link assignment based on simulated annealing for earth and moon space information networks. Wang et al.10 proposed the fast nondominated sorting genetic algorithm with elite strategy (NSGA-II) to solve the intersatellite link network routing.

2. The link assignment problem is modeled as a spatial search problem, and the graph search method is used to search for the optimal solution. Wang11 proposed a link assignment method for solving the link cost minimization of the whole network by the graph depth-first traversal method. Liu et al.12 minimized the end-to-end distance and maximized the link utilization by a perfect matching algorithm based on a bipartite graph.

3. The idea of divide and conquer is adopted to solve the problem of link assignment, and Shi et al.13 proposed a greedy link assignment algorithm. The communication cost was optimized, and the number of intersatellite observations was maximized. Zhou14 proposed an integrated weighted link assignment scheme. The link delay and link switching were optimized while ensuring link stability. Zhang et al.15 proposed a two-level solving scheme for solving satellite link scheduling. Qiao et al.16 proposed a dynamic programming method for intralayer link assignment optimization.

By studying this literature, we can find that there are many methods to solve intersatellite link assignment, and the effects are very different. The principles of these methods are similar. They all find the optimal solution as much as possible in local space. These methods are also the most effective methods for solving the problem of link assignment at present, but they cannot effectively show the solution process and give reasonable explainable results. Reinforcement learning can not only find the optimal solution but also provide an explainable process that imitates human behavior. However, there is currently no research or discussion on reinforcement learning applied to intersatellite link assignment. We propose that there are two reasons: the first is that there is no model suitable for directly using reinforcement learning. How can many problems be solved, including reinforcement learning concept mapping problems and training convergence problems? The second is that there is no value-based reinforcement learning method suitable for solving such a complex combinatorial optimization problem. The existing reinforcement learning model needs great space-time cost to solve this problem, and the existing model needs to be optimized.

Therefore, an intersatellite link assignment method based on reinforcement learning is proposed for the first time. The link assignment problem is modeled as the optimization of the link selection action sequence. The interaction model between link selection action and environment is established. By selecting the actions of intersatellite links by agents, the rewards related to the objectives are obtained, and link assignment is optimized by accumulating the multiple rewards.

Compared with the existing work, the method proposed in this article represents a new attempt at reinforcement learning in the field of satellite link assignment and provides a new solution for link assignment. In addition, in similar computational intelligence methods, evolutionary computing and other optimization algorithms focus more on the results of the solution, while reinforcement learning focuses more on the solution process. More human-like solution paths can be
obtained by agent behavior simulation, which can inspire people. Compared with other reinforcement learning methods applied to combinatorial optimization problem, our method is significantly different in principle, method, results, and conclusion. It is a more naive way of thinking about trying to solve this kind of problem.

Basic concepts

Scenario description

As shown in Figure 1, the intersatellite link assignment scenario is described. The satellites in Figure 1 are Walker constellation. A total of 12 satellites are distributed on three orbital planes, and there are four satellites on each orbital plane. It is assumed that each satellite can establish up to four intersatellite links with the surrounding visible satellites. For nonvisible satellites, they need to cross the earth, which cannot be realized by a microwave or laser. Since each satellite can communicate with up to four satellites at the same time, if the number of links is greater than four, it will be regarded as violating the constraint of the number of links. Due to the high-speed movement of the satellite, its intersatellite link topology changes at any time. How to optimize the performance of intersatellite links has become a complex and worthy research.

Link cycle table

Satellite cyberspace geometry will change over time. Using a static topology similar to the ground, it is likely to encounter satellite visibility and other issues caused by link interruption, which will lead to a decline in network performance. Satellite constellations are suitable for dynamic link topology, but the time of capturing satellite link is long enough that frequent link switching will lead to a decrease in the overall performance of the network. As a result, satellite constellation links need a time-sharing plan, that is, the intersatellite visibility in the constellation remains constant over an interval. Based on this idea, Chang et al. adopted the idea of a finite automaton, divided the cycles of low-orbit satellite systems into equal-length intervals, and maintained the intersatellite visibility unchanged during intervals. Dong et al. inherited this idea and divided the constellation system cycles into FSA states with one duration of tFSA. There is an overlapping link exchange time texch between two tFSAs. The intersatellite visibility remained the same during each (tFSA-texch) period. Multipair link switching is completed during the link exchange time. This article is also based on this idea. The link cycle table should be established by the convention that the intersatellite visibility is unchanged.

Link mathematical representation

According to the theory of weighted digraph, the problem of link assignment can be modeled as follows.

The whole constellation is regarded as a weighted digraph $G$, and its two tuples are defined as

$$ G = (V, E) $$

where $V$ is the vertex set, which satisfies the following conditions

$$ V = \{ v_i | 1 \leq i \leq N, v_i \in S \} $$

where $S$ is the collection of all satellites in the whole constellation, and the total number of satellites is $N$.

$E$ is the edge set of digraph $G$, which satisfies the following conditions

$$ E = \{ (v_i, v_j) | i \neq j, v_i, v_j \in V \} $$

The element in $E$ represents the intersatellite link. If the visibility constraint of the intersatellite link is not satisfied, it will not be inserted into $E$. The link assignment decision is represented by adjacency matrix $C$.

$$ C_{ij} = \begin{cases} R_{ij}, & (v_i, v_j) \in E \\ +\infty, & (v_i, v_j) \notin E \end{cases} $$

where $R_{ij}$ is the link cost, which is determined by the physical channel of the link. When $R_{ij}$ is equal to $+\infty$, the two satellites of the link are not visible. $C$ not only describes the link state at a certain time but also provides support for the agent to select link combinations.
**Constraints and goals**

Constraints can be divided into three categories:

1. **Intersatellite visibility constraints:** intersatellite visibility constraints are affected by satellite platforms, satellite-borne terminals, and satellite orbits;
2. **Connectivity constraints:** ensuring the connectivity of the whole network at any time is the most basic constraint;
3. **Link number constraint:** the number of links allocated by a single satellite does not exceed the number of satellite-borne antenna beams.

In the optimization process, link combinations that violate the above three constraints cannot occur.

A satellite system has high sensitivity to link delay, so the main optimization goal of this article is the delay performance of the whole network. The topology generation of the link is transformed into two delay optimization problems. Two delay indices are the average delay of the network and the maximum delay.

**Link assignment model based on reinforcement learning**

The link assignment model finds the optimal link combination that meets the constraints. However, traversing all the potential link assignment combinations requires considerable computations and storage space, which is expensive and does not meet the real-time needs of satellite networks. When we only consider 12 MEO satellites in the WALKER constellation, the space to traverse is close to $6 \times 10^{39}$. For LEO constellations that contain hundreds or even thousands of satellites, the computational efficiency will be lower. Therefore, a link assignment model based on reinforcement learning is proposed in this article. The agent selects the link combination, accumulates the selected utility into a selection decision, and finally iterates to achieve the best link assignment scheme.

The problem of link assignment is generally modeled as a combinatorial optimization model by evolutionary computation. Limited by principle, this kind of method has natural advantages and disadvantages that cannot be further improved. Therefore, this article attempts to establish a different model to solve this problem. With the agent’s behavior of selecting link combinations, obtaining the link combination feedback in satellite network environments is continuously attempted, and the next selection is rewarded through this feedback to select better link combinations. As shown in Figure 2, the real satellite network environment generates a variety of constraints, which restrict the generation and selection of link combinations. The agent selects link combinations and obtains the utility generated by this selection action, which is applied to the real satellite network environment. In addition, the action utility is updated to the policy decision, which affects the next selection. By learning the mapping from the environment to action, the agent can obtain the optimal environmental goal reward for the selected action.

In the application of reinforcement learning, two problems need to be solved: the definition of action and the design of the action utility function. For the action definition, our most intuitive idea is to define action as selecting different link combinations, but there is a problem: the selected action is single and random, and the utility of the previous action has no effect on the latter action, which makes the reward and memory of the action unable to be smoothly transferred to the next action, so it is difficult to generate the optimal solution. Therefore, we can decompose the action of selecting combinations into multiple consecutive actions of selection parts of combinations, that is, select the part of the combination first, then select a part from the remaining combination, and continue to select until all parts of the combination are selected. In this way, each selection action will have an impact on subsequent selection actions. Two variables affect the action utility: the average delay and the maximum delay of the network. The average delay can be indicated as $A_d$, and the maximum delay can be indicated as $M_d$. There are two goals for optimization: the minimum average delay and the minimum–maximum delay. These two goals may not be achieved at the same time. Therefore, the action utility ($A_u$) function can be designed as

$$A_u = M_d + w \cdot A_d$$

where $w$ is the weight of the average delay and maximum delay.

Realizing link topology traversal by selecting link combinations is inefficient. Here, we can adopt an equivalent scheme. We assume that the link topology is
fixed, and we can realize link topology traversal by changing the position of satellites, which not only reduces the action selection operations but also realizes link topology traversal.

**Link assignment algorithm**

We use a special data structure to store the value of each action utility. We call it $Q$. Here is a brief introduction of the data structure $Q$. As shown in Figure 3, it is a tree-like data structure. Let us assume that there are $N$ satellites ($N = k \times n$), from which $n$ satellites are selected. There are $P_n^N$ possibilities for this action of selecting $n$ satellites. In the same way, there are $P_n^{N-n}$ possibilities for the next action of selecting $n$ satellites. Similar selection will be performed continuously $k-1$ times until each satellite is traversed. When all satellites are selected, a satellite sequence is generated. According to this sequence, the corresponding link combination can be generated. The average delay and the maximum delay can be calculated based on the link cost in matrix $C$, and then the action utility value ($Au$) of selecting $n$ satellites is also calculated. The variable $min$ is used to point to the action of the minimum $Au$ value in the next possible action, which can reduce the number of times the whole data structure is traversed.

The input of the algorithm includes three parts: the edge set $E$, the cost matrix, and the initial parameters. Edge set $E$ is generated by satellite visibility constraints, and the link can be established if two satellites are visible. The communication quality of the link determines the cost of the link. When the current cost is infinite, the satellite cannot be visible, and the link cannot be assigned. Some parameters of the algorithm also need to be input, including the total number of iterations $IN$, preset target $D$, learning rate $SS$, greedy factor $Gr$, discount factor $DC$, and multigoal weight $w$. When the number of iterations exceeds $IN$, the algorithm will automatically stop and output the current optimal link assignment. By setting the preset target value $Dr$, the algorithm can stop automatically when it reaches $Dr$, which can save time on the premise of satisfying the user’s target. The learning rate $SS$ affects the action reward; if the $SS$ is less than 1, it strengthens the historical action reward. If $SS$ is equal to 1, it eliminates the historical action reward. The greedy factor $Gr$ affects the randomness of the selection action. The randomness of the selection action increases with the growth of $Gr$. Otherwise, the randomness decreases. The discount factor $DC$ affects the future action reward; the larger the $DC$, the greater the future action reward. Otherwise, the future action reward decreases. The multigoal weight $w$ affects the goal weights. Adjusting the value of $w$ obtains different goals that contain different average delays and maximum delays. The input can be adjusted dynamically according to the actual situation, and the output generates different link assignments.

The iterative selection process of the agents is very similar to that of human beings.

The algorithm output includes three parts: link assignment, link topology, and link performance. The form of the link assignment is $\{(S_1, S_2), \{S_1, S_3\}, \ldots \{S_i, S_j\}\}$, where $S_i$ is the $i$th satellite and $S_j$ is the link between the $i$th satellite and $j$th satellite. The link topology is generated by the link assignment. The link performance is the average delay and maximum delay of the output link assignment.

According to the visibility constraint of the link, the edge set $E$ is first initialized to generate a visibility matrix, and then based on the visibility matrix, the link
Algorithm 1. DOLA(E, C, Parameters)

1: Input(E)
2: Input(C)
3: Initialize(Q)
4: while \( IN < \text{threshold} \) do
5: \( \text{Input} \text{Parameters}(IN, D, SS, Gr, DC, w) \)
6: if FSB is false then
7: \( \text{while SLN > TLN do} \)
8: \( \text{while Dist} (i,j) > \text{INF do} \)
9: \( \text{Generate}(Rd) \)
10: \( \text{if } Rd > Gr \text{ then} \)
11: \( \text{SelectionMAction}(Q) \)
12: \( \text{else} \)
13: \( \text{SelectionRAction}() \)
14: \( \text{end if} \)
15: \( \text{end while} \)
16: \( \text{end while} \)
17: \( \text{Calculate}(Au) \)
18: \( AuinM = DC \times \min(Au) \)
19: \( \text{Calculate}(Rt) \)
20: \( \text{if } Rt > D \text{ then} \)
21: \( \text{FSB} = \text{True} \)
22: \( \text{else} \)
23: \( \text{Update}(Q) \)
24: \( \text{end if} \)
25: \( \text{IF} \text{ } \text{IN} = \text{IN} + 1 \)
26: \( \text{end if} \)
27: \( \text{end while} \)
28: \( \text{Output} \text{the optimal intersatellite link assignment} \)

FSB: final state bit; DC: discount factor; IN: total number of iterations.

If the link number constraints are not met, it indicates that some satellites carry too many links, which is inconsistent with reality. If the visible constraints are not met, there are some links in the scheme that cannot be directly connected. All selection actions that violate the constraints are invalid and are reselected. Once the link assignment scheme meets all constraints, the action utility \((Au)\) value is calculated and indicated as the predicted value of selection \((pre)\). Not all action utility values are meaningful. Only the link assignment with minimum cost has the optimal selection action utility value. The optimal selection action utility value can affect the next selection action. Therefore, we need to remember the optimal action utility value. Here, we set a discount factor \((DC)\), indicating that the previous action utility value may be forgotten or ignored. The formula is as follows

\[
AuinM = DC \times \min(Au)
\]

The result value \((Rt)\) represents the sum of the current action utility value \((Au)\) and the action utility value in memory \((AuinM)\), which is used to represent the overall reward of all actions thus far. When \(Rt\) reaches the expectation, the FSB is set to true, the number of iterations increases by 1, and a round of iterations is carried out. After each selection action is executed, the action utility matrix \(Q\) is updated, and each \(Au\) is affected by the overall reward and the utility value of the last action. The specific calculation is as follows

\[
Au = Au + SS \times (Rt - Pre)
\]

If the number of iterations is greater than the threshold, the program will not continue. In addition, the result may not converge, that is, the optimal link combination is not found.

In each iteration of satellite link assignment algorithm based on reinforcement learning, the parts with high time complexity are the process of action selection and the process of network delay calculation.

As shown in Figure 3, in the process of action selection, it is necessary to first select \(n\) satellites for link assignment, perform not more than \(P_N^n\) times of satellite combination selection action, and select the action with the greatest action utility, and then continue to select \(n\) satellites from the remaining satellites for link assignment. It is necessary to perform not more than \(P_{N-n}^n\) times of satellite combination selection action, and so on, the total times of satellite combination selection action shall not exceed \(k \times N^3/n!\). Therefore, its running time cost is \(O(n^6)\).

The process of network delay calculation needs to calculate the delay of the optimal link allocation of \(n\) satellites, and the delay value of not more than \(2^{n(n-1)}\) link assignment combination needs to be calculated. Therefore, its running time cost is \(O(2^n)\).
When \( n > 16 \), the time complexity of the algorithm is \( O(\max(n^4, 2^n)) = 2^n \); when \( n \leq 16 \), the time complexity of the algorithm is \( O(\max(n^4, 2^n)) = n^4 \). Therefore, we should try to select a smaller \( n \), that is, the number of satellites selected for each action should not exceed 4, so as to ensure that the time complexity of the algorithm does not exceed \( O(n^4) \). Our algorithm is a local search method. By the action selection mechanism of reinforcement learning, a large-scale solution problem is decomposed into several phased subproblems, so that a complex problem can be solved in limited time and space. Compared with the random search algorithm,\(^{19,20}\) our time complexity is still large. However, the method proposed in this article is a greedy algorithm that continuously approximates the solution, and its randomness is much lesser than the random search method, so the algorithm is easier to converge and stable.

**Performance analysis**

The dataset is an adjacency matrix, and the element of the matrix is the link delay. First, the AGI System Tool Kit (STK) is used to simulate the satellite spatial position, analyze the visibility between satellites, and then use the STK to estimate the distance of visible satellites. According to the estimated distance, the delay of the satellite link is calculated by using the delay formula. The delay formula is as follows

\[
T_{od} = T_{rd} + P_{pd} + P_{cd} + Q_{ud} \tag{8}
\]

where \( T_{od} \) represents the total delay, \( T_{rd} \) is the transmission delay, \( P_{pd} \) is short for the propagation delay, \( P_{cd} \) stands for the processing delay, and \( Q_{ud} \) represents the queuing delay. Because the algorithm verifies the impact of spatial location on link assignment optimization, it ignores the impact of data size and complexity on the link. Only the propagation delay is calculated. The propagation delay formula is as follows

\[
P_{pd} = C_l / T_s \tag{9}
\]

where \( P_{pd} \) is short of the propagation delay, \( C_l \) represents the channel length, and \( T_s \) is the transmission speed of the electromagnetic wave on the channel. After the above calculation, we obtain a delay matrix based on the distance between satellites.

The constellation simulation adopts two configurations: Walker8/2/1 and Walker12/3/1, which are distributed in the MEO layer. The track height is 21,528 km, and the track inclination is 55°.

The simulation experiment is divided into four parts: optimizing the model, generating the optimal link topology, link changes in the different topologies, and the performance comparison in the case of disruption. The first part is shown in Figures 5–8. The second part is shown in Figure 9. The third part is shown in Figures 10 and 11. The fourth part is shown in Figures 12 and 13. Figure 5 shows the delay comparison when the number of links is 3 and the number of satellites is 8. When the number of iterations exceeds 6,000, the average delay converges and reaches a stable value. When the number of iterations exceeds 8,000, the maximum delay converges and reaches a stable value. The average delay fluctuates in the initial stage of iterations and is not monotonically reduced. This shows that the algorithm can avoid the local optimal solution automatically to obtain the global optimal solution. The two delays converge and reach stability, which shows that the agent can find the optimal solution in the current simulation.

Figure 6 shows the delay comparison when the number of links is 4 and number of satellites is 8. When the number of iterations exceeds 13,000, the average delay converges and reaches a stable value. When the
number of iterations exceeds 9,000, the maximum delay converges and reaches a stable value. The two delays converge and reach stability, which shows that the agent can find the optimal solution in the current simulation. By comparing Figures 5 and 6, we can see that when the number of links increases from 3 to 4, the average delay and the maximum delay decrease. This is because the increase in links brings more low delay links and a greater possibility of link topology. The link assignment algorithm can find the link topology with lower delay. Therefore, both delay performances are improved. In addition, when the number of links increases from 3 to 4, the number of iterations increases. This is because the increase in links leads to an increase in the calculation.

Figure 7 shows the delay comparison when the number of links is 3 and the number of satellites is 12. When the number of iterations exceeds 17,000, the average delay converges and reaches a stable value. When the number of iterations exceeds 29,000, the maximum delay converges and reaches a stable value. The two delays converge and reach stability, which shows that the agent can find the optimal solution in the current simulation.

Figure 8 shows the delay comparison when the number of links = 4 and the number of satellites = 12. When the number of iterations exceeds 25,000, the average delay converges and reaches a stable value. When the number of iterations exceeds 31,000, the maximum delay converges and reaches a stable value. The two delays converge and reach stability, which shows that the agent can find the optimal solution in the current simulation.

By comparing Figures 7 and 8, we can see that when the number of links increases from 3 to 4, the delay performance is improved, and the number of iterations is also increased. This is very similar to the
result of the increase in links. In addition, by comparing Figures 5 and 7, we can see that when the number of satellites increases from 8 to 12, there is little difference in the delay comparison, but the number of iterations for convergence increases obviously. This is because the increase in satellites requires more calculations. It can be seen in Figures 5–8 that the link assignment algorithm based on reinforcement learning is effective in different types of satellite constellations.

Figure 9 shows three different link topologies. When the number of links is 3 and the number of satellites is 12, Figure 9(a) shows the initial link topology. The initial link topology can be manually set in the first FSA state, or it can be the previous link topology retained in the previous intervals. Figure 9(b) is the static topology. When the number of links is 3, each satellite has three fixed links in the static link topology, two of which are established with two adjacent satellites in the same orbit, and one of which is established with one adjacent phase satellite in a different orbit. Figure 9(c) shows the link topology generated by the optimal algorithm.

Figure 10 shows the number of link changes from the initial topology to the optimized topology. It can be seen that the increase in links and the increase in satellites affect the number of link changes. When the number of links increases from 3 to 4, it leads to an increase in link changes. When the number of satellites is 8, the number of link changes is 18. When the number of satellites is 12, the number of link changes is 29. When the number of satellites increases from 8 to 12, it also leads to an increase in link changes. When the number of links is 3, the number of link changes is 24. When the number of links is 4, the number of links is 29.

Figure 11 shows the number of link changes from static topology to optimized topology. We can obtain a conclusion similar to Figure 10, that is, an increase in links and satellites will affect the number of link changes. Both are positively related. By comparing Figures 10 and 11, it can be seen that the number of link changes in the static topology is more than the number of link changes in the initial topology, which may be due to the large gap between the static topology and the optimal topology. Link topology switching is frequent, and a static topology is not suitable for intersatellite networks.

Figure 12 shows the effect of link disruption on delay. Specifically, one link is interrupted at the 0th interval. Two links are interrupted at the 30th interval. Three links are interrupted at the 60th interval. All interrupted links are recovered at the 90th interval. The simulation results show that the optimal topology has better delay performance. Compared with the initial topology, the maximum delay and the average delay are lower, and the delay variance is also smaller.

Figure 13 shows the effect of satellite disruption on delay. Specifically, one satellite is interrupted at the 0th interval. Two satellites are interrupted at the 30th interval. Three satellites are interrupted at the 60th interval. All interrupted satellites are recovered at the 90th interval. The simulation results show that the optimal topology has better delay performance. Compared with the initial topology, the maximum delay and the average delay are lower, and the delay variance is also smaller. Satellite disruption has a greater impact on delay performance than link disruption. It can be seen in Figures 12 and 13 that the optimal link topology is not easily affected by link disruption and satellite disruption.

As shown in Table 1, the intersatellite link assignment algorithm based on reinforcement learning (RL) is compared with the two other methods. This is shown from three aspects: convergence iteration, cost time,
and delay performance. The three methods use the same hardware platform and are written in the Python language. They are interpreted and executed by the same version of the interpreter. The simulation scenario contains 12 satellites, and each satellite has 3 links. The quantum evolutionary (QE)\(^{19}\) method is an improvement of the genetic algorithm. Compared with the traditional algorithm, due to the cross operator and the mutation operator with quantum characteristics, it can prevent prematurity and make the algorithm converge faster. It is an excellent and representative evolutionary algorithm. The artificial bee colony (ABC)\(^{20}\) algorithm can consider global search and local search and achieve a good balance between solution research and solution selection. It is also an irreplaceable intelligent computing method. The abbreviations in the left column of Table 1 denote the following: KeyPara is the abbreviation of key parameter, ConvergeIter represents the convergent iteration, CostTime denotes the cost time, and DelayPerf stands for average delay performance. The key parameters of reinforcement learning include the greedy factor \((Gr = 0.76)\), discount factor \((DC = 0.9)\), and learning rate \((SS = 0.8)\). The key parameters of quantum evolution include the population number \((PN = 200)\), chromosome length \((CL = 132)\), and the angle of rotating mutation \((RM = 0.1\pi)\). The key parameters of the ABC include the maximum number of unimproved iterations \((MUI = 10)\) and the number of honey sources \((NHS = 100)\). It can be concluded from Table 1 that reinforcement learning has better average delay performance than two other algorithms, but the cost time is greater. Due to the human-like selection action of reinforcement learning, the number of iterations is much more than two other algorithms.

It must be noted that our algorithm-based reinforcement learning is easily affected by the calculation scale, mainly because human-like actions are too frequent, too much environment state information needs to be stored, and the cost of storage resources is very large, which indirectly lead to the small number of experimental simulation satellites. Although there are still some shortcomings in using reinforcement learning to solve the link assignment problem, we can still see some incomparable advantages of reinforcement learning. The algorithm proposed in this article is essentially a local search method, which can decompose a large-scale solution problem into several stage subproblems, and adjust the parameter setting to solve any scale problem. It is a process of constantly approaching the solution, and its randomness is much smaller than the random search method, so the algorithm is easier to converge and stable. Not only that, reinforcement learning algorithm can also generate selection action sequence. Its process of action selection is a human-like operation,
and humans can explain it. Moreover, humans can directly participate and act as subjective guide in the process of selection, which has strong practical significance in satellite link assignment, which is not available in other methods. In addition, the reinforcement learning method is a kind of highly extensible method. This article only tries to break the ice to verify that the value-based reinforcement learning method can solve this kind of problem. Later, researchers can apply more reinforcement learning methods to this field, including representatives suitable for solving complex problems such as AlphaGo Zero.\(^2\)

**Conclusion**

In this article, the reinforcement learning method was used to solve the link assignment problem. The link assignment problem was modeled as the decision optimization of the link selection action sequence. The number of antenna beams, network connectivity, and intersatellite visibility were considered as constraints. The constellation delay performance in a period was taken as the optimization objective. Combination optimization was transformed into Markov optimal decision sequence optimization, in which reinforcement learning is suitable for solving. It not only provides a new modeling method for link assignment but also provides a set of specific solutions.

The proposed method is an attempt to solve link assignment by reinforcement learning, but there are still many deficiencies, which mainly focus on two aspects: one is to use the idea of Q-learning to solve problems, but the scale of Q-learning will greatly increase with the expansion of the problem scale, which will affect the optimization results; the second is that the number of satellites and links in simulations is too small to meet the simulation requirements of a large number of satellites and links in LEO constellation.

Future work will focus on two aspects: one is to improve the proposed reinforcement learning model or attempt another reinforcement learning model for more complex optimization; the second is to increase the number of satellites and links in the simulation and improve it closer to the real LEO environment.

**Acknowledgements**

The authors thank the anonymous reviewers for their feedback on the paper.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is funded by the National Key R&D Program of China (no. 2018YFB1800303), the NSFC Project (no. 61872211 and U21A2045), the Science and Technology Planning Project of Jilin Province (grant no. 20180414024GH), and the Science and Technology Projects of Jilin Provincial Education Department (13th Five-Year Plan; grant no. JJKH20200795KJ).

**ORCID iD**

Weiwu Ren https://orcid.org/0000-0002-3787-636X

**References**

1. Iridium project, https://www.iridium.com (1996, accessed 30 May 2021).
2. Globalstar project, https://www.globalstar.com/en-us/corporate/home (2000, accessed on 30 May 2021).
3. OneWeb system, https://www.oneweb.world (2018, accessed 30 May 2021).
4. Starlink program, https://www.starlink.com/ (2019, accessed 30 May 2021).
5. Hongyun project, http://www.casic.com.cn (2018, accessed 30 May 2021).
6. Chang HS, Kim BW, Lee CG, et al. FSA-based link assignment and routing in low-earth orbit satellite networks. *IEEE T Veh Technol* 1998; 47: 1037–1048.
7. Yan HC, Zhang Q and Sun Y. Link assignment problem of navigation satellite networks with limited number of inter-satellite links. *Acta Aeronaut Astronaut Sin* 2015; 36: 2329–2339.
8. Dong MJ, Lin BJ, Liu YC, et al. Topology dynamic optimization for inter-satellite laser links of navigation satellite based on multi-objective simulated annealing method. *Chin J Laser* 2018; 45: 0706004.
9. Liu BY, Wang LQ and Zhu WG. Link assignment algorithm research on earth-moon spatial information network. *J Deep Space Explor* 2019; 6: 553–560.
10. Wang S, Wang Q, Yin H, et al. Research on inter-satellite link network routing algorithm based on multi-objective optimization. In: *Proceedings of the China satellite navigation conference (CSNC)*, Chengdu, China, 22–25 November 2020, pp. 702–712. Berlin: Springer.
11. Wang DH. *Research on networking of navigation inter-satellite link oriented the optimization of ranging and communication*. Master’s Thesis, National University of Defense Technology, Changsha, China, 2014.
12. Liu Z, Guo W, Deng C, et al. Perfect match model based link assignment for optical satellite network. In: *Proceedings of the 2014 IEEE international conference on communications (ICC)*, Sydney, NSW, Australia, 10–14 June 2014, pp. 4149–4153. Piscataway, NJ: IEEE.
13. Shi LY, Xiang W and Tang XM. A link assignment algorithm applicable to crosslink ranging and data exchange for satellite navigation system. *J Astronaut* 2011; 32: 1971–1977.
14. Zhou ZH. Research of inter-satellite link assignment of LEO satellite network. Master's Thesis, Nanjing University of Posts and Telecommunications, Nanjing, China, 2015.
15. Zhang T, Li J, Li H, et al. Ground resource optimization for navigation constellation based on satellite link scheduling. Eng Optimiz 2019; 51: 646–662.
16. Qiao L, Zhang H, Zhang R, et al. Multilayer satellite network topology design technology based on incomplete IGSO/MEO constellation. In: Proceedings of the international conference on space information network (SINC), Wuzhen, China, 19–20 September 2019. Singapore: Springer.
17. Wang Q, Hao Y and Cao J. Learning to traverse over graphs with a Monte Carlo tree search-based self-play framework. Eng Appl Artif Intel 2021; 105: 104422.
18. Kalakanti AK, Verma S, Paul T, et al. RL SolVeR Pro: reinforcement learning for solving vehicle routing problem. In: Proceedings of the 2019 1st international conference on artificial intelligence and data sciences (AiDAS), Ipoh, Malaysia, 19 September 2019. Piscataway, NJ: IEEE.
19. Yang S, Jiao L and Liu F. Quantum evolutionary algorithm. Chin J Eng Math 2006; 2: 235–246.
20. Ren W, Zhang B, Di X, et al. Density clustering anomaly intrusion detection algorithm based on ABC-DBSCAN. J Jilin Univ: Sci Ed 2018; 56: 95–100.
21. Schrittwieser J, Antonoglou I, Hubert T, et al. Mastering Atari, Go, chess and shogi by planning with a learned model. Nature 2020; 588: 604–609.