Reinforcement Learning Based Capacity Management in Multi-Layer Satellite Networks

Chunxiao Jiang, *Senior Member, IEEE*, and Xiangming Zhu

***Abstract*— The development of satellite networks is drawing much more attention in recent years due to the wide coverage ability. Composed of geosynchronous orbit (GEO), medium earth orbit (MEO), and low earth orbit (LEO) satellites, the satellite network is a three-layer heterogeneous network of high complexity, for which comprehensive theoretical analysis is still missing. In this paper, we investigate the problem of capacity management in the three-layer heterogeneous satellite network. We first construct the model of the network and propose a low-complexity method for calculating the capacity between satellites. Based on the time structure of the time expanded graph, the searching space is greatly reduced compared to traditional augmenting path searching strategies, which can significantly reduce the computing complexity. Then, based on Q-learning, we proposed a long-term optimal capacity allocation algorithm to optimize the long-term utility of the system. In order to reduce the storage and computing complexity, a learning framework with low-complexity is constructed while taking the properties of satellite systems into account. Finally, we analyze the capacity performance of the three-layer heterogeneous satellite network and also evaluate the proposed algorithms with numerical results.**

***Index Terms*— Capacity management, reinforcement learning, multi-layer networks, satellite networks.**

I. Introduction

With the rapid development in technology, terrestrial communication networks now can provide broadband service for mobile users worldwide [1], [2]. However, due to the construction cost of network facilities, terrestrial cellular networks are still unavailable in many less-developed areas or areas of low population density. Different from terrestrial base stations (BSs), the wide coverage of the satellite comes to be a rescue to provide extra services for these users [3]–[6]. Thus the development of satellite networks is drawing much more attention in recent years, and the concept of satellite internet is proposed to enable ubiquitous access all over the world [7], [8]. According to the height of the operating orbit, the satellite around the earth can be classified into the GEO satellite, the MEO satellite and the LEO satellite, which are of different capabilities and functionalities. Composed of the three types of satellites, the satellite network is a heterogeneous multi-layer system of high complexity [9]. Thus studying and exploring the network properties is of great importance to provide theoretical support for both construction and operation of future satellite networks.

While traditional single-layer networks have been well studied in the past, remarkable theoretical developments of multi-layer networks remain lacking until recently [10], and there are still many challenging issues that have not been solved. However, most networks in real-world are in fact composed of multi-type nodes or edges, which cannot be simply modeled by single-layer networks [11]. Thus the research on multi-layer networks now comes to be an important part of network theory. In [12], the authors gave a comprehensive discussion of the history of multi-layer networks. Then, a generalized definition of multi-layer networks was introduced, as well as the basic theories and tools for further study of multi-layer networks. The transportation network is a typical example of multi-layer networks in real-world. In [13], based on real data of the public transport in the United Kingdom, the multi-modal property of the transport system was investigated, in which different types of transport modes formed a multi-layer transport network.

The communication network is also one type of typical multi-layer networks. Composed of the three types of satellites, the satellite network is a three-layer heterogeneous network. Furthermore, since MEO and LEO satellites move around the earth periodically, the edges in satellite network will keep on changing over time. Thus the satellite network is in fact a multi-layer network of two aspects [12], in which the first aspect is the node type and the second aspect is time. To depict the dynamic properties of networks in time, the model of time varying graph (TVG) is proposed [14], and has been applied to the study of various dynamic networks. In [15], based on the model of TVG, the dynamic topology of the hybrid satellite network was modeled by snapshots, and a genetic algorithm was proposed to optimize the gateway placement problem. Similarly, in [16], the routing problem in satellite networks was solved by particles swarm methods, while considering the system as a series of static snapshots.

Manuscript received December 2, 2019; revised January 27, 2020 and February 25, 2020; accepted April 2, 2020. Date of publication April 14, 2020; date of current version July 10, 2020. This work was supported by the National Nature Science Foundation of China under Grant 61922050 and Grant 91638205. The associate editor coordinating the review of this article and approving it for publication was C. Huang. *(Corresponding author: Chunxiao Jiang.)*

Chunxiao Jiang is with the Tsinghua Space Center, Tsinghua University, Beijing 100084, China, and also with the Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084, China (e-mail: jchx@tsinghua.edu.cn).

Xiangming Zhu is with the Space Network Research Center, Zhejiang Laboratory, Hangzhou 311121, China, and also with the Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084, China (e-mail: zhuxm@zhejianglab.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2020.2986114

In wireless communication networks, capacity performance is one of the key properties [17]. However, the model of TVG does not work well in calculating the capacity of satellite networks, because it neglects the relation between different topologies. For delay-tolerant services in satellite networks, the transmission may consist of several forward processes before arriving at the destination, and the whole transmission process may experience different topologies across time [18]. Thus the capacity calculation in satellite networks should consider the correlation of different topologies, instead of analyzing each snapshot separately. In order to maintain the correlation of time in dynamic networks, the model of time expanded graphs is proposed [19], in which the network is replicated across time to formulated an expanded graph. Then, the max-flow algorithm for calculating capacity in static networks can be applied [20]. To reduce the time complexity and space complexity, the time aggregated graph model is then proposed [21], in which the dynamic properties of the network are modeled by aggregated values of edges. Due to the multi-layer properties of the satellite network in both the node aspect and time aspect, calculating the capacity of the satellite network is of high complexity. Comprehensive theoretical analysis of the capacity performance and the network constructions is still missing. In [22], the capacity of a two-layer satellite network was approximately obtained by calculating both the upper bound and the lower bound, but accurate capacity analysis remained lacking.

Generally, the total transmission capacity between satellites is shared by multiple users of the satellite. Since the transmission capacity is limited, the satellite needs to allocate the capacity among the users according to their communication demands. To optimize the long-term utility of the capacity allocation problem, we need to learn the long-term influence of different capacity allocation strategies at each time. Recently, machine learning is widely used in various areas due to its adaptability and ability of learning in dynamic and complex systems [23]–[26]. In this paper, we use Q-Learning to solve the dynamic capacity allocation problem, which is a type of reinforcement learning proposed by Watkins [27] and Watkins and Dayan [28], and has been applied in many different communication systems for resource allocation [29]–[33]. In [29], the author investigated the joint computing resource allocation problem in satellite-terrestrial networks. A Q-learning based resource allocation algorithm was proposed to improve the resource efficiency. In [30], Q-learning was used to allocate time-slot resource in unmanned aircraft systems. The long-term transmitted data size was maximized while considering the changing of environment and demands. In [31], the author investigated the resource allocation problem in fog radio access network, and proposed a Q-learning based algorithm to guarantees efficient utilization of the constraint resource while taking long-term utility into account. In [32], a Q-learning based distributed power allocation algorithm was proposed to enable ongoing transmit power adaptation in dense heterogeneous networks. Then, in [33], the author investigated the dynamic resource allocation problem in unmanned aerial vehicles networks. A Q-learning based algorithm was proposed to maximize the long-term rewards.

In this paper, we investigate the problem of capacity management in the three-layer heterogeneous satellite network based on reinforcement learning. Considering the node type and time, we model the satellite network as a multi-layer network of two aspects, based on which we construct the capacity model from any source satellite to any destination satellite. Then, we propose two algorithms for capacity calculation and capacity allocation separately to optimize the system performance. The contributions of this paper are summarized as follows:

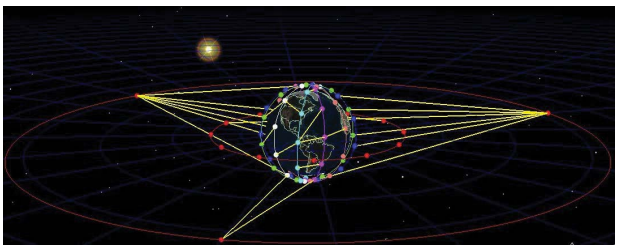
- We propose the capacity model for the three-layer heterogeneous satellite network. Considering both the node type and time, the satellite network is modeled as a multi-layer network of two aspects. Then, by calculating both of the continuous transmission process and the store-and-forward transmission process, the capacity between any two satellites is derived for any given time period.
- We propose a time structure based augmenting path searching method for calculating the capacity in the three-layer heterogeneous satellite network. Based on the time structure of the time expanded graph, the searching space is greatly reduced compared to traditional augmenting path searching strategies, which can significantly reduce the computing complexity.
- We propose a long-term optimal capacity allocation algorithm based on Q-learning to optimize the long-term utility in the three-layer heterogeneous satellite systems. In order to reduce the storage and computing complexity, we construct a learning framework with low-complexity while taking the properties of satellite systems into account.
- We analyze the capacity performance of the three-layer heterogeneous satellite network with numerical results. Based on the model and algorithms proposed, we investigate the key parameters of the network for intra-layer capacity and inter-layer capacity separately, and also analyze the trade-off when designing real systems.

The rest of this paper is arranged as follows. In the next Section, we introduce the network model and capacity model of the three-layer heterogeneous satellite network. Then, we give the capacity calculation algorithm and the long-term optimal capacity allocation algorithm in Section III and Section IV separately. The numerical results and analysis are shown in Section V. Finally, Section VI gives the conclusion.

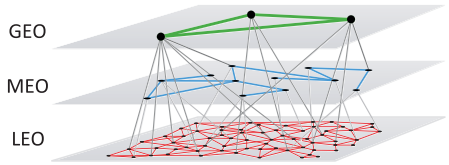
II. System Model

A. Network Model

With the development of satellite networks, various satellite constellation projects have been proposed to enable ubiquitous coverage for ground users, such as SpaceX, OneWeb, and O3b [7], [8]. The proposed satellite constellation projects include both LEO and MEO satellite constellations. Then, taking the existing GEO satellite systems into consideration, the three types of satellites will constitute a complex three-layer heterogeneous satellite network [9]. Studying and exploring the heterogeneous network properties is of great importance to provide theoretical support for both construction and operation



(a) Three-layer heterogeneous satellite network



(b) Abstracted three-layer graph

Fig. 1. Multi-dimensional satellite networks.

of future satellite networks. Thus we consider a three-layer heterogeneous satellite network as depicted in Fig. 1 (a), which consists of N_L LEO satellites, N_M MEO satellites, and N_G GEO satellites. Satellites of the same layer are connected by intra-satellite links (ISLs), while satellites of different layers are connected by inter-layer links (ILLs), and all links in the network are bidirectional. Abstracting satellites as nodes, and the links among satellites as edges, the network can be modeled by a multi-layer graph as $G = (V, E)$, as depicted in Fig. 1 (b), where

- $V = \{V^L, V^M, V^G\}$ is the set of all satellite nodes.
- $E = \{e(u, v)|, u \neq v\}$ is the set of all edges. $e(u, v) = 1$ represents that there exists a link between satellite u and v, while $e(u, v) = 0$ represents that there is no link between satellite u and v.
- $N_L^{ISL}, N_M^{ISL}, N_G^{ISL}$ are the average ISL numbers of each satellite. For example, N_L^{ISL} is the average ISL number between one LEO satellite and other LEO satellites. For $N_L^{ISL} = 4$, it means that each LEO satellite is connected with other 4 LEO satellites by average.
- $N_{L,M}^{ILL}, N_{L,G}^{ILL}, N_{M,G}^{ILL}$ are the average ILL numbers of each satellite. For example, $N_{L,M}^{ILL}$ is the average ILL number between one LEO satellite and other MEO satellites. Also, since links are bidirectional, all ILL numbers can be determined by the three variables.

Due to the limitation of payload capacity and cost [34], we consider that only part of the satellites have ILLs. Then, the ratios of the satellites that have ILLs are defined by $\gamma_{L,M}, \gamma_{L,G}, \gamma_{M,G}$, where $\gamma_{L,M} = 0.2$ represents that 20% of the LEO satellites have ILLs with MEO satellites. In this case, $N_{L,M}^{ILL}, N_{L,G}^{ILL}, N_{M,G}^{ILL}$ are the average ILL numbers for satellites that have ILLs.

As discussed above, the satellite network is a multi-layer network of two aspects, in which the first aspect is the node type and the second aspect is time. Since MEO and LEO satellites move around the earth periodically, the edges in satellite network will keep on changing over time. Consider there are total N_T topologies during time T, and the time duration of the tth topology is T_t, where the constraint $\sum_{t=1}^{N_T} T_t = T$ is naturally satisfied. Then the edges in the satellite network are redefined as

$$E = \{e(u, v, t)|, u \neq v, t \in [1, N_T]\} \tag{1}$$

Considering both the node type and time, the satellite network is modeled as a multi-layer network of two aspects, which consists of $N = N_L + N_M + N_G$ nodes and $N(N-1)N_t$ edges.

B. Capacity Model

In wireless communication networks, capacity performance is one of the key properties [17]. Different from terrestrial networks, delay-tolerant data transmission is common and important in satellite networks [35], [36], for which a certain amount of delay is tolerable and pre-resource management on the ground is possible. In this paper, we define the capacity as the maximum data that be transmitted from source satellite v_S to destination satellite v_D during time period $[t_0, t_1], t_0, t_1 \in [1, N_T]$. Generally, there are two types of data transmission process in the network: the continuous transmission process and the store-and-forward transmission process [19]. In the continuous transmission process, there exists a continuous path from the source node to the destination node, and the whole transmission process occurs during the same topology. As shown in Fig. 2, from source node 1 to destination node 4, the transmission process 1-2-4 at time 1 and the transmission process 1-4 at time 3 are both the continuous transmission process. However, in Fig. 2, there is no continuous path from node 1 to node 6 for all topologies. Node 1 needs to first transmit the data to node 4 at time 3. The node 4 stores the data at time 3, and then transmits the data to node 6 at time 4. Different from the continuous transmission process, the store-and-forward transmission process covers several different topologies. The intermediate nodes need to store the data, and then forward the data after the topology changing.

Let $R(u, v)$ be the mean transmit rate of the link between satellite u and v. The maximum data that can be transmitted from u to v during the tth topology can be calculated by

$$c(u, v, t) = e(u, v, t)R(u, v)T_t, \quad \forall u \neq v, t \in [1, N_T], \tag{2}$$

where $c(u, v, t) = 0$ represents that there is no transmission capacity between satellite u and v during the tth topology. Then the capacity from v_S to v_D during time $[t_0, t_1]$ can be defined as

$$\begin{aligned} C(v_S, v_D, t_0, t_1) &= \max \sum_{u \in V, u \neq v_S, t \in [t_0, t_1]} f(v_S, u, t) \\ &= \max \sum_{u \in V, u \neq v_D, t \in [t_0, t_1]} f(u, v_D, t) \end{aligned}$$

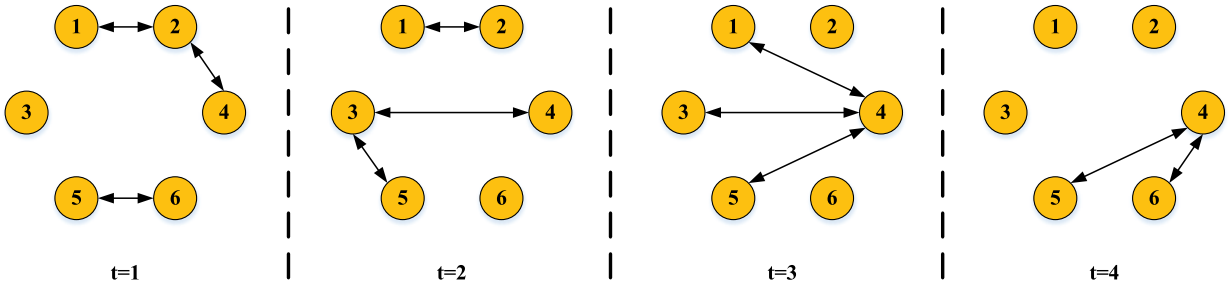


Fig. 2. Time varying graph.

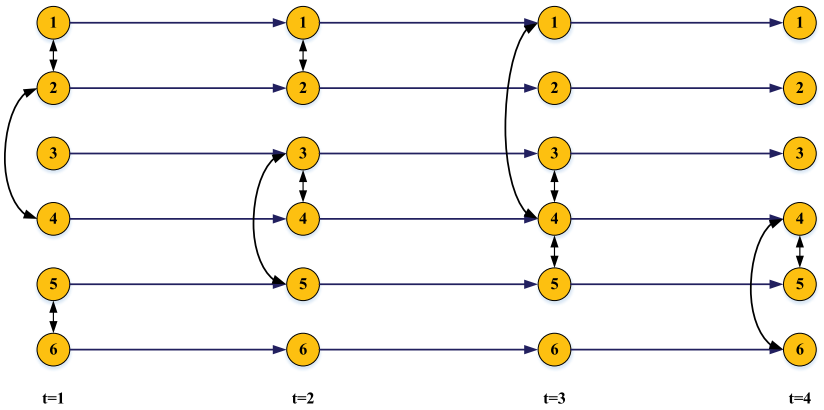


Fig. 3. Time expanded graph.

$$
\begin{aligned}
&C1:\ 0\le f(u,v,t)\le c(u,v,t),\quad \forall u,v\in V, u\neq v, t\in[t_0,t_1],\\
&C2:\ \sum_{u\in V,u\neq v,t\in[t_0,t_1]} f(u,v,t)\\
&\quad=\sum_{u\in V,u\neq v,t\in[t_0,t_1]} f(v,u,t),\quad \forall v\in V, v\notin\{v_S,v_D\},
\end{aligned}
\tag{3}
$$

where $f(u,v,t)$ is the data transmitted from node u to node v during the tth topology, C1 is the link capacity constraint, and C2 is the flow balance constraint. For any node except the source node and the destination node, the data out of the node should be equal to the data into the node during the whole period. With constraint C2, it can be guaranteed that all the data out of v_S will finally flow into v_D.

III. Low-Complexity Capacity Calculation Algorithm

A. Capacity Calculation

To calculate the capacity from source node v_S to destination node v_D, we need to consider all the possible transmission processes in the network, and find the transmission scheme that achieves the maximum capacity. For the continuous transmission process, the network topology remains static during the whole transmission process. In this case, the capacity calculation problem is equal to the max flow problem in static graphs, which can be solved by methods like Edmond-Karp (EK) algorithm [37]. The basic idea of the EK algorithm is to search augmenting paths from the source node to the destination node iteratively in the static graph until there is no new path. However, the EK algorithm cannot be applied directly to the store-and-forward transmission process, since the network topology changes during the transmission process. To solve this problem, we extend the network model to the time expanded graph by introducing the storage edge, in which the network is replicated across time to formulate an expanded graph [19]. As shown in Fig. 3, the copies of the same node of different time are connected by the storage edge. The data flow along the storage edge represents the data storage process over time. For example, the edge from node 1 at time 1 to node 1 at time 2 in fact does not exist in the real network. Instead, it means that the data is stored at node 1 from time 1 to time 2. Based on the time expanded graph, the capacity calculation problem can be redefined as

$$
\begin{aligned}
&C(v_S,v_D,t_0,t_1)\\
&\quad=\max[f(v_S,t_0,t_0+1)+\sum_{u\in V,u\neq v_S} f(v_S,u,t_0)]\\
&\quad=\max[f(v_D,t_1-1,t_1)+\sum_{u\in V,u\neq v_D} f(u,v_D,t_1)]\\
&C1:0\le f(u,v,t)\le c(u,v,t),\\
&\qquad\forall u,v\in V,u\neq v,t\in[t_0,t_1],\\
&C2:f(v,t,t+1)\le c(v,t),\quad \forall v\in V,t\in[t_0,t_1-1],\\
&C3:f(v,t-1,t)+\sum_{u\in V,u\neq v} f(u,v,t)\\
&\qquad=f(v,t,t+1)+\sum_{u\in V,u\neq v} f(v,u,t),\\
&\qquad\forall v\in V,t\in[t_0,t_1],\{v,t\}\notin\{\{v_S,t_0\},\{v_D,t_1\}\},
\end{aligned}
\tag{4}
$$

where $f(v,t,t+1)$ is the data stored in node v from time t to $t+1$, $c(v,t)$ is the storage capacity of node v, C2 is the storage capacity constraint, and C3 is the new flow balance constraint considering the storage edges. For any node except the source node at time t_0 and the destination node at the time t_1, the data out of the node at time t plus the data stored from time t to $t+1$ should be equal to the data into the node at time t plus the data stored from time $t-1$ to t. Also, since the storage capacity is generally much larger than the link capacity, the constraint C2 is assumed to be naturally satisfied in this paper. In the path searching problem (4), the objective is to find the maximum data flow from source node v_S to destination node v_D, while considering all the possible paths from v_S to v_D. The design variables are the data flow values of all the edges in the network.

By connecting the topologies of different time with storage edges, the dynamic network is transferred into a static expanded graph. Then in the expanded graph, the EK algorithm can be applied to compute the capacity from any source node v_S to any destination node v_D. The basic idea of the EK algorithm is to find augmenting paths from v_S to v_D iteratively until there is no new path. Let $cl(u,v,t) = c(u,v,t) - f(u,v,t)$ be the residual capacity of the link from u to v at time t. Then the augmenting path is a path from v_S to v_D, where the residual capacities of all edges in the path are non-zero. Considering the augmenting path starts from v_S at time t_0 and reaches v_D at time t_1, all the nodes in the augmenting path can be defined as

$$\{v_S, v^A_{1,t_0}, v^A_{2,t_0}, \dots v^A_{N_{A,t_0},t_0}, \dots, v^A_{1,t}, v^A_{2,t}, \dots, v^A_{N_{A,t},t}, \dots, v^A_{1,t_1}, v^A_{2,t_1}, \dots v^A_{N_{A,t_1},t_1}, v_D\}, \quad (5)$$

where $v^A_{n,t}$ is the nth node passed during time t, $N_{A,t}$ is the number of nodes passed during time t, and $v^A_{N_{A,t},t} = v^A_{1,t+1}$ represents the storing process over time. Then the capacity from v_S to v_D will be added by ε, where

$$\varepsilon = \min\{cl(v_S, v^A_{1,t_0}, t_0), \dots, cl(v^A_{i,t}, v^A_{i+1,t}, t), \dots, cl(v^A_{N_{A,t_1},t_1}, v_D, t_1) | t \in [t_0, t_1], i \in [1, N_{A,t}-1]\} > 0. \quad (6)$$

Updating the data flow and residual capacity for all edges in the path by $f(u,v,t) \leftarrow f(u,v,t) + \varepsilon$ and $cl(u,v,t) \leftarrow cl(u,v,t) - \varepsilon$, then the total capacity can be obtained by iteratively searching augmenting paths from v_S to v_D until there is no new path.

B. Time Structure Based Augmenting Path Searching

By applying the EK algorithm to the time expanded graph, the capacity from any node v_S to node v_D can be calculated. The main complexity of the EK algorithm comes from the augmenting path searching process. To find the augmenting path from v_S to v_D, it needs to traverse all the possible edges in the expanded graph. Since the network is replicated across time, the expanded graph tends to be of large scale, which significantly increases the computational complexity. On the other hand, the large number of nodes in the expanded graph are in fact replicated from the same set of nodes across time. As shown in Fig. 3, the nodes in each row are in fact the same node of different time. Taking this time structure into account, the augmenting path searching process can be improved by the constraint as follows

$$\text{If } \exists f(v^A_{i,t}, v^A_{i+1,t}, t), \quad t \in [t_0, t_1], \ i \in [1, N_{A,t}-1],$$
$$\text{then } \forall f(v^A_{j,\tau}, v^A_{j+1,\tau}, \tau), \tau > t \vee j > i \Rightarrow v^A_{j+1,\tau} \neq v^A_{i,t}. \quad (7)$$

For any augmenting path as in (5), once the data is transmitted from node $v^A_{i,t}$ to node $v^A_{i+1,t}$ at time t, the data will never be retransmitted to node $v^A_{i,t}$ for all the following transmission processes.

Proof: If $\exists v^A_{j+1,\tau} = v^A_{i,t}$ satisfying the following constraint

$$\exists f(v^A_{i,t}, v^A_{i+1,t}, t) \wedge \exists f(v^A_{j,\tau}, v^A_{j+1,\tau}, \tau) \wedge (\tau > t \vee j > i), \quad (8)$$

it means that the data is transmitted out of $v^A_{i,t}$ at time t and transmitted back to $v^A_{i,t}$ at time τ.

Then we can delete the nodes $\{v^A_{i+1,t}, \dots, v^A_{j,\tau}\}$ and also the relative edges $\{f(v^A_{i,t}, v^A_{i+1,t}, t), \dots, f_A(v^A_{j,\tau}, v^A_{j+1,\tau}, \tau)\}$ from the original augmenting path, and connect $v^A_{i,t}$ with $v^A_{j+1,\tau}$ across time by storage edges $\{f(v^A_{i,t}, t, t+1), \dots, f(v^A_{i,t}, \tau - 1, \tau)\}$. The capacity of the new augmenting path can be calculated by

$$\begin{aligned}\varepsilon_2 &= \min\{cl(v_S, v^A_{1,t_0}, t_0), \dots, cl(v^A_{i-1,t}, v^A_{i,t}, t), \\ &\quad cl(v^A_{j+1,\tau}, v^A_{j+2,\tau}, t), \dots, cl(v^A_{N_{A,t_1},t_1}, v_D, t_1)\} \\ &\geq \min\{cl(v_S, v^A_{1,t_0}, t_0), \dots, cl(v^A_{i-1,t}, v^A_{i,t}, t), \\ &\quad cl(v^A_{i,t}, v^A_{i+1,t}, t), \dots, cl(v^A_{j,\tau}, v^A_{j+1,\tau}, t), \\ &\quad cl(v^A_{j+1,\tau}, v^A_{j+2,\tau}, t), \dots, cl(v^A_{N_{A,t_1},t_1}, v_D, t_1)\} = \varepsilon > 0.\end{aligned} \quad (9)$$

Since the edge set in the new augmenting path is the subset of the edge set in the original augmenting path, the capacity of the new augmenting path will be no less than the original augmenting path. We can then replace the original augmenting path with the new one, and obtain the equal or larger capacity. Thus the constraint (7) will always be satisfied to obtain the maximum capacity. ■

As shown in Fig. 4 (a), we aim to calculate the capacity from node 1 to node 6 in this expanded graph. Fig. 4 (b) gives a possible augmenting path from node 1 to node 6. However, the data is transmitted from node 4 to node 3 at time 2, and then transmitted back to node 4 from node 5 at time 3. By deleting the nodes and related edges from node 4 at time 2 to node 4 at time 3, and adding the storage path, we obtain the new augmenting path in Fig. 4 (c). As discussed above, the capacity of the new path is no less than the original path, and the original path can be replaced by the new one.

In the traditional augmenting path searching strategy, it needs to traverse all the possible edges in the expanded graph to obtain the maximum data flow. However, based on the time structure of the time expanded graph, the searching space can be greatly reduced using the constraint (7). Thus the time structure based searching strategy can significantly reduce the computational complexity for capacity calculation.

IV. Long-Term Optimal Allocation Algorithm

A. Capacity Allocation in Multi-Layer Satellite Networks

We now can calculate the total transmission capacity from any satellite u to v during any time t_0 and t_1, which is the

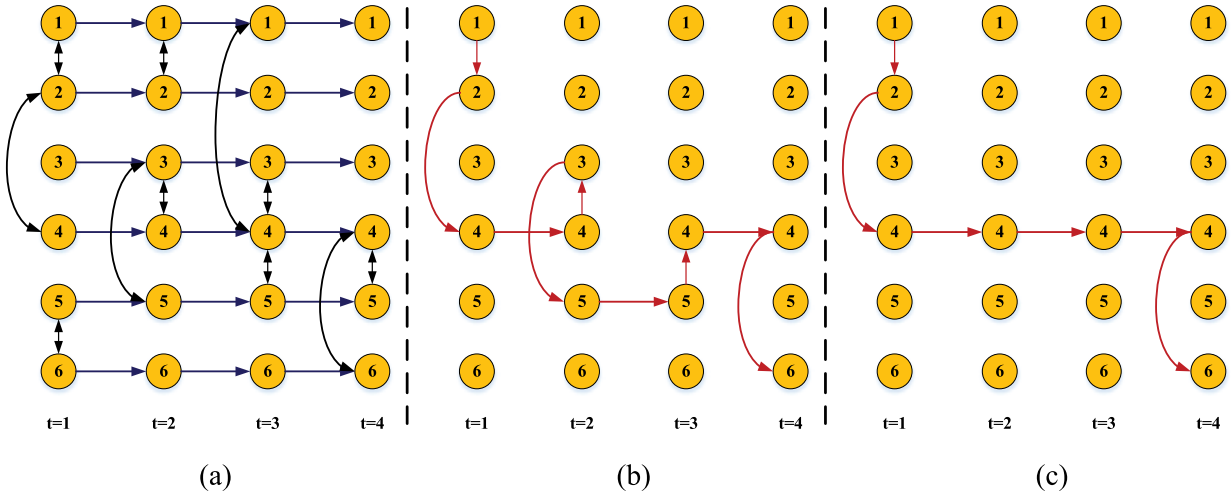


Fig. 4. Time structure based augmenting path searching.

maximum data that can be transmitted between the two satellites. Generally, the total transmission capacity from satellite u to satellite v is shared by multiple users of the satellite. Since the transmission capacity is limited, the satellite needs to allocate the capacity among the users according to their communication demands. In future satellite-terrestrial system, the satellite can be used to provide backhaul transmission for terrestrial BSs [38]. In this case, the satellite users are the BSs, and the flow management among different BSs is of great importance with limited network capacity. Also, for better system performance, the satellite users are assume to be equipped with multiple antennas, which are the new trend in wireless systems [39].

Considering the capacity allocation problem from satellite u to v during time t_0 and t_1, the total transmission capacity is $C(u, v, t_0, t_1)$, which can be calculated as above. Although all the data will reach the satellite v by time t_1, the transmission processes may start at different time. For each time t, the transmission capacity is $C(t)$, which means a total of $C(t)$ data can be transmitted out of satellite u at time t, and the $C(t)$ data will reach satellite v by time t_1. Then the constraint $\sum_{t=t_0}^{t_1} C(t) = C(u, v, t_0, t_1)$ is naturally satisfied. Also, for each time, the maximum data that can be transmitted out of satellite u is limited, which is determined by the ISLs and ILLs of the satellite. We use $C_{\max}$ to represent the maximum data that can be transmitted at any time t considering the actual network constraints, which is obtained based on the algorithm in Section III.

For satellite u, there are B users that need to transmit data to satellite v, and these users will share the total $C(u, v, t_0, t_1)$ capacity during the whole transmission process. The capacity demand of user b at time t is $c_b^{req}(t)$, and all users will report its own demand to the visible satellite when there is new transmission demand. To avoid the capacity demands of some users being larger than the limit $C_{\max}$, the capacity demand is restricted to be less than $c_{\max}^{req}$, where $c_{\max}^{req} \leq C_{\max}$. Also, since MEO and LEO satellites move around the earth periodically, the satellite will be invisible to the users periodically. We use $\tau_b(t) = \{0, 1, \ldots, \tau_{max}\}$ to represent the remaining visible time of user b, where $\tau_b(t) = 0$ means that the satellite is invisible to user b at time t, and the maximum visible time is τ_{max}. Also, we have

$$\tau_b(t+1) = \tau - 1, \quad \forall \tau_b(t) = \tau > 0. \tag{10}$$

For users with $\tau_b(t) = 1$, the satellite will be invisible in the next time slot. Due to the mobility of the satellite, the users can only transmit data via the satellite when $\tau_b(t) > 0$.

After receiving the capacity demands of all users, the satellite then allocates the total transmission capacity among the users. The capacity allocated to user b at time t is represented by $c_b^{rce}(t)$, and we define the utility function of user b at time t as

$$\begin{aligned} r_b(t) = &\min(c_b^{rce}(t), c_b^{req}(t)) \log_2(\tau_b(t)+1) \\ &-\lambda\varphi(\tau_b(t)) \max(0, c_b^{req}(t) - c_b^{rce}(t)), \\ \lambda \geq 0, \varphi(\tau_b(t)) = &\begin{cases} 1, & \tau_b(t) = 1 \\ 0, & \text{others.} \end{cases} \end{aligned} \tag{11}$$

The utility in (11) consists of two parts. The first part is the utility from the allocated capacity. This part of utility is linear to the capacity being allocated, and will be no more than the capacity demand of the user. Since there will be no utility for over-allocated capacity in (11), the system will learn to avoid over-allocated capacity based on the Q-learning process. The constraint $c_b^{rce}(t) \leq c_b^{req}(t)$ will be naturally satisfied based on the utility function design. Also, since the remaining visible time $\tau_b(t)$ of users is different, the utility obtained is related to the allocation time, and allocating the capacity to users with larger $\tau_b(t)$ will obtain larger utility. In the case of $\tau_b(t) = 0$, the satellite is invisible to the user. Then allocating capacity to this user will obtain no utility. The second part of the utility in (11) is the punishment for unsatisfied capacity demand. For users with $\tau_b(t) = 1$, the satellite will be invisible in the next time slot. Then, the capacity demand that has not been satisfied in this time slot will be discarded because of the invisibility. We introduce the punishment factor $\lambda \geq 0$

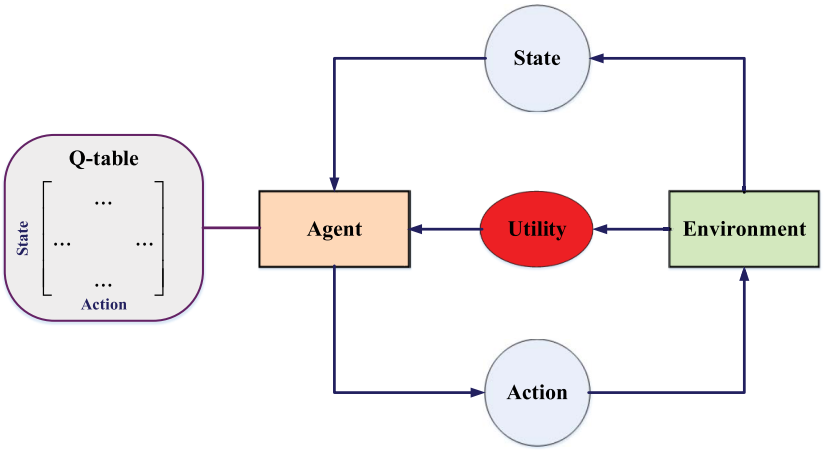


Fig. 5. The framework of Q-learning.

to control the weight of punishment in the utility function. If $\lambda = 0$, there is no punishment for unsatisfied capacity demand.

By adding up the utility of all users, the system utility at time t, and total utility during time t_0 and t_1 are

$$R(t) = \sum_{b=1}^{B} r_b(t),$$
$$R(t_0, t_1) = \sum_{t=t_0}^{t_1} R(t), \tag{12}$$

where $r_b(t)$ has been defined in (11). The satellite needs to determine the capacity among users to optimize the total system utility during time t_0 and t_1. However, if we simply optimize the utility of each time t, the total utility may not be optimal since the capacity allocation problem at time t is coupled with the subsequent problems after t. If we allocate large capacity to users with large value of τ_b at time t to maximize the utility at time t, we may obtain large punishment at time $t+1$ because of the unsatisfied capacity demand. For example, assuming that $\tau_1(t) = 2$ and $\tau_2(t) = 1$, if we allocate large capacity to user 1 to maximize the utility at time t, the capacity demand of use 2 will not be satisfied at time t. Then, at time $t + 1$, the satellite will be invisible for user 2, and there will be large punishment at time $t + 1$ because of the unsatisfied capacity demand of user 2. By introducing punishment for unsatisfied capacity in (11), always allocating capacity to users with larger τ_b will lead to large performance loss due to the unsatisfied capacity of users with smaller τ_b. The system needs to balance the capacity allocated among users with different τ_b to improve the system performance. Since the capacity allocation strategy at time t will influence the utility after time t, we need to consider the long-term utility when allocating the capacity at each time t. Thus the capacity allocation problem is a combined resource allocation and time scheduling problem, whose optimal solutions cannot be directly obtained. In the next section, we apply Q-learning to the system and find the capacity allocation strategy for long-term utility.

B. Structure of Q-Learning

To optimize the long-term utility of the capacity allocation problem, we need to learn the long-term influence of different capacity allocation strategies at each time. In this paper, we use Q-learning to solve the dynamic capacity allocation problem, which is a model free reinforcement learning method that does not require priori knowledge of the system model. Instead, Q-learning can automatically learns the optimal strategies of the system by interacting with the environment [26]. As depicted in Fig. 5, the framework of the Q-learning system consists of learning agent, environment, state, action, and the utility. The agent needs to learn the optimal action of each state by interacting with the environment in order to optimize the utility. The core of Q-learning is that the agent will maintain a Q-table, which records the Q-value of each action at each state. The larger the Q-value in the Q-table means that the action will bring lager utility at this state. After configuration of the Q-table, the optimal action at each state can be simply obtained by selecting the action with the largest Q-value. Thus the most important procedure in Q-learning is the configuration of the Q-table. The Q-table can be obtained by the updating equation as follows [40]

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha\{R(t) + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\}, \tag{13}$$

in which s_t is the current state, a_t is the selected action, $Q(s_t, a_t)$ is the Q-value for selecting action a_t at state s_t, $R(t)$ is the immediate utility for selecting action a_t at state s_t, s_{t+1} is the next state after selecting action a_t at state s_t, α is the learning factor, and γ is the discount factor for long-term utility. After initialization of the system, the agent can observe the current state of the system s_t. For the current state s_t, the agent selects an action a_t based on the action selecting strategy, which will be discussed later. Then, the agent will receive the immediate utility for selecting action a_t at state s_t, and the system will turn into the next state s_{t+1}. The agent then updates the Q-table based on the updating equation (13). By adjusting the learning factor α and discount

factor γ, we can control the learning speed and the weight of the long-term utility during the learning process. Through iteratively updating the Q-table as above, the agent will finally obtain the optimal Q-table $Q^*(s,a)$. Then the maximum long-term utility at state s_t is

$$V^*(s_t) = \max_{a_t} Q^*(s_t, a_t). \tag{14}$$

The optimal action at state s_t is

$$a^*(s_t) = \arg\max_{a_t} Q^*(s_t, a_t). \tag{15}$$

By means of Q-learning, we can obtain the optimal action for the system at any state without priori knowledge of the system model, especially for the system of high dynamic and complexity, which is difficult to be modeled.

We then apply Q-learning to the capacity allocation problem. Due to the limited computation and energy resources on the satellite, the resource management process is generally implemented on the ground in current satellite systems [41]. In the capacity allocation problem, the agent is the control station of the satellite on the ground, and the environment is the whole satellite system. The capacity allocation strategies of all the satellite are centralized at the control station. The environment states among the satellites will be acquired by the control channel at the control station. These environment states consist of only a small amount of data, which can be supported by the feeder link of the satellite. Generally, there are multiple feeder links for different satellite. Thus the transmission overhead will not be large. The control station will allocate the capacity of each time t to users after receiving the demands of users, while considering the current available capacity and the visible time of users. Thus the state s_t consists of the transmission capacity at time t, the capacity demands of users, and the visible time of users. The state at time t can be defined as

$$s_t = [C(t), c_1^{req}(t), \tau_1(t), c_2^{req}(t), \tau_2(t), \ldots, c_B^{req}(t), \tau_B(t)]. \tag{16}$$

The agent needs to allocate the transmission capacity of each time t among users, and the action at time t is the capacity allocation strategy. Then the action at time t can be defined as

$$a_t = [c_1^{rce}(t), c_2^{rce}(t), \ldots, c_B^{rce}(t)]. \tag{17}$$

The immediate utility obtained from action a_t is the utility obtained from satisfied capacity demand and also the punishment from unsatisfied demand, as defined in (11) and (12). In order to determine the optimal action at each time t, the agent will maintain a Q-table $Q(s_t, a_t)$ for each action a_t at each state s_t, and the optimal Q-table can be obtained by the updating equation in (13). After obtaining the optimal Q-table $Q^*(s_t, a_t)$, the capacity allocation strategy of each time t can be obtained by

$$a^*(s_t) = \arg\max_{a_t} Q^*(s_t, a_t). \tag{18}$$

The optimal long-term utility at each time t is obtained as

$$V^*(s_t) = \max_{a_t} Q^*(s_t, a_t). \tag{19}$$

C. Proposed Long-Term Optimal Capacity Allocation Algorithm

In order to construct the Q-table $Q(s_t, a_t)$, we need to maintain a Q-value for each action at each state. The size of the Q-table will be $N_s \times N_a$, where N_s is the number of states and N_a is the number of actions. As defined in (16), since the transmission capacity and the capacity demands at time t are continuous, the number of states will be infinite. To solve this problem, we normalize the transmission capacity and the capacity demands by the minimum allocation unit Δc. Then the normalized capacity can be defined as

$$\begin{aligned} \overline{C}_{\max} &= \left\lfloor \frac{C_{\max}}{\Delta c} \right\rfloor, \\ \overline{c}_{\max}^{req} &= \left\lceil \frac{c_{\max}^{req}}{\Delta c} \right\rceil. \end{aligned} \tag{20}$$

The possible values of transmission capacity $C(t)$ is $[0, 1, 2, \ldots, \overline{C}_{\max}]$, and the possible values of capacity demand $c_b^{req}(t)$ is $[0, 1, 2, \ldots, \overline{c}_{\max}^{req}]$. For each user b, the number of possible combinations is $(\overline{c}_{\max}^{req} + 1)\tau_{\max} + 1$, considering the different capacity demand and visible time, and the capacity demand can only be 0 if $\tau_b(t) = 0$. Taking all users into account, the number of the states of the system can be calculated as

$$N_s = (\overline{C}_{\max} + 1)((\overline{c}_{\max}^{req} + 1)\tau_{\max} + 1)^B. \tag{21}$$

We can observe that the number of states rapidly increases with the user number B, even if we normalize the capacity by a large allocation unit. For example, if we set $\overline{C}_{\max} = \overline{c}_{\max}^{req} = \tau_{max} = 2$, which are rather small, the number of states will still be more than 8×10^8 for only 10 users. In this case, the size of the Q-table will be too large. Maintaining both the storage and the Q-table will be difficult or even unrealizable.

To make the construction of the Q-table realizable, it is necessary to reduce the number of system states. In the above analysis, we treat each user as an individual. We need to record the different states of each user, and thus the number of states increases rapidly with the user number. In fact, from the perspective of the overall utility of the system, two users of the same normalized capacity demand and visible time can be treated as the same, since satisfying the capacity demand of either user will bring the same utility. Thus we do not need to record the different states of each user. Instead, we only need to record the number of user of each different state. To record the states of different capacity demands, we classify the capacity demand of users into $L_1 + 1$ levels, from c_0^{req} to $c_{L_1}^{req}$, in which

$$c_0^{req} = 0, c_l^{req} \in (\frac{c_{\max}^{req}}{L_1}(l-1), \frac{c_{\max}^{req}}{L_1} l]. \tag{22}$$

Then all users can be classified into $L_2 = (L_1 + 1)\tau_{\max} + 1$ types, and each type represents a different combination of capacity demand and visible time. Generally, larger value of quantization level L_1 means less loss of the details, but the storage and computation complexity will be higher. In performance evaluation, we show that small values of L_1 can achieve good performance, which will be discussed later. Then, the

state of the system s_t can be redefined as the number of users of different types, and we have

$$s_t = [C(t), n_1(t), n_2(t), \ldots, n_{L_2}(t)], \tag{23}$$

in which $n_l(t)$ is the number of users of type l at time t. Each possible state of the system is a possible classification of the B users into L_2 types. Then the number of states of the system can be calculated as

$$N_s = (C_{\max}+1)C_{B+L_2-1}^{L_2-1} = (C_{\max}+1)\frac{(B+L_2-1)!}{B!(L_2-1)!}. \tag{24}$$

Similarly, if we set $\overline{C}_{\max} = L_1 = \tau_{max} = 2$, the number of states will be only 2.4×10^4 for 10 users, which is much smaller compared with the 8×10^8 states above.

At each state, the agent needs to allocate the $C(t)$ capacity among the users. Similarly, we also normalize the allocated capacity by the minimum allocation unit Δc. Then the action at each state is to allocate the normalized capacity among L_2 types of users, and the number of actions at time t can be calculated as

$$N_a = C_{\overline{C}(t)+L_2-1}^{L_2-1} = \frac{(\overline{C}(t)+L_2-1)!}{(\overline{C}(t))!(L_2-1)!}, \tag{25}$$

where $\overline{C}(t) = \left\lfloor \frac{C(t)}{\Delta c} \right\rfloor$ is the normalized capacity at time t. We can observe that the number of actions is different for different value of $C(t)$. In this case, we need to maintain a Q-table for each possible value of $C(t)$, which significantly increases the complexity. Also, the number of actions will still be large for small value of Δc or large number of user types. Then, the size of Q-table will be large, and finding the action of maximum Q-value will be resource-intensive.

The large number of actions mainly comes from the combinatorial problem when allocating multiple capacity units among users. To reduce the complexity, we can decompose the allocation problem by allocating one unit of capacity Δc each time. Then we only need to choose one type of user from the L_2 types of users, and the number of actions will be reduced to only $N_a = L_2$. Then the action can be redefined as

$$a = [n_l]. \tag{26}$$

Since there can be multiple units of capacity in one time slot of the real time, one time slot may consist of multiple actions. When the agent chooses the action and turn into the next state, the time slot may be the same if there are still remaining capacity for this time slot. Thus we redefine the system state as

$$s = [C, c_1^{req}, \tau_1, c_2^{req}, \tau_2, \ldots, c_B^{req}, \tau_B], \tag{27}$$

which is independent of the time t. In (27), C is the current transmission capacity that can be allocated. During the capacity allocation process in the Q-learning algorithm, the allocated capacity cannot exceed the transmission capacity C. The updating equation can be rewritten as

$$Q(s,a) = Q(s,a) + \alpha\{R + \gamma \max_{a'} Q(s',a') - Q(s,a)\}, \tag{28}$$

where s' is the next state after selecting action a at state s. If the current transmission capacity C is more than one unit of capacity, the system will continue to allocate the transmission capacity at time t. The punishment in (11) will not be considered since the capacity allocation of this time slot will be continued in the next state. In this case, the utility function is

$$r_b = \min(c_b^{rce}, c_b^{req}) \log_2(\tau_b+1). \tag{29}$$

If the current transmission capacity C is only one unit of capacity, the system will turn to allocate the transmission capacity of time $t+1$, and the utility function is the same as (11). Since we allocate one unit of capacity for each action, the number of actions at any time slot is equal to the total transmission capacity of this time slot, which is no larger than the capacity constraint C_{max}.

The entire long-term optimal capacity allocation algorithm is summarized in Algorithm 1. At each state, the action is selected based on the ϵ-greedy strategy [33], which can avoid local optimum results by introducing a random variable ϵ while selecting the action. Since only one episode of learning will not find the optimal strategies, the learning process will be repeated until the utility converges or the maximum episode limit is reached. Also, the control station does not need to transmit the action result to the satellite by each action. Instead, the control station can calculate all the actions during time t_0 and t_1 on the ground base on Algorithm 1, and transmit the whole action results to the satellite. For burst traffic when there is a new user and capacity demand for satellite u, the control center can also recalculate all the actions during time t_0 and t_1, and transmit the whole action results to the satellite. The action results consist of only a small amount of data, which can be supported by the feeder link of the satellite. With multiple feeder links of different satellite, the transmission overhead will not be large.

V. Performance Evaluation

In this section, numerical results are provided to evaluate the proposed algorithms for both capacity calculation and allocation. We first analyze the capacity performance of the three-layer heterogeneous satellite network based on the capacity calculation algorithm. Then, the results of the proposed long-term optimal capacity allocation algorithm are discussed with full analysis.

A. Capacity Performance Analysis

To investigate the capacity performance of the multi-layer satellite network, we analyze the key parameters of the network for intra-layer capacity and inter-layer capacity separately. Then, combined with the implementation constraints, we discuss the trade-off when designing real systems, and find the preferred system architecture for better capacity performance. The basic setting of the network is shown in Tab. I. The number of LEO satellites is set as 66, referring to the Iridium satellite. The number of MEO satellites is set as 12, referring to the O3b network. The number of GEO satellites is set as 3, since 3 GEO satellites can cover the entire surface of the earth. R_L, R_M, R_G are the intra-layer transmit rate of each layer, while R_{LM}, R_{LG}, R_{MG} are the inter-layer transmit rate between layers. We consider there

Algorithm 1 Long-Term Optimal Capacity Allocation Algorithm

Initialize $N_{episode} = 1$, $N_{episode\_max}$, t_{max}, ϵ
Calculate the capacity $C(v_S, v_D, 1, t_{max})$ based on the algorithm in Section III
Initialize $Q(s, a)$ for all states and actions
repeat
  Initialize $t = 1$
  repeat
    Observe the current system state $s = [C, c_1^{req}, \tau_1, c_2^{req}, \tau_2, \ldots, c_B^{req}, \tau_B]$
    Generate a random value $\epsilon' \in [0, 1]$
    if $\epsilon' < \epsilon$ **then**
      Randomly select the action from all possible actions
    else
      Select the action with the maximum Q-value, $a = \arg\max_a Q(s, a)$
    end if
    if $C = \Delta c$ **then**
      $t = t + 1$
      Calculate the immediate utility R referring to (11)
    else
      Calculate the immediate utility R referring to (29)
    end if
    Update the Q-table referring to (28)
    The system turns to the next state s'
  until $(t > t_{max})$
  Calculate the system total utility of all time $R(1, t_{max})$ referring to (12)
  $N_{episode} = N_{episode} + 1$
until $R(1, t_{max})$ converges or $N_{episode} > N_{episode\_max}$

TABLE I
BASIC SETTING OF THE NETWORK

Parameter	Value	Parameter	Value	Parameter	Value
N_L	66	$N_{L,M}^{ILL}$	1	R_L	1
N_M	12	$N_{L,G}^{ILL}$	1	R_M	3
N_G	3	$N_{M,G}^{ILL}$	1	R_G	8
N_L^{ISL}	4	$\gamma_{L,M}$	0.2	R_{LM}	1
N_M^{ISL}	2	$\gamma_{L,G}$	0.2	R_{LG}	1
N_G^{ISL}	2	$\gamma_{M,G}$	0.2	R_{MG}	3

are total $N_T = 100$ topologies during time $T = 100$, and the time duration of each topology is 1.

Fig. 6 (a) shows the intra-layer capacity of LEO satellites with different LEO number and LEO ISLs. We can observe that the intra-layer capacity of LEO satellites decreases while the number of LEO satellite increases. Since the average ISL number of each satellite is fixed, increasing the total number of LEO satellites will lead to the decreasing of the paths from the source to the destination. For $N_L^{ISL} = 8$, when N_L increases from 10, almost full connection, to 100, the intra-layer capacity of LEO satellites decreases by about 10%. In addition, by increasing the average ISL number, the source node can transmit at a higher total rate with extra links, and the network is also connected more tightly. For $N_L = 10$, the capacity increases by about 450% when N_L^{ISL} increase from 2 to 8. However, the number of ISLs is generally limited, and more links means higher construction cost. We will discuss this later in the system design part.

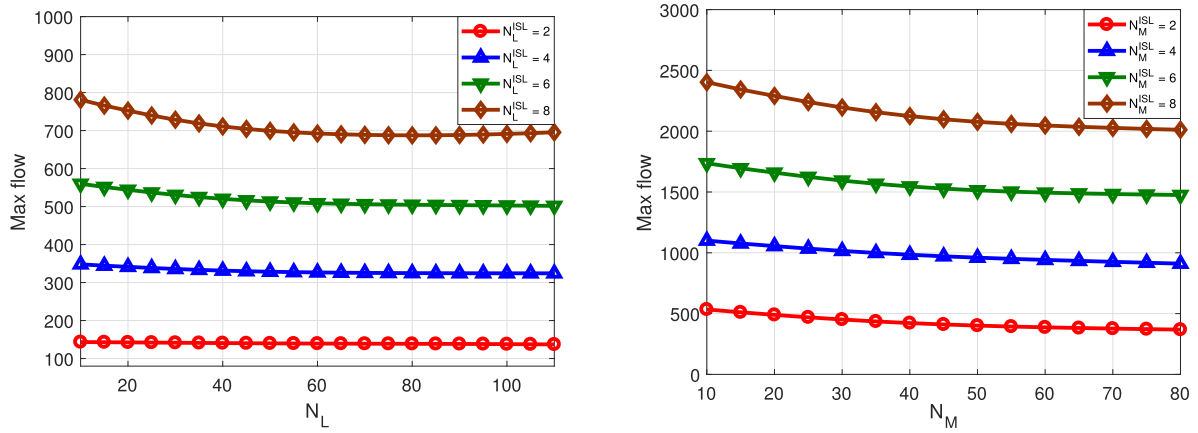
Fig. 6 (b) shows the intra-layer capacity of MEO satellites with different MEO number and MEO ISLs. Similarly, the intra-layer capacity decreases while the number of MEO satellite increases. However, the intra-layer capacity of MEO satellites decreases faster than LEO satellites. For $N_M^{ISL} = 8$, when N_M increases from 10 to 80, the intra-layer capacity of MEO satellites decreases by about 20%.

Fig. 7 shows the inter-layer capacity between LEO and MEO satellites with different LEO number and LEO ISLs. We can observe that the inter-layer capacity between LEO and MEO satellites first increases and then tends to be stable as N_L increases. The inter-layer capacity between LEO and MEO satellites mainly comes from the ILLs between LEO and MEO satellites. Increasing the LEO number will bring more ILLs, and thus the inter-layer capacity will first increase as N_L increases. For $N_L^{ISL} = 4$, the inter-layer capacity increases by 25% when N_L increases from 10 to 60. However, since the ISLs and ILLs of the source node and destination node are fixed, the main restriction of the inter-layer capacity will be the transmission capacity of the source node and destination node with large number of LEO satellites. For $N_L^{ISL} = 4$, the inter-layer capacity increases by 4% when N_L increases from 60 to 100. On the other hand, for $N_L^{ISL} = 8$, the inter-layer capacity increases by 10% when N_L increases from 60 to 100. With more ISLs, the increasing period will be longer as N_L increases.

In Fig. 6 and Fig. 7, we can observe that the intra-layer capacity and inter-layer capacity increase significantly with the number of ISLs. In order to investigate the trade-off between capacity gains and cost of ISLs, we introduce the variable of capacity ratio as follows:

$$C_L^R(N_L^{ISL}) = \frac{C(v_S, v_D, t_0, t_1)}{N_L^{ISL}}, \quad v_S, v_D \in V_L,$$
$$C_{LM}^R(N_L^{ISL}) = \frac{C(v_S, v_D, t_0, t_1)}{N_L^{ISL}}, \quad v_S \in V_L, v_D \in V_M. \quad (30)$$

Fig. 8 (a) gives the capacity ratio of intra-layer capacity of LEO satellites. We can observe that the capacity ratio increases with the LEO ISLs. For $N_L = 70$, the capacity ratio increases by 20% when N_L^{ISL} increases from 2 to 8. Also, we can observe that for the same number of ISLs, the capacity ratio decreases as the number of LEO satellites increases. For $N_L^{ISL} = 2$, the capacity ratio decreases by 10% when N_L increases from 10 to 70. Although more links brings higher construction costs, the intra-layer capacity of LEO satellites increases nonlinearly with the LEO ISLs, and larger value of N_L^{ISL} will bring higher capacity ratio. Thus increasing the LEO ISLs is the efficient method to increase the



(a) LEO intra-layer capacity

(b) MEO intra-layer capacity

Fig. 6. Intra-layer capacity with different satellite number and ISLs.

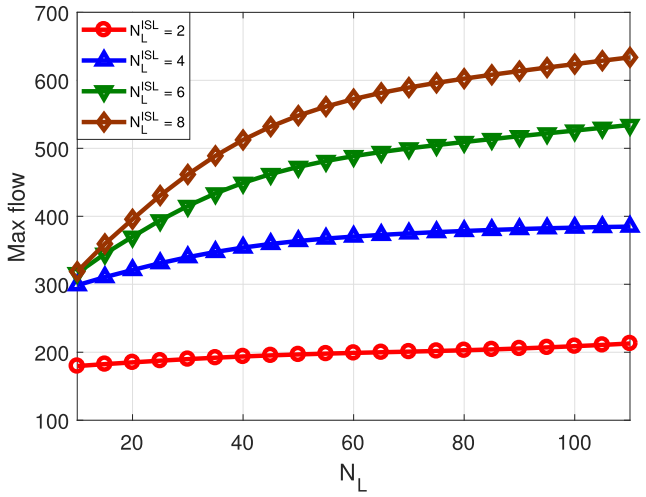


Fig. 7. Inter-layer capacity between LEO and MEO satellites with different LEO number and LEO ISLs.

intra-layer capacity of LEO satellites. On the other hand, larger number of LEO satellites will bring lower capacity ratio. This also needs to be taken into consideration when designing the network.

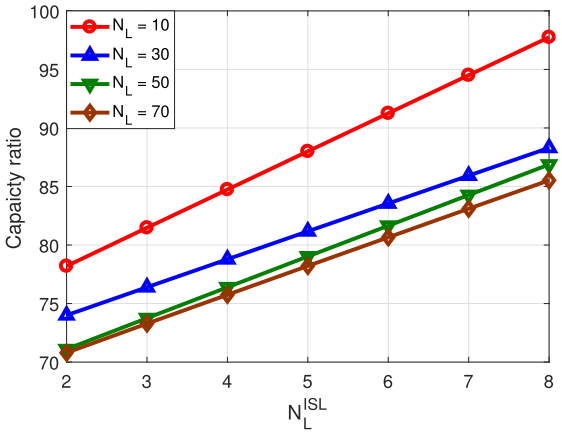
Fig. 8 (b) gives the capacity ratio of inter-layer capacity between LEO and MEO satellites. Different from the case of intra-layer capacity, the inter-layer capacity ratio decreases with the LEO ISLs. For $N_L = 70$, the capacity ratio decreases by 25% when N_L^{ISL} increases from 2 to 8. On the other hand, we can observe that the capacity ratio increases as the number of LEO satellites increases. For $N_L^{ISL} = 2$, the capacity ratio increases by 10% when N_L increases from 10 to 70. The variation tendency of the inter-layer capacity ratio between LEO and MEO satellites is completely opposite to the variation tendency of the intra-layer capacity ratio of LEO satellites. Thus the capacity ratio of intra-layer capacity and inter-layer capacity cannot both be satisfied when designing LEO parameters of the network.

Due to the construction costs, the ILLs are generally limited in the network. Considering that the total number of ILLs is constant, we investigate the configuration mode of ILLs in the network in Fig. 9, in which the two variables of $N_{L,M}^{ILL}$ and $\gamma_{L,M}$ satisfy $N_{L,M}^{ILL}\gamma_{L,M} = 1$. We can observe that the intra-layer capacity of LEO satellites significantly decreases as $N_{L,M}^{ILL}$ increases. For $N_L^{ISL} = 2$, the intra-layer capacity decreases by 30% when $N_{L,M}^{ILL}$ increases from 1 to 5, in which $\gamma_{L,M}$ decreases from 1 to 0.2 correspondingly. The inter-layer capacity also decreases slightly as $N_{L,M}^{ILL}$ increases. For $N_L^{ISL} = 2$, the intra-layer capacity decreases by 5% when $N_{L,M}^{ILL}$ increases from 1 to 5. For both intra-layer capacity and inter-layer capacity, we can find that allocating ILLs among all LEO satellites averagely is the best configuration mode for capacity performance.

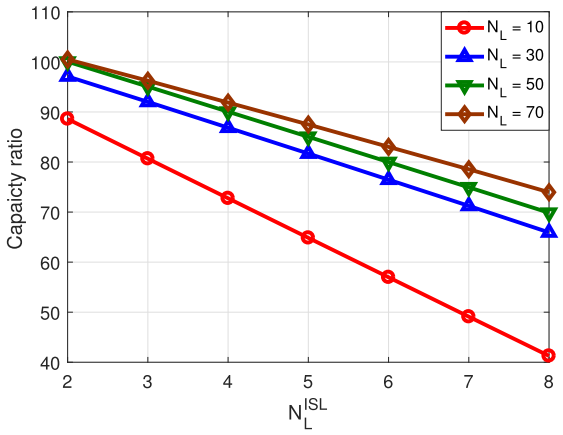
B. Long-Term Optimal Capacity Allocation

In this subsection, we then evaluate the proposed capacity allocation algorithm and provide a full analysis. Similarly, we consider there are total $N_T = 100$ topologies during time $T = 100$, and the time duration of each topology is 1. The total simulation time is $t_{max} = 100$. The number of users is set as $B = 5$, and the maximum visible time is set as $\tau_{max} = 3$. The capacity demand is classified into $L_1 = 3$ levels, and thus the total user type $L_2 = (L_1 + 1)\tau_{max} + 1 = 13$. The punishment factor is set as $\lambda = 1$, while the learning factor and discount factor is set as $\alpha = 0.5$ and $\gamma = 0.5$ separately. The maximum episode limit is set as $N_{episode\_max} = 1000$. We compare the three capacity allocation algorithms as follows:

1) **Optimal**. The optimal capacity allocation results are obtained by traversal search without simplification of the states.
2) **Long-term optimal**. This is the capacity allocation algorithm we proposed in Section V as Algorithm 1.
3) **Short-term optimal**. In the short-term optimal algorithm, the long-term utility is not considered. Instead, the capacity is allocated to users that can maximize the immediate utility of each time t.
4) **Random**. The capacity will be randomly allocated among users.

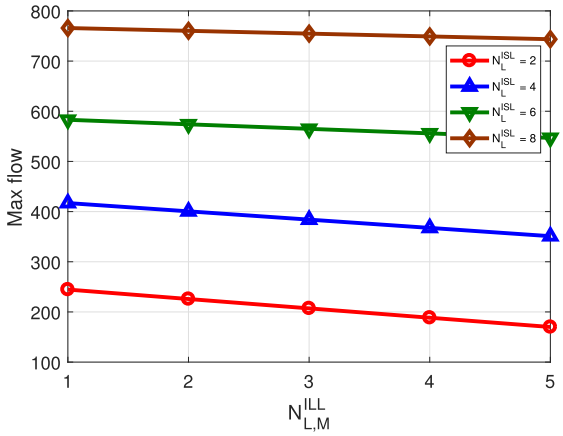


(a) Intra-layer capacity of LEO satellites

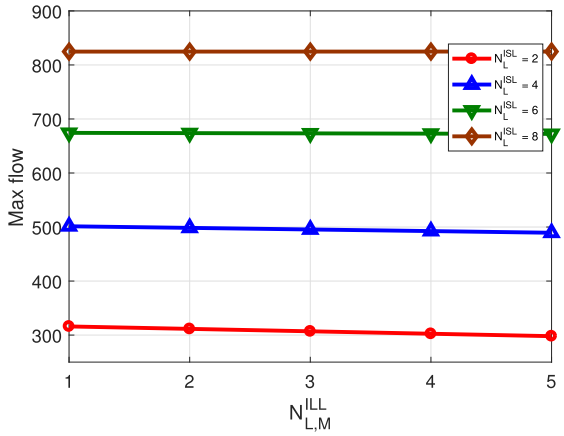


(b) Inter-layer capacity between LEO and MEO satellites

Fig. 8. Capacity ratio of LEO ISLs with different LEO number.



(a) Intra-layer capacity of LEO satellites



(b) Inter-layer capacity between LEO and MEO satellites

Fig. 9. Trade-off between $\gamma_{L,M}$ and $N_{L,M}^{ILL}$ with different LEO ISLs.

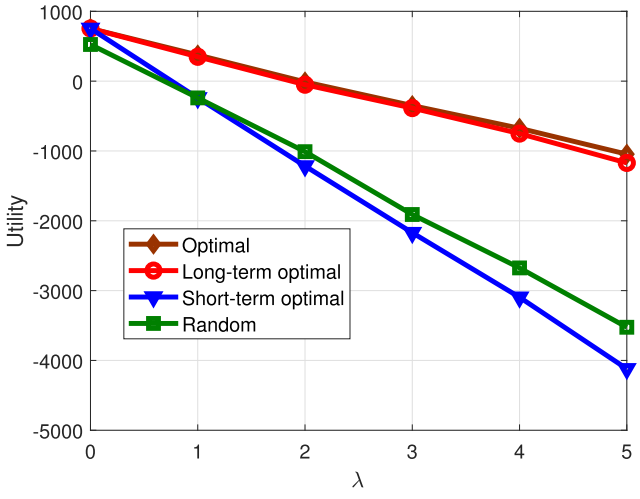


Fig. 10. Comparison of the three algorithms of different λ.

Fig. 10 shows the performance of the proposed long-term optimal algorithm while compared with the other two algorithms, and also the optimal results. The difference between the short-term utility and long-term utility mainly comes from the punishment of the unsatisfied capacity in (11). Thus we analyze the system utility of different punishment factor λ. We can observe that the long-term optimal algorithm can achieve significantly higher utility compared with the two other algorithms. While optimizing the total utility over one period of time, we cannot simply optimize the utility of each time t. If we always allocate large capacity to users with large value of τ_b at time t to maximize the utility at time t, we may obtain a large punishment at time $t+1$ because of the unsatisfied capacity demand. Thus we need to consider the long-term utility when allocating the capacity at each time t. Also, we can observe that compared with the optimal results obtained by traversal search, there is only slight performance loss for the proposed algorithm. Since the capacity allocation problem is a combined resource allocation and time scheduling problem, finding the optimal results by traversal search is of extremely high complexity, and only solvable for small-scale case. Based on the proposed long-term optimal algorithm, the quasi-optimal results can be obtained with only slight performance loss.

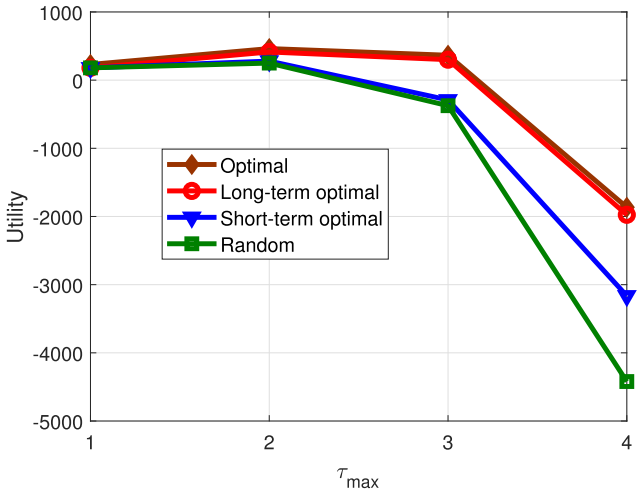


Fig. 11. Comparison of the three algorithms of different τ_{max}.

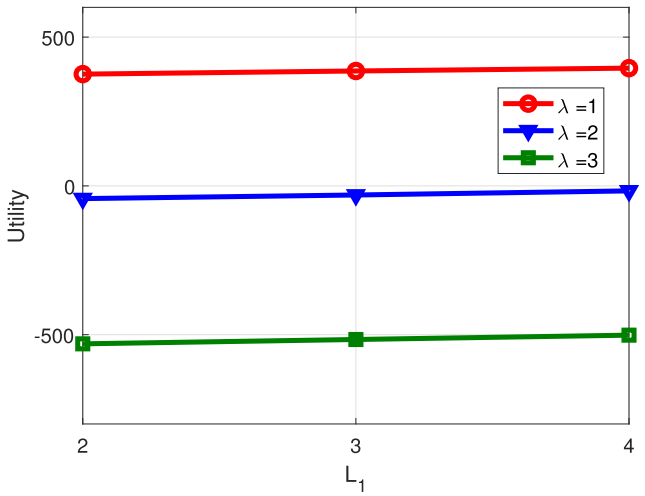


Fig. 12. Performance analysis of different L_1.

In the case of $\lambda = 0$, there is in fact no punishment for unsatisfied utility. Thus optimizing the total utility over time is equal to optimizing the immediate utility at each time t, and the short-term optimal algorithm is the optimal algorithm. In Fig. 10, we can observe that the long-term optimal algorithm can also achieve the optimal performance for $\lambda = 0$. Although the long-term optimal algorithm is designed to learn the long-term utility, it can also learn the short-term utility in the case where short-term utility is the optimal solution. Thus the proposed algorithm can always achieve the optimal utility for different system design. Also, larger value of the punishment factor λ means larger punishment for unsatisfied capacity, and the long-term utility will be more important for larger value of λ. If λ is larger than 2, the utility of the short-term optimal algorithm is even lower than the random algorithm, since the short-term optimal algorithm completely ignores the long-term utility. In systems where the loss is large for unsatisfied capacity, it is important to consider the long-term utility, and the short-term optimal algorithm will lead to low system utility. We can set large punishment factor λ for these systems to protect the system performance.

Fig. 11 shows the system utility of different maximum visible time τ_{max}. For $\tau_{max} = 1$, the visible times of all users are the same, and thus the utilities of all users are the same. The three algorithms can achieve the same performance in this case. However, if only τ_{max} is larger than 1, the optimal total system utility cannot be obtained if we only consider short-term utility. Generally, the visible times of different satellites are different. For satellite with longer visible time, the capacity allocation will be more complex due to the more complex user properties. Then optimizing the total utility will be difficult while taking the long-term utility into account. Also, if the capacity demands of users come with the same rate, the demand conflicts among users will be more frequent for longer visible time, which will lead to the decrease of system utility. When τ_{max} increase from 3 to 4, the system utility decreases significantly due to the unsatisfied capacity demands.

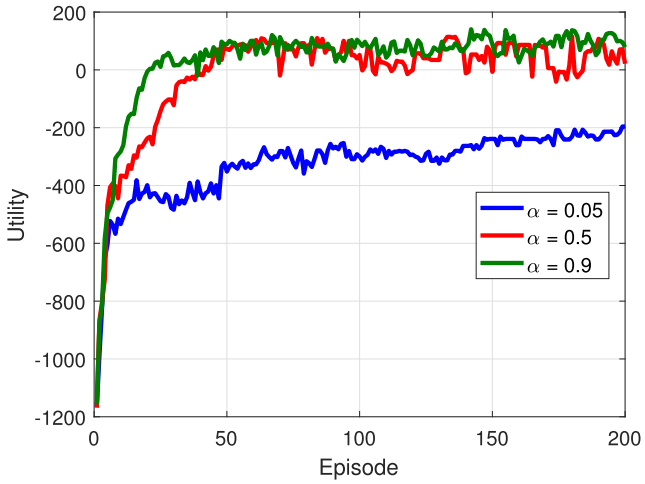


Fig. 13. Learning process of different α.

In order to reduce the number of system states, we classify the capacity demand of users into $L_1 + 1$ levels in the proposed algorithm. In Fig. 12, we give the system performance of different value of L_1 with different λ. By classifying the capacity demand of users into $L_1 + 1$ levels, we ignore some details of the user demands, which will lead to performance loss of the total utility. On one hand, larger value of L_1 can maintain more details of the user demands, and the capacity allocation can be more accurate. On the other hand, larger value of L_1 will also lead to larger number of system states, which will add to the system complexity. In Fig. 12, we can observe that the performance variation is small for different value of L_1. When L_1 decreases from 4 to 2, the utility loss is only 5%. However, for the case of $B = 5$, $C_{max} = 10$, $c_{max}^{req} = 10$, $\tau_{max} = 3$, the system states increases from 2.2×10^4 to 1.5×10^5 when L_1 increases from 2 to 4, increasing by almost 600%. Larger number of system states

will lead to larger storage, larger computing complexity, and longer training episodes. Thus there is a trade off between the complexity and the performance. In addition, although there is a little performance loss when L_1 is not large enough, the total system utility is still significantly larger than the short-term optimal algorithm, which proves the effectiveness of the proposed algorithm.

Finally, in Fig. 13, we show the learning process of the long-term optimal algorithm with different learning factor α. We can observe that the system learns faster with larger value of α. For $\alpha = 0.5$, the optimal utility is obtained after 50 episodes. However, for $\alpha = 0.9$, the same utility can be obtained within 30 episodes. Also, if only the learning factor is not too small, the system utility will converge to the same value for different value of α. On the other hand, we can observe that the system utility is still much lower than the optimal utility after 200 episodes for $\alpha = 0.05$. For higher learning efficiency, the learning factor should not be too small when applying the algorithm.

VI. Conclusion

In this paper, we investigated the problem of capacity management in the three-layer heterogeneous satellite network. Considering the node type and time, the satellite network was modeled as a multi-layer network of two aspects, and we proposed a low-complexity algorithm for calculating the capacity from any source satellite to any destination satellite. Then, based on Q-learning, we proposed a long-term optimal capacity allocation algorithm to optimize the long-term utility when allocating capacity among users. Finally, by means of numerical results, we analyzed the capacity performance of the three-layer heterogeneous satellite network and also evaluated the effectiveness of the proposed algorithms. In this paper, the proposed Q-learning algorithm works in a centralized manner, which may lead to large state space in the case of large-scale users. Then, the application of deep learning methods, such as deep Q-learning, can be further explored for better performance. Also, variable punishment factors for different types of users and services can be considered in the case of more complex user model and service model, which can be further investigated in future and following works.

References

[1] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[2] V. W. Wong, R. Schober, D. W. K. Ng, and L.-C. Wang, *Key Technologies for 5G Wireless Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[3] L. Kuang, X. Chen, C. Jiang, H. Zhang, and S. Wu, "Radio resource management in future terrestrial-satellite communication networks," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 81–87, Oct. 2017.

[4] X. Zhu, C. Jiang, L. Kuang, N. Ge, S. Guo, and J. Lu, "Cooperative transmission in integrated terrestrial-satellite networks," *IEEE Netw.*, vol. 33, no. 3, pp. 204–210, May 2019.

[5] X. Zhu, C. Jiang, L. Kuang, N. Ge, and J. Lu, "Non-orthogonal multiple access based integrated terrestrial-satellite networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 10, pp. 2253–2267, Oct. 2017.

[6] X. Zhu, C. Jiang, L. Yin, L. Kuang, N. Ge, and J. Lu, "Cooperative multigroup multicast transmission in integrated terrestrial-satellite networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 5, pp. 981–992, May 2018.

[7] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2714–2741, 4th Quart., 2018.

[8] B. Di, L. Song, Y. Li, and H. V. Poor, "Ultra-dense LEO: Integration of satellite access networks into 5G and beyond," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 62–69, Apr. 2019.

[9] N. Kato *et al.*, "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 140–147, Aug. 2019.

[10] F. Battiston, V. Nicosia, and V. Latora, "The new challenges of multiplex networks: Measures and models," *Eur. Phys. J. Special Topics*, vol. 226, no. 3, pp. 401–416, Feb. 2017.

[11] K.-K. Kleineberg, M. Boguñá, M. Á. Serrano, and F. Papadopoulos, "Hidden geometric correlations in real multiplex networks," *Nature Phys.*, vol. 12, no. 11, pp. 1076–1081, Nov. 2016.

[12] M. Kivela, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *J. Complex Netw.*, vol. 2, no. 3, pp. 203–271, Jul. 2014.

[13] R. Gallotti and M. Barthelemy, "The multilayer temporal network of public transport in great Britain," *Sci. Data*, vol. 2, no. 1, pp. 1–8, Dec. 2015.

[14] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, "Time-varying graphs and dynamic networks," *Int. J. Parallel, Emergent Distrib. Syst.*, vol. 27, no. 5, pp. 387–408, Apr. 2012.

[15] R. Dhaou, L. Franck, A. Halchin, E. Dubois, and P. Gelard, "Gateway selection optimization in Hybrid MANET-satellite network," in *Proc. Int. Conf. Wireless Satell. Syst.* Cham, Switzerland: Springer, 2015, pp. 331–344.

[16] R. Fdhila, T. M. Hamdani, and A. M. Alimi, "A multi objective particles swarm optimization algorithm for solving the routing pico-satellites problem," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Oct. 2012, pp. 1402–1407.

[17] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[18] T. Zhang, H. Li, J. Li, S. Zhang, and H. Shen, "A dynamic combined flow algorithm for the two-commodity max-flow problem over delay-tolerant networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 7879–7893, Dec. 2018.

[19] P. Wang, X. Zhang, S. Zhang, H. Li, and T. Zhang, "Time-expanded graph-based resource allocation over the satellite networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 360–363, Apr. 2019.

[20] G. F. Italiano, Y. Nussbaum, P. Sankowski, and C. Wulff-Nilsen, "Improved algorithms for min cut and max flow in undirected planar graphs," in *Proc. 43rd Annu. ACM Symp. Theory Comput. (STOC)*, 2011, pp. 313–322.

[21] H. Li, T. Zhang, Y. Zhang, K. Wang, and J. Li, "A maximum flow algorithm based on storage time aggregated graph for delay-tolerant networks," *Ad Hoc Netw.*, vol. 59, pp. 63–70, May 2017.

[22] R. Liu, M. Sheng, K.-S. Lui, X. Wang, D. Zhou, and Y. Wang, "Capacity of two-layered satellite networks," *Wireless Netw.*, vol. 23, no. 8, pp. 2651–2669, Nov. 2017.

[23] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.

[24] N. D. Vanli, M. O. Sayin, I. Delibalta, and S. S. Kozat, "Sequential nonlinear learning for distributed multiagent systems via extreme learning machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 546–558, Mar. 2017.

[25] J. Tang, G. Leu, and H. A. Abbass, "Networking the boids is more robust against adversarial learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 5, no. 2, pp. 141–155, Apr./Jun. 2018.

[26] C. Jiang, Y. Chen, Q. Wang, and K. J. R. Liu, "Data-driven auction mechanism design in IaaS cloud computing," *IEEE Trans. Services Comput.*, vol. 11, no. 5, pp. 743–756, Sep./Oct. 2018.

[27] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Cambridge Univ., Cambridge, U.K., 1989.

[28] C. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, May 1992.

[29] C. Qiu, H. Yao, F. R. Yu, F. Xu, and C. Zhao, "Deep Q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5871–5883, Jun. 2019.

[30] Y. Kawamoto, H. Takagi, H. Nishiyama, and N. Kato, "Efficient resource allocation utilizing Q-learning in multiple UA communications," *IEEE Trans. Netw. Sci. Eng.*, vol. 6, no. 3, pp. 293–302, Jul./Sep. 2019.

[31] A. Nassar and Y. Yilmaz, "Reinforcement learning for adaptive resource allocation in fog RAN for IoT with heterogeneous latency requirements," *IEEE Access*, vol. 7, pp. 128014–128025, 2019.

[32] R. Amiri, M. A. Almasi, J. G. Andrews, and H. Mehrpouyan, "Reinforcement learning for self organization and power control of two-tier heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 3933–3947, Aug. 2019.

[33] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, Feb. 2020.

[34] F. Fidler, M. Knapek, J. Horwath, and W. R. Leeb, "Optical communications for high-altitude platforms," *IEEE J. Sel. Topics Quantum Electron.*, vol. 16, no. 5, pp. 1058–1070, Sep./Oct. 2010.

[35] B. Deng, C. Jiang, L. Kuang, S. Guo, J. Lu, and S. Zhao, "Two-phase task scheduling in data relay satellite systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1782–1793, Feb. 2018.

[36] T. Li, H. Zhou, H. Luo, and S. Yu, "SERvICE: A software defined framework for integrated space-terrestrial satellite communication," *IEEE Trans. Mobile Comput.*, vol. 17, no. 3, pp. 703–716, Mar. 2018.

[37] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *J. ACM*, vol. 19, no. 2, pp. 248–264, Apr. 1972.

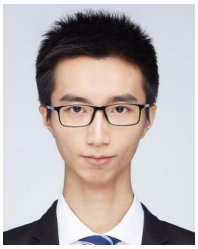
[38] R. Gopal and N. BenAmmar, "Framework for unifying 5G and next generation satellite communications," *IEEE Netw.*, vol. 32, no. 5, pp. 16–24, Sep./Oct. 2018.

[39] J. Zhang, E. Björnson, M. Matthaiou, D. W. K. Ng, H. Yang, and D. J. Love, "Prospective multiple antenna technologies for beyond 5G," 2019, *arXiv:1910.00092*. [Online]. Available: http://arxiv.org/abs/1910.00092

[40] Q. Wei, F. L. Lewis, Q. Sun, P. Yan, and R. Song, "Discrete-time deterministic Q-learning: A novel convergence analysis," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1224–1237, May 2017.

[41] Y. Su, Y. Liu, Y. Zhou, J. Yuan, H. Cao, and J. Shi, "Broadband LEO satellite communications: Architectures and key technologies," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 55–61, Apr. 2019.

Chunxiao Jiang (Senior Member, IEEE) received the B.S. degree (Hons.) in information engineering from Beihang University, Beijing, in 2008, and the Ph.D. degree (Hons.) in electronic engineering from Tsinghua University, Beijing, in 2013. He is currently an Associate Professor with the School of Information Science and Technology, Tsinghua University. His research interests include application of game theory, optimization, and statistical theories to communication, networking, and resource allocation problems, in particular space networks and heterogeneous networks. He was a recipient of the Best Paper Award from the IEEE GLOBECOM in 2013, the Best Student Paper Award from the IEEE GlobalSIP in 2015, the IEEE Communications Society Young Author Best Paper Award in 2017, the Best Paper Award IWCMC in 2017, the IEEE ComSoc TC Best Journal Paper Award of the IEEE ComSoc TC on Green Communications and Computing 2018, the IEEE ComSoc TC Best Journal Paper Award of the IEEE ComSoc TC on Communications Systems Integration and Modeling 2018, and the Best Paper Award from ICC 2019. He received the Chinese National Second Prize in Technical Inventions Award in 2018 and the Natural Science Foundation of China Excellent Young Scientists Fund Award in 2019. He has served as a member of the Technical Program Committee and the Symposium Chair for a number of international conferences, including the IEEE CNS 2020 Publication Chair, the IEEE WCSP 2019 Symposium Chair, the IEEE ICC 2018 Symposium Co-Chair, the IWCMC 2020/19/18 Symposium Chair, the WiMob 2018 Publicity Chair, the ICCC 2018 Workshop Co-Chair, and the ICC 2017 Workshop Co-Chair. He has also served as an Editor for the IEEE INTERNET OF THINGS Journal, the *IEEE Network*, and the IEEE COMMUNICATIONS LETTERS, and a Guest Editor for the *IEEE Communications Magazine*, the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, and the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING.



Xiangming Zhu received the B.S. and Ph.D. degrees in electronic engineering from Tsinghua University, China, in 2014 and 2019, respectively. He is currently a Post-Doctoral Researcher with the Zhejiang Lab, Beijing National Research Center for Information Science and Technology, Tsinghua University. His major research interests include satellite networking, integrated terrestrial-satellite communications, and resource allocation problems. He received the Best Paper Award IWCMC in 2017.