mathematics

MDPI

Article

An Exact Algorithm for Multi-Task Large-Scale Inter-Satellite Routing Problem with Time Windows and Capacity Constraints

Jinming Liu [1], Guoting Zhang [2], Lining Xing [3,*], Weihua Qi [1] and Yingwu Chen [1]

[1] College of Systems Engineering, National University of Defense Technology, No. 109 Deya Street, Changsha 410073, China
[2] Beijing Institute of Tracking and Telecommunications Technology, Beijing 100094, China
[3] College of Electronic Engineering, Xidian University, Xi'an 710071, China
* Correspondence: lnxing@xidian.edu.cn

Abstract: In the context of a low-orbit mega constellation network, we consider the large-scale inter-satellite routing problem with time windows and capacity constraints (ISRPTWC) with the goal of minimizing the total consumption cost, including transmission, resource consumption, and other environmentally impacted costs. Initially, we develop an integer linear programming model for ISRPTWC. However, a difficult issue when solving ISRPTWC is how to deal with complex time window constraints and how to reduce congestion and meet transmission capacity. Along this line, we construct a three-dimensional time-space state network aiming to comprehensively enumerate the satellite network state at any moment in time and a task transmission route at any given time and further propose a time-discretized multi-commodity network flow model for the ISRPTWC. Then, we adopt a dynamic programming algorithm to solve the single-task ISRPTWC. By utilizing a Lagrangian relaxation algorithm, the primal multi-task routing problem is decomposed into a sequence of single-task routing subproblems, with Lagrangian multipliers for individual task route nodes and links being updated by a subgradient method. Notably, we devise a novel idea for constructing the upper bound of the ISRPTWC. Finally, a case study using illustrative and real-world mega constellation networks is performed to demonstrate the effectiveness of the proposed algorithm.

Keywords: low-orbit mega constellation network; inter-satellite routing problem with time windows and capacity constraints; time-discretized multi-commodity network flow model; Lagrangian relaxation algorithm

MSC: 90-08; 90B06; 90B40

check for updates

Citation: Liu, J.; Zhang, G.; Xing, L.; Qi, W.; Chen, Y. An Exact Algorithm for Multi-Task Large-Scale Inter-Satellite Routing Problem with Time Windows and Capacity Constraints. *Mathematics* **2022**, *10*, 3969. https://doi.org/10.3390/math10213969

Academic Editor: Catalin Stoean

Received: 6 September 2022
Accepted: 14 October 2022
Published: 26 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Though modern information technology and network service have reached unprecedented levels, some regions still lack easy access to the internet [1,2]. With wide coverage, all-weather, low latency, and low-cost characteristics, a low-orbit satellite network has become a necessary tool to solve the problem of insufficient ground network coverage. Meanwhile, building a low-orbit mega constellation network has become a current development trend. The emerging low-orbit mega constellation network intends to complement the ground network by providing wide coverage, low latency, and high-volume internet access services. However, there are many challenges in the management of the mega constellation network, among which is inter-satellite routing technology, as technical support for navigation enhancement, broadband internet access, and other functions, is one of the most urgent challenges to be solved.

Inter-satellite routing refers to the uploading of tasks to the satellite network through a designated satellite within a definite time and then using inter-satellite links to download to a designated device within a definite time. Meanwhile, inter-satellite routing planning

refers to using routing algorithms to plan transmission schemes for multiple tasks simultaneously, with the aim of minimizing the use of satellite resources or scheduling as many tasks as possible while meeting user requirements. As shown in Figure 1, a user terminal uploads a task to the entry satellite. Then, the inter-satellite link transmission scheme of the task is obtained using a routing algorithm. Finally, the task is downloaded to the designated device by a gateway satellite according to the transmission scheme. However, due to the characteristics of a low-orbit mega constellation network with numerous satellites, user services, and complex task requirements, inter-satellite routing planning has become a difficult problem to be solved.

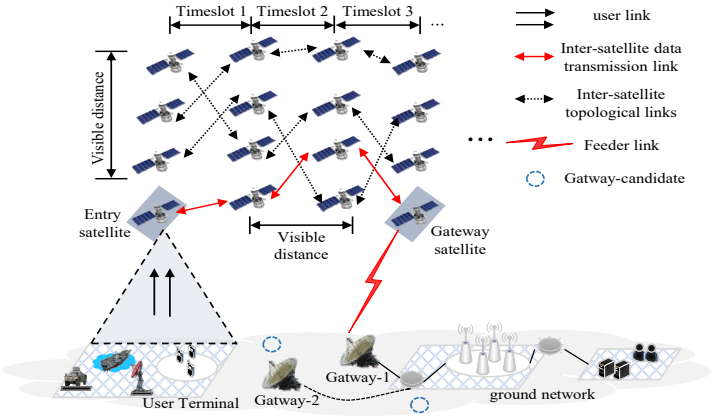


Figure 1. Task transmission process of low-orbit constellation network.

The inter-satellite routing problem has been solved using heuristics [3,4], but the inter-satellite routing problem with mega constellations, large amounts of tasks, and complex requirements have not been well solved. In this study, to meet the routing requirements in highly dynamic network topologies, especially for real-world mega constellation networks such as OneWeb and Starlink, we propose a new mathematical programming model for large-scale inter-satellite routing problems with time windows and capacity constraints (ISRPTWC), which can fully consider time windows and network capacity constraints and avoid network congestion. Meanwhile, based on the Lagrangian relaxation solution framework, we further design an efficient solution algorithm for solving the ISRPTWC and verify the feasibility of the proposed algorithm with a constellation size of 900 satellites and 1200 user requirements.

1.1. Literature Review on the Inter-Satellite Routing Algorithm

Although the inter-satellite routing problem has been a research topic and challenge in low-orbit satellite networks, which has led to many related studies [5,6], it still has received less attention compared to other types of satellite scheduling such as satellite imaging and measurement scheduling [7–9].

Based on the routing characteristics, the existing satellite network routing algorithms can be divided into centralized and distributed routing algorithms, static routing, and dynamic routing algorithms. According to the constellation topology dynamic processing method, it can be further divided into routing algorithms based on the virtual topology method and the virtual node method. Some representative routing schemes are discussed below. The main idea of the centralized routing algorithm is to compute the optimal route solution for all tasks uniformly while avoiding constraint violations. Du et al. [10] established software-defined networking (SDN) to handle multi-route TCP for Low Earth Orbit (LEO) satellite constellations. The SDN framework with a centralized method enhances system scalability but also increases system complexity making it difficult to solve large-scale routing problems. The advantage of distributed routing algorithms is that each satellite node in the constellation network calculates its route individually [11], reducing the transmission cost of a large number of messages in the network [12] and eliminating the

requirement for a centralized control center [13]. Utilizing a distributed framework, Eylem et al. [14] proposed a datagram routing algorithm to solve the LEO satellite network problem. Moreover, to improve the system flexibility and facilitate the management of ground facilities, Papapetrou et al. [15] developed a location-assisted on-demand routing protocol (LAOR), and the LAOR protocol showed better results in comparison with centralized routing protocols. However, some situations where the topology of the satellite network is constantly changing [16] also greatly increase the difficulty of studying inter-satellite routing algorithms. Therefore, many researchers classified routing characteristics into static and dynamic routing. Chang et al. [17] considered the LEO satellite network as a finite-state automaton (FSA), where the FSA framework converted a link assignment problem in a satellite network into a set of link assignment problems in a fixed topology network. Their simulation results showed that a static routing algorithm performed better than a dynamic routing algorithm for the same link assignment scenario. However, Soret et al. [18] argued that dynamic routing algorithms must be used in dynamic satellite networks. Compared to static routing algorithms, dynamic routing algorithms were better suited to deal with unexpected situations such as node and link changes and network congestion. Therefore, Soret et al. proposed a simple distributed GomHop autonomous routing algorithm for a low-orbit mega constellation. Later, Li et al. [19] proposed a novel Temporal Netgrid Model (TNM) for the problem of traditional routing algorithms that cannot handle satellite trajectories, and they developed an efficient Netgrid-based Shortestroute Routing (NSR) algorithm based on the TNM. Nevertheless, inter-satellite routing problems have become more and more complex as user requirements have increased. The shortcomings of dynamic routing algorithms in dealing with inter-satellite routing problems have gradually emerged. Therefore, some researchers have adopted a "divide and conquer" approach to the inter-satellite routing problem. The specific idea is to use the proposed virtual topology method [20] to deal with the satellite topology change problems. Then, some congestion avoidance and load-balancing routing algorithms [21,22] were designed to solve the routing problems. Additionally, in recent years, some routing algorithms based on machine learning and deep reinforcement learning [23] have gradually gained attention in satellite network routing problems. Given that a mega constellation is highly complex and dynamic, artificial intelligence approaches may also present new opportunities for the design of routing algorithms for a mega constellation.

Although the above studies have proposed many algorithms for the inter-satellite routing problem, most of them do not apply to the large-scale inter-satellite routing problem. Meanwhile, many studies have oversimplified the inter-satellite routing problem that cannot be used in practical applications, and these algorithms are still lacking in theoretical interpretation. Consequently, how to balance the complexity of the actual problem and the effectiveness of the routing algorithm is a very critical and urgent problem to be solved.

1.2. Literature Review of Exact Solution Methods on Routing Problems

To the extent of our knowledge, few studies have focused on the applicability of exact solution methods to the inter-satellite routing problem with time windows and capacity constraints (ISRPTWC). However, the vehicle routing problem with time windows (VRPTW) [24] problem and the vehicle routing problem with pickup and delivery (VRPPD) [25,26] problem have been studied, and a large number of exact solution algorithms have been proposed. In this work, the ISRPTWC refers to optimizing the transmission routes and schedules of tasks without violating all satellites and links capacity constraints, in which the schedule refers to specifying the upload start time and downlink termination time for each task. Without considering multi-tasks scenarios, ISRPTWC and vehicle routing problems with pickup and delivery with time windows (VRPPDTW), which VRPPDTW consists of optimizing a set of vehicle routes and schedules with known passenger demand without violating vehicle capacity constraints, are essentially sequential combinatorial optimization problems with complex constraints.

Based on VRPPDTW, some applications have been extended in maritime, air, and road transportation, in which the studies related to maritime routing and scheduling [27,28], air cargo routing and scheduling [29], and road freight routing and scheduling [30,31] are well known. School bus routing and scheduling [32] and ridesharing [33,34] are also of great interest in recent years' research on VRPPDTW. These research methods are useful for solving ISRPTWC. Considering the single-vehicle pickup and drop-off problem with time windows (PDPTW), Psaraftis [35] developed an exact forward dynamic programming solution to minimize the waiting time and service time for all passengers. Sexton et al. [36] applied the Benders' decomposition method to handle the main routing problem and the scheduling subproblem for the single-vehicle PDPTW. Later, many researchers changed their focus from single-vehicle PDPTW to multi-vehicles PDPTW. Savelsbergh et al. [37] proposed an algorithm combining heuristics in which the primary objective was to minimize the number of demanded vehicles, and the secondary objective function was to minimize the route cost. Lu et al. [38] proposed a double exponential formulation and used a branch-and-cut algorithm to solve the multi-vehicles PDPTW. Ropke [39] developed a novel branch-and-cut-and-price algorithm in which the lower bound was computed using a column generation algorithm, and various inequalities were designed to improve the computational efficiency. Kok et al. [40] and Gromicho et al. [41] considered the route congesting case in the multi-vehicle PDPTW and solved it using a directed graph approach. Monirehalsadat et al. [42] designed an exact solution framework to solve VRPPDTW considering traffic congestion cases, in which they first developed a discrete-time multi-commodity flow model and decomposed the problem into the main problem and single-vehicle PDPTW subproblems, then solved the subproblem using dynamic programming methods in a Lagrangian relaxation framework, and finally validated it in Chicago sketch and Phoenix regional network.

Combining the above methods and considering the complex space environment of the mega constellation, we use the space-time network scheme to construct the time-space state network for ISRPTWC. The constructed time-space state network can easily represent a task upload window and downlink. Also, the introduced time-space state network ensures that we can use the dynamic programming algorithm to solve the single-task ISRPTWC. For the multi-task ISRPTWC, we use the Lagrangian relaxation framework to decompose the original problem into a series of sub-problems with time window constraints by relaxing the satellite and linking capacity-related constraints into the objective function and design subgradient methods to update the Lagrangian multipliers.

1.3. Motivation and Potential Contributions

Due to the large size of satellites and user requirements, and complex constraints of ISRPTWC in the low-orbit mega constellation, few researchers have proposed a nicely exact solution algorithm. Meanwhile, exact solution methods are considered to handle only small-scale problems in general combinatorial optimization problems, so many researchers use meta-heuristic methods, routing strategies, and routing protocols to solve inter-satellite routing problems. However, these methods have certain drawbacks, such as oversimplification of the problem and poor theoretical interpretation of algorithms. Therefore, we hope to propose an exact solution algorithm to solve the ISRPTWC for the low-orbit mega constellation in a reasonable time.

We aim to merge pioneering work in several different directions. Specifically, the main contributions are summarized as follows. (1) We present the ISRPTWC for a low-orbit mega constellation, develop an integer linear programming model, and further construct a time-discrete multi-commodity flow model. (2) Time-space state networks are constructed for the ISRPTWC, and the single-task routing problem is efficiently solved using a dynamic programming method in the context of time-space state networks. (3) The extended Lagrangian relaxation framework is proposed to solve multi-task ISRPTWC, and a novel feasible solution construction method is constructed to obtain a tight upper bound.

Finally, we further design cases to verify the computational efficiency and solution quality of the proposed algorithm.

The remainder of this paper is structured as follows. Section 2 presents the ISRPTWC description and mathematical formulation. In Section 3, the Lagrangian relaxation solution framework is present, and the lower and upper bound of the ISRPTWC solution methods is proposed. Some solution space reduction strategies for handling large-scale multi-task ISRPTWC are presented in Section 4. Then, Section 5 provides experimental results of the illustrative and actual cases. Finally, we provide our conclusions in Section 6.

2. Problem Statement and Model Formulation

In this section, we first define the ISRPTWC and build an integer linear programming model. Then, based on the ISRPTWC, we construct the time-space state network. In the time-space state network, a time-discrete multi-commodity flow model is developed for the ISRPTWC.

2.1. Problem Statement

The ISRPTWC is how to compute the optimal transmission route or set of optimal transmission routes for tasks with multiple upload windows and multiple downlink windows, given known network capacity constraints. As shown in Figure 1, first, the user defines a task at a specific moment, including the data size, the upload time window, and the downlink time window. After that, the optimal transmission routes are calculated using the known inter-satellite topology, node, and link capacity information. Finally, the tasks are downlinked to the designated locations through the constellation network. In this study, we aim to find a set of transmission routes for multiple tasks that minimizes the total transmission cost. The overall transmission cost includes the transmission route cost, the environmental loss cost, and the inherent cost of the resources.

As shown in Figure 2, consider an inter-satellite routing network $G = (N, E)$, in which N is a set of nodes and E is a set of connected edges. We define t_{ij}^{d}, c_{ij}^{d} to denote the transmission time and transmission cost of a task d in using the edge (i, j). For simplicity, we use the number of inter-satellite task forwarding hops instead of the inter-satellite task transmission time to calculate a task transmission cost [3]. Meanwhile, Cap_i and Cap_{ij} are used to indicate the maximum transmission capacity of nodes and links. To reduce the packet loss rate, we define the maximum congestion time Blo_d, which is the duration of a task on a satellite. Sets, indices, and parameters of this study are summarized in Table 1. It is worth noting that all nodes in the network are divided into three categories, including the set of task upload nodes ST and the set of task downlink nodes AR, the task intermediate transmission nodes $N_1 \cup N_2$, and the virtual upload node 0 and virtual downlink node $2n + 1$. The starting and ending node of all tasks are defined to the virtual upload node and virtual downlink node, respectively.

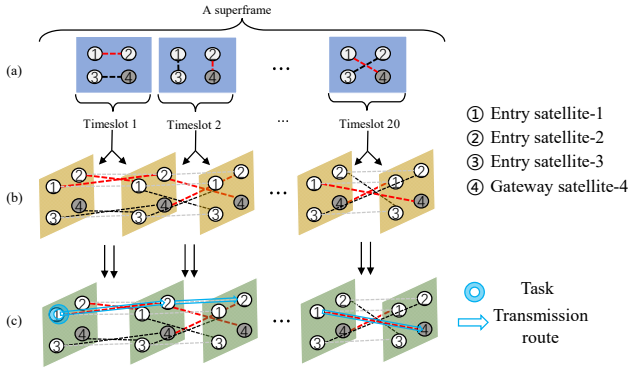


Figure 2. Inter-satellite routing network structure. (**a**) Inter-satellite link schemes within each time slice; (**b**) Inter-satellite link schemes in space-time networks; (**c**) Routing schemes based on inter-satellite topologies.

Table 1. Sets, indices, and parameters.

Symbol	Definition
S	Satellite set, $S = \left\{s_1, \cdots, s_{\|S\|}\right\}$.
$GATA$	Gateway station set, $GATA = \left\{g_1, \cdots, g_{\|GATA\|}\right\}$.
T	Timeslot set, $TS = \left\{ts_1, \cdots, ts_{\|TS\|}\right\}$.
N	A set of nodes consisting of different timeslot satellites and letter gateway stations for the current superframe and the next superframe, $N = N_1 \cup N_2$.
N_1	The set of nodes of the current superframe, $N_1 = \left(\left\{ts_1, \cdots, ts_{\|TS\|}\right\}, \left\{s_1, \cdots, s_{\|S\|}, g_{\|S\|+1}, \cdots, g_{\|S\|+\|GATA\|}\right\}\right)$.
E	The set of edges consisting of connection routes between satellites with different timeslots.
G	Direct Graph, $G = (N, E)$.
D	Task Set, $\|D\| = n$.
ST	Task upload nodes Set, $ST = \{st_1, \cdots, st_n\}$, $st\epsilon\left(\left\{ts_1, \cdots, ts_{\|TS\|}\right\}, \left\{s_1, \cdots, s_{\|S\|}\right\}\right)$.
AR	Task downlink nodes Set, $AR = \{ar_1, \cdots, ar_n\}$, $ar\epsilon\left(\left\{ts_1, \cdots, ts_{\|TS\|}\right\}, \left\{g_1, \cdots, g_{\|GATA\|}\right\}\right)$.
SA	Set of task upload and download nodes, $\{ST, AR\}$.
0	Task virtual start node.
$2n+1$	Task virtual termination node.
d	Task Index.
Dur_d	Maximum transmission time for task d,$d \in D$.
Pri_d	Transmission priority of task d, $d \in D$.
Blo_d	Maximum blocking time for task d, $d \in D$.
Cap_i	Capacity of node i.
(i, j)	Index of link between adjacent nodes i and j.
c_{ij}^d	Transmission cost of link i to j.
t_{ij}^d	Transmission time of link i to j.
Cap_{ij}	Transmission capacity of link i to j.

Before modeling, some reasonable assumptions need to be made for the ISRPTWC:

(1) The transmission process between the gateway satellite and the gateway station and between the gateway station and the ground network is not considered;
(2) Access satellites are defined as satellites that receive tasks, and gateway satellites are defined as satellites that connect to ground networks;
(3) A user can communicate with only two satellites at any time, including an ascending and a descending satellite;
(4) Ignoring transmission failures caused by other factors, packet loss occurs only when a task transmission exceeds the maximum congestion time.

2.2. Integer Linear Programming Model

Considering that the transmission cost of each task is different, we define the decision variable of the integer linear programming model (denoted as M1) as x_{ij}^d which takes the value one when task d is transmitted using link (i, j) and zero otherwise. It is worth noting that each task is defined with a priority attribute, and a task with higher priority is transmitted first. However, the objective function of the M1 model is to minimize the total transmission cost. Therefore, we consider the reciprocal of each task priority as the weight of each task in the objective function. The integer linear formulation of the multi-task ISRPTWC can be expressed as follows:

$$Min \sum_{d \in D} \left(\frac{1}{Pri_d}\right) \sum_{i \in N} \sum_{j \in N} c_{ij}^d x_{ij}^d \tag{1}$$

$$\sum_{st_i \in ST} \sum_{j \in N} x^d_{st_{i,j}} = 1, \ \forall d \in D \tag{2}$$

$$\sum_{j \in N} x^d_{st_{i,j}} - \sum_{j \in N} x^d_{ar_{i,j}} = 0, \forall st_i \in ST, \forall ar_i \in AR \tag{3}$$

$$\sum_{j \in ST} x^d_{0j} = 1, \ \forall d \in D \tag{4}$$

$$\sum_{j \in N} x^d_{ji} - \sum_{j \in N} x^d_{ij} = 0, \ \forall d \in D, \forall i \in N \tag{5}$$

$$\sum_{i \in AR} x^d_{i,2n+1} = 1, \ \forall d \in D \tag{6}$$

$$\sum_{i \in N} \sum_{j \in N} t^d_{ij} x^d_{ij} \leq Dur_d, \ \forall d \in D \tag{7}$$

$$\sum_{d \in D} \sum_{i \in N \cup \{0\}} x^d_{ij} \leq Cap_j, \forall i \in N \tag{8}$$

$$\sum_{d \in D} x^d_{ij} \leq Cap_{ij}, \forall i, j \in N, i.satellite \neq j.satellite \tag{9}$$

$$\sum_{k=0}^{Blo_d} x^d_{i+k*|S|, i+(k+1)*|S|} \leq Blo_d, \ \forall i \in N_1, \forall d \in D \tag{10}$$

$$x_{ij} \leq Cap_{ij}, \forall i, j \in N \tag{11}$$

$$x^d_{ij} \in \{0, 1\}, \ \forall i \in N \cup \{0\}, \forall j \in N \cup \{2n+1\}, \forall d \in D \tag{12}$$

The objective function (1) is to minimize the total transmission cost. Constraints (2) and (3) represent that each task has a unique upload node and a downlink node. Constraints (4), (5), and (6) are flow balancing constraints, where (4) means all tasks start from the virtual origin node, (5) means all tasks cannot be reserved at intermediate nodes, and (6) means all tasks are transferred to the virtual termination node. Constraint (7) limits the maximum transmission time of tasks. Constraints (8) and (9) represent the capacity limits of nodes and links in the network. Constraint (10) denotes the duration of a task that is less than {Blo}_d on the same satellite. Constraint (11) indicates that links with insufficient capacity cannot be defined as transmission routes. The decision variables are defined in Constraint (12).

2.3. Time-Space State Network Modeling Perspective

Although the integer linear programming model for multi-task ISRPTWC is well established and can be mathematically explicit, the multi-task ISRPTWC with similar properties to the multi-vehicle VRPPDTW [42] is also an NP-hard problem in nature and is difficult to solve efficiently using existing solvers. Therefore, based on the time-extended network [43], we propose a novel mathematical idea, which is to formulate a time-discrete multi-commodity flow model based on the time-space state network for the multi-task ISRPTWC.

In this section, we construct the time-space state network for multi-task ISRPTWC. As shown in Figure 3, we first use the virtual topology method to obtain the set of time slices. After that, the time-space state network is constructed based on the topological distribution relationship between satellites in different time slices. In the network, there are four states of a task transmission process, namely upload, wait, transmit, and downlink, and the transition relationship between the states is shown in Table 2. For example, the transfer state matrices for task 1 and task 2 are $[S_1, S_2, S_3, S_3, S_2, S_2, S_4]$, $[S_1, S_2, S_2, S_3, S_2, S_2, S_4]$, respectively. After constructing the time-space state network, we redefine a node index in the network as (i, s, w), in which i represents the satellite index, s represents the current time index, and w represents the current state of a task. The link can be represented as (i, j, s, e, w, w'),$t(d, i, j, s, e, w, w')$, and $c(d, i, j, s, e, w, w')$,

and can be expressed as the transmission time (the transmission time has been simplified above as the number of transmission hops) and the transmission cost of the task d on link (i, j, s, e, w, w'), respectively. In addition, the capacity of nodes and links has been embedded in the time-space state network. The specific parameters are defined in Table 3. Therefore, we can obtain the transmission route of task 1 in Figure 3 as $(0, 06, Waiting\ for\ unload) \rightarrow (7, 07, S_1) \rightarrow (7, 08, S_2) \rightarrow (6, 09, S_3) \rightarrow (9, 10, S_3) \rightarrow (9, 11, S_2) \rightarrow (GS, 12, S_4)$, and the transmission cost is also obtained.

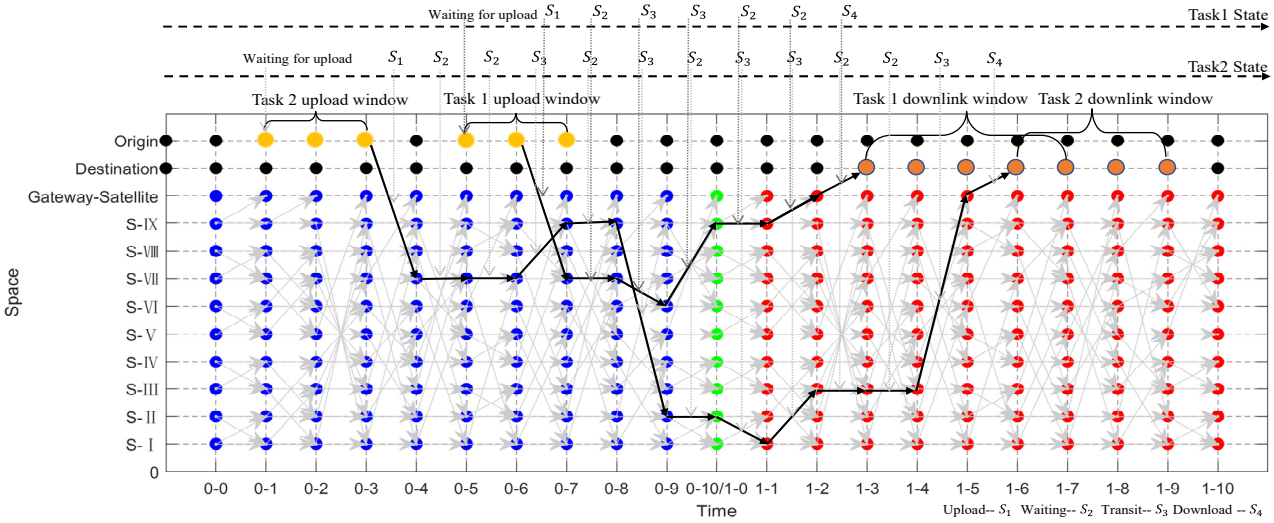


Figure 3. Time-space state network.

Table 2. Task state transition representation.

$\frac{w}{w'}$	Upload/S_1	Waiting/S_2	Transit/S_3	Download/S_4
Upload/S_1	No change	True	True	True
Waiting/S_2	False	No change	True	True
Transit/S_3	False	True	True	True
Download/S_4	False	False	False	—

Table 3. Sets, indices, and parameters.

Symbol	Definition
$(i, s, w),\ (j, e, w')$	Time-space state index of nodes.
(i, j, s, e, w, w')	Link index, meaning that the current link can transmit a task with state w on node i at time s to node j at time e and change the task state to w'.
$\{Pri_d, Dur_d, Blo_d\} \subseteq d$	Task d contains attributes such as priority Pri_d, maximum transmission time Dur_d, maximum congestion time Blo_d.
$c(d, i, j, s, e, w, w')$	The transmission cost of task d.
$t(d, i, j, s, e, w, w')$	The transmission time of task d.
$y(d, i, j, s, e, w, w')$	$y(d, i, j, s, e, w, w') = 1$ if link (i, j, s, e, w, w') is used to transmit task d; and zero otherwise.
$(i, j, s, e, w, w') \in D_d$	The link set D_d of task d.
$\phi_{(j,e,w')}$	The set of forward links connecting nodes (j, e, w').
Γ_d, Λ_d	Γ_d denotes the set of links where a task is in the waiting state and, Λ_d denotes the set of links where a task is not in the waiting state, respectively. $D_d = \Gamma_d \cup \Lambda_d$
$Cap_{(i,s,w)}, Cap_{(i,j,s,e,w,w')}$	$Cap_{(i,s,w)}, Cap_{(i,j,s,e,w,w')}$ denotes node (i, s, w) capacity and link (i, j, s, e, w, w') capacity, respectively.

2.4. Time-Discretized Multi-Commodity Flow Model

In this section, we build a time-discrete multi-commodity flow model (denoted as M2) for multi-task ISRPTWC. The decision variable of the M2 model is defined as $y(d,i,j,s,e,w,w')$ which takes the value one when a task d is transmitted using the link (i,j,s,e,w,w') and zero otherwise. The M2 model is as follows:

$$Min\ Z = \sum_{d\in D} \frac{1}{Pri_d} \sum_{(i,j,s,e,w,w')\in D_d} c(d,i,j,s,e,w,w')y(d,i,j,s,e,w,w') \tag{13}$$

$$\sum_{(i,j,s,e,w,w')\in D_d} y(d,i,j,s,e,w,w') = 1, \quad (i,s) = st_i,\ w = w_s, \forall d \in D \tag{14}$$

$$\sum_{(i,j,s,e,w,w')\in D_d} y(d,i,j,s,e,w,w') = 1, \quad (j,e) = ar_i,\ w = w_e, \forall d \in D \tag{15}$$

$$\sum_{(j,e,w')} y(d,i,j,s,e,w,w') - \sum_{(i',s',w'')} y(d,i',i,s',s,w'',w) = 0, \quad (i,s) \notin \{st_i, ar_i\}, w \notin \{w_s, w_e\}, \forall d \in D \tag{16}$$

$$\sum_{d\in D} \sum_{(i,j,s,e,w,w')\in \phi_{(j,e,w')}} y(d,i,j,s,e,w,w') \le Cap_{(j,e,w')} \tag{17}$$

$$\sum_{d\in D} y(d,i,j,s,e,w,w') \le Cap_{(i,j,s,e,w,w')} \tag{18}$$

$$y(d,i,j,s,e,w,w') \in \{0,1\} \quad \forall (i,j,s,e,w,w') \in D_d, \forall d \in D \tag{19}$$

The objective function (13) is to minimize the total transmission cost. Constraints (14), (15), and (16) represent flow balance constraints in the time-space state network. Constraint (17) represents the node capacity constraint, and (18) represents the link capacity constraint. Constraint (19) indicates that the decision variables are binary.

Intuitively, we can find that the time-dependent constraints are dissipated in the M2 model because they can be ensured not to be violated in the time-space state network. As shown in Table 4, the time-discrete multi-commodity flow model (M2) contains all constraints of the integer linear programming model (M1).

Table 4. Comparison of M1 model and M2 model constraints.

M1 Model	M2 Model
The decision variable is x_{ij}^d.	The decision variable is $y(d,i,j,s,e,w,w')$.
The objective function (M1-1) is the minimum transmission cost.	The objective function (M2-13) is the minimum transmission cost.
(M1-2) and (M1-3) are to ensure that each task start node and end node are matched.	(M2-14) and (M2-16) are to ensure that each task start node and end node are matched.
(M1-4) is to ensure that each task is transmitted.	(M2-14) is to ensure that each task is transmitted.
(M1-4) and (M1-6) are to ensure the flow balance constraint of the network.	(M2-14) and (M2-16) are to ensure the flow balance constraint of the network
(M1-6) represents that each task is transmitted to the termination node.	(M2-16) represents that each task is transmitted to the termination node.
(M1-7) represents each task must be transmitted to the terminating node within the maximum transmission time.	Defined task transmission time using time-space state network.
(M1-8) represents the capacity of each node to be less than the maximum capacity.	(M2-17) represents the capacity of each node to be less than the maximum capacity.
(M1-9) constrains the maximum congestion time of each task.	Dissipating the maximum congestion time constraint according to the task state in the time-space state network (Theorem 1).
(M1-10) represents the capacity of each link to be less than the maximum capacity.	(M2-18) represents the capacity of each link to be less than the maximum capacity.
(M1-11) represents that if a link capacity is 0, then it cannot transmit tasks.	Time-space state network structure ensures a set of feasible links.
(M1-12) defines the range of values of the decision variables.	(M2-19) defines the range of values of the decision variables.

3. Lagrangian Relaxation-Based Solution Approach

The purpose of the constructed time-space state network and time-discrete multi-commodity flow model is not only to simplify the problem constraints but also to construct a more suitable solution algorithm. In this section, we design a Lagrangian relaxation-based solution framework and further introduce the dynamic programming algorithm for solving the lower bound of the multi-task ISRPTWC and the feasible solution generation method for solving the upper bound of the multi-task ISRPTWC.

3.1. Solution Framework

As shown in Figure 4, we extend the Lagrangian relaxation framework (Algorithm 1.) to solve the multi-task ISRPTWC. Herein, the inter-satellite topology information and the set of tasks are the inputs, the time-space state network is constructed, and the algorithm parameters are initialized based on the input information. Then, after using the reduced solution space approach, a set of subproblems of the multi-task ISRPTWC are solved by the proposed dynamic programming algorithm to obtain the lower bound cost. After calculating the solutions to the subproblems, the nodes and links that violate constraints are eliminated using the feasible solution construction method, and the upper bound cost is calculated. Finally, the Lagrange multiplier is updated by the subgradient formula and determines whether the termination condition is reached. The specific steps are as follows.

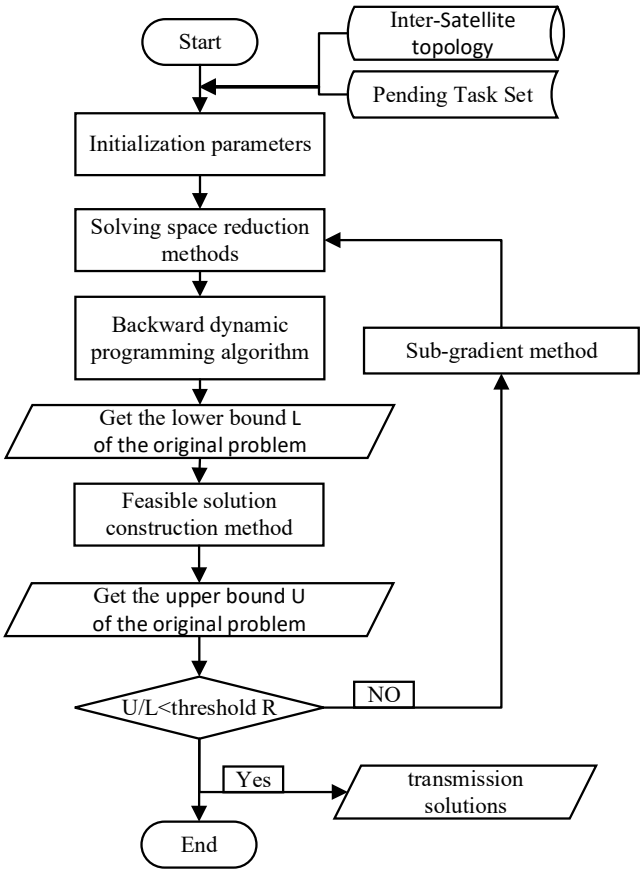


Figure 4. Lagrangian relaxation solving framework.

Algorithm 1. Lagrangian relaxation-based solution framework.

// Initialization parameters.
Set the iterations k;
Initialize LB_0, UB_0 and $\lambda_1(n), \lambda_2(e)$ to zero;
Initialize $LB^* = -\infty$, $UB^* = +\infty$, $\theta^0 \in [0, 2]$;
Define termination conditions:(1) Reaching the maximum iterations.
(2) Gap of the lower bound and the upper bound $\leq 1\%$;
// Constructing a time-space state network based on the input information.
While termination condition is false:
For each iteration k do:
// Generate and update LB_k
Use the dynamic programming algorithms to find the lowest cost route for each task; // The algorithm is in Section 3.3
Update LB^* by $\max(LB_k, LB^*)$;
// Generate and update UB_k.
Use the feasible solution construction method to find the lowest cost route for all tasks while meeting all constraints; // The method is in Section 3.4
Update UB^* by $\min(UB_k, UB^*)$;
// Subgradient calculation. // The calculation process is shown in Section 3.2.
Calculate the capacity of each node and link using expression (20);

$$\begin{cases} \sum\limits_{d \in D} \sum\limits_{(i,j,s,e,w,w') \in \phi_{(j,e,w')}} y(d,i,j,s,e,w,w') \\ \sum\limits_{d \in D} y(d,i,j,s,e,w,w') \end{cases} \tag{20}$$

Calculate the subgradient by equation (21) and record all non-zero subgradients to the set S_t;

$$\begin{cases} \nabla\lambda_1(n) = \max\left\{ \sum\limits_{d \in D} \sum\limits_{(i,j,s,e,w,w') \in \phi_{(j,e,w')}} y(d,i,j,s,e,w,w') - Cap_{(j,e,w')}, 0 \right\} \\ \nabla\lambda_2(e) = \max\left\{ \sum\limits_{d \in D} y(d,i,j,s,e,w,w') - Cap_{(i,j,s,e,w,w')}, 0 \right\} \end{cases} \tag{21}$$

Update all node and link multipliers through equation (22);

$$\begin{cases} \lambda_1(n) = \max\left\{ \lambda_1(n) + \theta^k \nabla\lambda_1(n), 0 \right\} \\ \lambda_2(e) = \max\left\{ \lambda_2(e) + \theta^k \nabla\lambda_2(e), 0 \right\} \end{cases} \tag{22}$$

// Since all nodes are not merged into the cost during the calculation, we transfer the cost of all nodes to the cost of the corresponding links, updating the cost using equation (23).

$$\xi(d,i,j,s,e,w,w') = \frac{1}{Pri_d} c(d,i,j,s,e,w,w') + \lambda_1(n) + \lambda_2(e) \tag{23}$$

Update the gradient step using equation (24);

$$\theta^{k+1} = \frac{UB_k - LB_k}{|S_t|_2} \theta^k \tag{24}$$

// Calculate the gap value and update the iteration k.
The gap between LB^* and UB^* is obtained by $\frac{UB^* - LB^*}{UB^*} \times 100$, $k = k + 1$;
end
end

3.2. Lagrangian Relaxation Solution Procedure

Although the time-discrete multi-commodity flow model can simplify the time-dependent constraints in multi-task ISRPTWC, the capacity constraints of nodes and links have been unresolved complex constraints. Meanwhile, if all node and link constraints are eliminated, the multi-task ISRPTWC becomes a set of single-task ISRPTWCs, and it can be calculated using the dynamic programming algorithm. Along this line, we reformulate the problem by relaxing this complex constraint (17) and (18) into the objective function and use the Lagrange multipliers, $\lambda_1(n)$ and $\lambda_2(e)$, to build the dual Lagrangian function (25).

$$\begin{aligned} L = & \sum_{d\in D}\frac{1}{Pri_d}\sum_{(i,j,s,e,w,w')\in D_d} c(d,i,j,s,e,w,w')y(d,i,j,s,e,w,w') \\ & + \sum_{n\in N}\lambda_1(n)\left[\sum_{d\in D}\sum_{(i,j,s,e,w,w')\in \phi_{(j,e,w')}} y(d,i,j,s,e,w,w') - Cap_{(j,e,w')}\right] \\ & + \sum_{e\in E}\lambda_2(e)\left[\sum_{d\in D} y(d,i,j,s,e,w,w') - Cap_{(i,j,s,e,w,w')}\right] \end{aligned} \tag{25}$$

Therefore, the Lagrangian relaxation dual model (denoted as M3) can be expressed as follows.

$$Min\ L \tag{26}$$

$$\sum_{(i,j,s,e,w,w')\in D_d} y(d,i,j,s,e,w,w') = 1, \quad (i,s) = st_i,\ w = w_s, \forall d \in D \tag{27}$$

$$\sum_{(i,j,s,e,w,w')\in D_d} y(d,i,j,s,e,w,w') = 1, \quad (j,e) = ar_i,\ w = w_e, \forall d \in D \tag{28}$$

$$\begin{aligned} \sum_{(j,e,w')} y(d,i,j,s,e,w,w') - \sum_{(i',s',w'')} y(d,i',i,s',s,w'',w) = 0,\ (i,s) \notin \{st_i, ar_i\}, \\ w \notin \{w_s, w_e\}, \forall d \in D \end{aligned} \tag{29}$$

$$y(d,i,j,s,e,w,w') \in \{0,1\} \quad \forall (i,j,s,e,w,w') \in D_d, \forall d \in D \tag{30}$$

So far, we have obtained a set of single-task ISRPTWCs by the Lagrangian relaxation process. Then, we further process the function L to get the connection of Lagrange multipliers with node and link costs. The derivation process of the simplified function L and the simplified results are shown below.

The links consist of a virtual link formed by nodes with the same satellite attributes in the front and back consecutive time slices and a physical link constructed by nodes with different satellite attributes in the front and back consecutive time slices, and there is no capacity limit for the virtual link. Thus, it is necessary to divide all links in function L into two categories.

$$\begin{aligned} L = & \sum_{d\in D}\sum_{(i,j,s,e,w,w')\in D_d}\frac{1}{Pri_d}c(d,i,j,s,e,w,w')y(d,i,j,s,e,w,w') + \sum_{d\in D}\sum_{(i,j,s,e,w,w')\in \Gamma_d}\lambda_1(n)y(d,i,j,s,e,w,w') \\ & + \sum_{d\in D}\sum_{(i,j,s,e,w,w')\in \Lambda_d}\lambda_1(n)y(d,i,j,s,e,w,w') - \sum_{n\in N}\lambda_1(n)Cap_{(j,e,w')} + \sum_{d\in D}\sum_{(i,j,s,e,w,w')\in D_d}\lambda_2(e)y(d,i,j,s,e,w,w') \\ & - \sum_{(i,j,s,e,w,w')\in D_d}\lambda_2(e)Cap_{(i,j,s,e,w,w')} \end{aligned} \tag{31}$$

Combining links with different attributes in function L.

$$\begin{aligned} L = & \sum_{d\in D}\sum_{(i,j,s,e,w,w')\in \Gamma_d} y(d,i,j,s,e,w,w') \times \left[\frac{1}{Pri_d}c(d,i,j,s,e,w,w') + \lambda_1(n) + \lambda_2(e)\right] \\ & + \sum_{d\in D}\sum_{(i,j,s,e,w,w')\in \Lambda_d} y(d,i,j,s,e,w,w') \times \left[\frac{1}{Pri_d}c(d,i,j,s,e,w,w') + \lambda_1(n) + \lambda_2(e)\right] \\ & - \left[\sum_{n\in N}\lambda_1(n)Cap_{(j,e,w')} - \sum_{(i,j,s,e,w,w')\in D_d}\lambda_2(e)Cap_{(i,j,s,e,w,w')}\right] \end{aligned} \tag{32}$$

Let $\xi(d,i,j,s,e,w,w') = \frac{1}{Pri_d}c(d,i,j,s,e,w,w') + \lambda_1(n) + \lambda_2(e)$ be the new transmission cost.

$$\begin{aligned} L = & \sum_{d\in D}\sum_{(i,j,s,e,w,w')\in \Gamma_d} \xi(d,i,j,s,e,w,w')y(d,i,j,s,e,w,w') \\ & + \sum_{d\in D}\sum_{(i,j,s,e,w,w')\in \Lambda_d} \xi(d,i,j,s,e,w,w')y(d,i,j,s,e,w,w') \\ & - \left[\sum_{n\in N}\lambda_1(n)\mathrm{Cap}_{(j,e,w')} - \sum_{(i,j,s,e,w,w')\in D_d}\lambda_2(e)\mathrm{Cap}_{(i,j,s,e,w,w')}\right] \end{aligned} \tag{33}$$

The final simplified result of the function L is obtained.

$$L = \sum_{d \in D} \sum_{(i,j,s,e,w,w') \in D_d} \xi(d,i,j,s,e,w,w') y(d,i,j,s,e,w,w') - \left[\sum_{n \in N} \lambda_1(n) Cap_{(j,e,w')} - \sum_{(i,j,s,e,w,w') \in D_d} \lambda_2(e) Cap_{(i,j,s,e,w,w')} \right] \tag{34}$$

Notably, the function L in $\xi(d,i,j,s,e,w,w')$ is composed of the multiplying $\frac{1}{Pri_d} c(d,i,j,s,e,w,w')$ of a link cost and an inverse of a task priority, the multiplier $\lambda_1(n)$ of the link connected nodes and the multiplier $\lambda_2(e)$ of the corresponding link. Therefore, the cost $\xi(d,i,j,s,e,w,w')$ changes as the multipliers are updated, and this also affects the results of the dynamic programming algorithm for solving the single-task subproblem. This process ensures that the ideal lower-bound solution does not violate the capacity constraint and guarantees the feasibility of this method.

3.3. Dynamic Programming Algorithm

In this subsection, we use a dynamic programming algorithm (Algorithm 2.) to solve the shortest route problem with time window constraints, which is a subproblem of the multi-task ISRPTWC, where the constructed time-space state network can nicely ensure that the time-dependent constraints are not violated. According to the time-discrete multi-commodity flow model in Section 2.4, we define $L(.,.,.)$ is denoted as the vertex $(.,.,.)$, including $L\left(o_d^*, s_d^*, w^*\right)$ denoted as the starting node of task d, and $L\left(t_d^*, e_d^*, w'^*\right)$ denoted as the termination node of task d. The specific dynamic programming algorithm flow is as follows.

Single-task ISRPTWC can be solved well by the dynamic planning algorithm, but it is difficult to check for task congestion during transmission, i.e., tasks are transmitted continuously on nodes with the same satellite attributes over multiple time slices. Therefore, we add the process of checking a task transmission route at the end of the dynamic planning algorithm. If a task transmission route causes congestion, we go back to the first congestion-generating node and recalculate its optimal route while avoiding using subsequent congested nodes. To ensure the optimality of the backtracking method, we combine it with the relevant theorem of the dynamic programming algorithm to prove it, as shown in Theorem 1.

Theorem 1. *If a transmission route of a task d reaches the upper limit of the congestion Blo_d, it is sufficient to go back to the first node (i,s,w) that caused the congestion on its optimal route S_d^* and modify the connection structure of node $S_d^*(n)$ to recalculate its optimal route.*

Proof. Suppose the current set of routes of the task d is D_d, the optimal sequence of task d is $S_d^* \subseteq \mathrm{D}_d$, and the set of nodes causing congestion is $\{(i,s,w),(i,s+1,w),\cdots,(i,s+Blo_d,w)\} \subseteq S_d^*$. □

Taking node $(i,t,w), t \in [s, s+Blo_d]$ as the breakpoint. That is, the connection link $l_{(i,t,w)}$ between node$(i,t-1,w)$ and node (i,t,w) is destroyed, and the set of destroyed links is viewed to be $\frac{\mathrm{D}}{l_{(i,t,w)\ d}}$. According to the no-aftereffect characteristic in dynamic planning, the optimality of the route $\{(i,t,w),\cdots,destination\} \subseteq S_d^*$ is not affected in backward planning and thus does not need to be recalculated. However, the optimality of the route $\{origin,\cdots,(i,t,w)\} \subseteq S_d^*$ is broken and needs to be recomputed. Moreover, we obviously know that the fewer route structures are destroyed, the smaller the impact on the routes D_d. Therefore, we must find the node that minimizes the damage while meeting the maximum congestion time constraint. That is, we must find the maximally effective routes (35).

$$\max\left(\frac{\mathrm{D}_{eff}}{l_{(i,t,w)\ d}}\right), t \in [s, s+Blo_d] \tag{35}$$

The effective route set $\frac{D_{eff}}{I_{(i,t,w)}}_d$ is both the set of routes that the task d can transmit from the origin node to the termination node. Therefore, we can obviously conclude that $\frac{\mathrm{D}_{eff}}{I_{(i,t,w)}}_d \subseteq \frac{\mathrm{D}_{eff}}{I_{(i,t-1,w)}}_d$, so that the breakpoint $(i,t,w) = (i,s,w)$ can make $\frac{\mathrm{D}_{eff}}{I_{(i,t,w)}}_d$ maximize.

In summary, the proof is over.

Algorithm 2. Dynamic programming algorithms.

For each task $d \in D$ do
  calculate the set $D_d \subseteq D_*$ of possible routes for the task d;
  // initialization of relevant parameters.
$L(.,.,.) := +\infty$; // the cost of the initial value of each node in the route is infinity;
original vertex $L\left(o_d^*, s_d^*, w^*\right)$ of the task d;
for $e_d^* \in \left[s_d^* + 1, s_d^* + Dur_d\right]$ do
get the terminating vertex $L\left(t_d^*, e_d^*, w'^*\right) := 0$ of the task d;
for each time $e \in \left[e_d^*, s_d^*\right]$ do
  for each link (j,i) do
    for each state w' do
      determine the state of the last vertex w from the state of the current vertex w';
        if $L(j,e,w') + \xi(d,i,j,s,e,w,w') \leq L(i,s,w)$ then
          $L(i,s,w) = L(j,e,w') + \xi(d,i,j,s,e,w,w')$;
          update node i, time s, state w by using node j, time e, state w';
          record vertex (j,e,w') to S_d^*;
        end;
      end;
    end;
  end;
  check if the maximum congestion time $Blo\left(S_d^*\right) \leq Blo_d$; // according to theorem 1
 end;
end;

3.4. Feasible Solution Generation Methods

Since the optimal route for a single task is computed by a dynamic planning algorithm without considering capacity constraints, this solution is essentially infeasible in multi-task ISRPTWC. Therefore, we design some rules to adjust the infeasible solutions into feasible ones. Current studies [42–44] involve the construction of feasible solutions using methods that are basically heuristic rule constructions. However, it is difficult to construct a tight upper bound by heuristic rules. To address this problem, we newly combine heuristic rules with an integer linear programming model to find feasible solutions to ensure computational efficiency. Specifically, we view the feasible solution construction process as a new combinatorial optimization problem.

(1) Enhanced lower bound solution for integer linear model

As shown in Figure 5, the transmission routes of the tasks may exist with partial overlap due to the capacity constraints of nodes and links, which leads to violating the capacity constraints. Therefore, we first compute the set of nodes P and the set of links B that violate the constraints in all task route schemes. Then, we define the D_d as a set of the current routes of the task d, D'_d as the set of routes obtained by removing some nodes and links in the set of capacity constraints violated by task d, and the route computed in D_d is called the optimal route scheme of task d, and the route computed in D'_d is called the suboptimal route scheme of task d. The route cost N_{d0} of the optimal route solution for task d is known, and we define the cost N_{dl} of the suboptimal route solution for task d to consist of two parts, including its cost N^*_{dl} incurred in D'_d and the degree of global impact N'_{dl} of the new route, in which the degree of the global impact is mainly the impact of the new route scheme of task d on other task route schemes. This effect causes other nodes and links that do not violate the constrained routes to violate the capacity constraint due to modified task routes. Thus, the feasible solution construction problem is how to find routes for each task that reduce the number of nodes and links that violate the constraint while increasing the total route cost as little as possible. The specific sets, parameters, and decision variables are shown in Table 5.

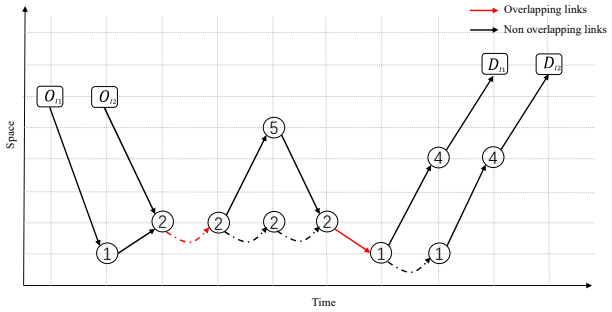


Figure 5. Explanation of tasks transmission route.

Table 5. Sets, parameters, variables.

Symbol	Definition
D_d, D'_d	D_d represents the current set of routes of task d and D'_d represents the set of routes of task d after removing the nodes and links in the set of violated constraints corresponding to $x_{dk} = 1$.
N_{d0}	N_{d0} represents the route cost of the optimal route for task d under D_d.
$N_{dl}, N^*_{dl}, N'_{dl}$	N^*_{dl} represents the route cost required for the optimal route l of task d under D'_d; N'_{dl} represents the degree of influence of the optimal route l of task d under D'_d on the optimal route of other data; $N_{dl} = N^*_{dl} + N'_{dl}$ represents the total cost of task d after modifying the optimal route. $l \in \left[1, 2^{\lvert \mathrm{p}_d \rvert + \lvert b_d \rvert} - 1\right]$
N_d	N_d represents the set of route costs for task d.
P, B	P, B represent the set of nodes and the set of links that violate the constraints, respectively.
p_d, b_d	p_d, b_d represent the set of nodes and the set of links for which the optimal route scheme of task d violates the constraints. $\mathrm{P}_d \subseteq P, b_d \subseteq B$
$Cap_{(i,s,w)}$, $Cap_{(i,j,s,e,w,w')}$	$Cap_{(i,s,w)}, Cap_{(i,j,s,e,w,w')}$ represent the node capacity and link capacity, respectively.
$\Gamma_{(i,s,w)}$	The set of tasks that violate the node (i, s, w) constraint.
$\mathrm{Y}_{(i,j,s,e,w,w')}$	The set of tasks that violate the link (i, j, s, e, w, w') constraint.
x_{dk}	x_{dk} represents whether the kth node or link in the set of routes of task d that violates the constraint is removed, $k \in \left[0, \lvert \mathrm{p}_d \rvert + \lvert b_d \rvert - 1\right]$.
y_{dl}	y_{dl} represents the lth route in the set of routes for task d that is used. If $y_{d0} = 1$, the optimal route is selected, $l \in \left[1, 2^{\lvert \mathrm{p}_d \rvert + \lvert b_d \rvert} - 1\right]$

This integer linear programming model (denoted as M4) is formulated as follows.

$$Min \sum_{d \in D} \sum_{l \in [0, 2^{|p_d|+|b_d|}-1]} y_{dl} N_{dl} \tag{36}$$

$$\sum_{d \in \Gamma_{(i,s,w)}} x_{d,(i,s,w)} = Cap_{(i,s,w)}, \forall (i, s, w) \in P \tag{37}$$

$$\sum_{d \in Y_{(i,j,s,e,w,w')}} x_{d,(i,j,s,e,w,w')} = Cap_{(i,j,s,e,w,w')}, \forall (i, j, s, e, w, w') \in B \tag{38}$$

$$\sum_{k \in [0, |p_d|+|b_d|-1]} 2^k x_{dk} = \sum_{l \in [0, 2^{|p_d|+|b_d|}-1]} y_{dl} l, \quad d \in D \tag{39}$$

$$\sum_{l \in [0, 2^{|p_d|+|b_d|}-1]} y_{dl} = 1, \quad d \in D \tag{40}$$

$$x_{dk}, y_{dl} \in \{0, 1\}, d \in D, k \in [0, |p_d| + |b_d| - 1], \; l \in \left[0, 2^{|p_d|+|b_d|} - 1\right] \tag{41}$$

The objective function (36) expresses the lowest total route cost for all tasks. Constraints (37) and (38) are to eliminate the nodes and links that currently violate the capacity constraint, respectively. Constraint (39) represents the relationship between the decision variables y_{dl} and x_{dk} taking values. For example, task d has five links and nodes that violate the constraints. When $[x_{d0}, x_{d1}, x_{d2}, x_{d3}, x_{d4}]$ of task d is [0,0,0,0,0,1], then the current

decision variable y_{d1} is 1. In other words, the selected route scheme is computed in D'_d of the node or link corresponding to the deletion constraint x_{d4}. The range of values of the decision variables is defined in constraints (40) and (41).

We use the CPLEX solver to solve the M4 model. Although the solutions obtained from the integer linear model are not feasible, it certainly converges toward the direction of constructing feasible solutions. Thus, with a reasonable time, we can repeat the construction of the M4 model and solve it using the solver.

(2) Heuristic methods to construct feasible solutions

Although the M4 model has largely reduced the number of nodes and links that violate the capacity constraint, the obtained route schemes are still not feasible solutions. Therefore, in this section, we design heuristics for feasible solution construction. The purpose of the constructed M4 model is to reduce the constraints as much as possible while controlling the route cost. However, in contrast to the M4 model idea, the main idea of the heuristic methods is to reduce the route cost as much as possible while eliminating the nodes and links that violate the constraints. In the process of constructing the feasible solution, we analyze the relationship between the two types of costs of the suboptimal route and the two types of costs of the optimal route of a task.

Then, different processing strategies apply to different situations. The specific steps are as follows.

Step1: Obtain the set of task routes, and calculate the optimal route and the number of constraint violations for each task.

Step2: Calculate the suboptimal route and the degree of global impact for each task.

Step3: Compare the suboptimal route cost and global impact degree of each task, find the optimal task adjustment scheme, and record this adjustment scheme into the tabu table to prevent subloops.

Step3.1: If there is a task with a null set of violated constraints, then there is no need to compute a suboptimal route.

Step3.2: If there exists a task whose suboptimal route cost is the same as the optimal route cost and the global impact of the suboptimal route is lower than the global impact of the optimal route, the optimal route of the task replaces by the suboptimal route.

Step3.3: If there are tasks with a suboptimal route cost greater than the optimal route cost, and the global impact of the sub-optimal route is less than the global impact of the optimal route, then we will select the sub-optimal route with the least global impact among these tasks, and record the nodes and links of the task that violate the optimal route in the tabu table.

Step3.4: If the cost of the suboptimal route of each task is greater than the cost of the optimal route, and the global impact of the suboptimal route is not lower than the global impact of the optimal route, the optimal route of all tasks is invariant. At the same time, we find the task corresponding to the optimal route with the largest global impact and record the nodes and links whose optimal routes of this task violate the constraints in the forbidden table.

Step3.5: If there exists a task that cannot get a suboptimal route, the cost of the task's suboptimal route is set to positive infinity, and the degree of global impact is set to positive infinity to ensure that its optimal route is not modified.

Step3.6: If all tasks do not get the suboptimal route, the task with the optimal route that has the greatest global impact is deleted.

Step4: If the set of violated constraints is not null, then return to Step2; otherwise, end.

4. Search Space Reduction for Tackling Computational Challenges on Large-Scale Networks

The dynamic programming algorithm and the feasible solution construction method in the proposed Lagrangian relaxation framework are time-consuming when dealing with large-scale problems. Some search space reduction methods can deploy to reduce computational challenges in large-scale networks. Considering that the single-task route

transmission process is mainly limited by time-dependent constraints, we divide the boundaries of possible transmission nodes of a task according to its start time and termination time. In other words, the set of possible transmission nodes for each task is considered a non-complete binary tree. Meanwhile, we record the non-complete binary trees of all tasks, calculate the overlap between all binary trees, and save the nodes and links that may violate the capacity constraint into the set. Finally, only the nodes and links in this set are considered when checking the capacity constraint.

On the other hand, we find that the search space of the M4 model can be further reduced. Specifically, we first explain the dual-route concept. The dual-route is defined as: for any task d, if the cost of the optimal route under its current route set D_d is N^*_{d0}, and there exists a suboptimal route with route cost $N^*_{d0_{dual}}$ and $N^*_{d0_{dual}} - N^*_{d0} \leq \varepsilon$ (ε is the cost difference threshold), while the global impact degree of this suboptimal route is smaller than the global impact degree of the optimal route, the suboptimal route corresponding to route cost $N^*_{d0_{dual}}$ is called the dual-route of the current optimal route. Due to the large number of tasks, the size of the inter-satellite network, and the existence of task aggregation, the possibility of a dual route for each task is very high. Therefore, we check and record the dual-routes for each task before proceeding with the feasible solution construction process. If a dual-route exists for a task, the optimal route for the task is replaced, while the decision variable $y_{d(l=dual)}$ in the M4 model is determined directly. This process greatly reduces the search space of the solver when computing the M4 model. In addition to the parameter definitions in model M4, we have added some parameters, shown in Table 6.

Table 6. Parameter Definition.

Symbol	Definition
$Data_{dual}$	$Data_{dual}$ represents the set of tasks for which there are dual-routes, $Data_{dual} \subseteq \zeta^{vio}$.
$N_{d0_{dual}}$	$N_{d0_{dual}}$ represents the total cost of task d dual-routes.
$N^*_{d0_{dual}}$	$N^*_{d0_{dual}}$ represents the cost of routes with global effects removed.
ζ^{vio}	ζ^{vio} represents the set of tasks for routes that violate node and link constraints.

Therefore, we add the redundancy constraint (47) to the M4 model. Because the dual-route of task d must exist in the set of suboptimal routes, we directly constrain the decision variable $y_{d(l=dual)}$ corresponding to the dual-route to reduce the solution time of the model. The specific improved model (denoted as M5) is shown below.

$$Min \sum_{d \in D} \sum_{l \in [0, 2^{|p_d|+|b_d|}-1]} y_{dl} N_{dl} \tag{42}$$

$$\sum_{d \in \Gamma_{(i,s,w)}} x_{d,(i,s,w)} = Cap_{(i,s,w)}, \forall (i,s,w) \in P \tag{43}$$

$$\sum_{d \in Y_{(i,j,s,e,w,w')}} x_{d,(i,j,s,e,w,w')} = Cap_{(i,j,s,e,w,w')}, \forall (i,j,s,e,w,w') \in B \tag{44}$$

$$\sum_{k \in [0, |p_d|+|b_d|-1]} 2^k x_{dk} = \sum_{l \in [0, 2^{|p_d|+|b_d|}-1]} y_{dl} l, \quad d \in D \tag{45}$$

$$\sum_{l \in [0, 2^{|p_d|+|b_d|}-1]} y_{dl} = 1, \quad d \in D \tag{46}$$

$$y_{d(l=dual)} = 1, d \in Data_{dual} \tag{47}$$

$$x_{dk}, y_{dl} \in \{0,1\}, d \in D, k \in [0, |p_d| + |b_d| - 1],\ l \in \left[0, 2^{|p_d|+|b_d|} - 1\right] \tag{48}$$

5. Numerical Experiments

The algorithm proposed in this paper is coded in Java 1.8.0. The experiments are conducted using Intel (R) Core (TM) i5-10210U CPU at 1.60 GHZ in Windows 10 with 16 GB

of RAM. Since the inter-satellite routing problem for low-orbit mega constellations has been little studied and multi-task ISRPTWC has not been specifically proposed in the literature, no publicly available datasets and algorithms can be used and compared. Therefore, in this section, we use the illustrative dataset, the randomly generated dataset, and the realistic constellation network dataset to validate our proposed model and algorithm, respectively. Specifically, we initially examine our model and algorithm for a 20×10 inter-satellite transmission network (20 timeslots, 10 satellites) and illustrate the solution principle of the algorithm. Then, we use ten randomly generated datasets to demonstrate the proposed algorithm solution efficiency and further test our novel approach to feasible solution construction. Finally, we obtain publicly available Starlink constellation data through various simulation tools and build a medium-scale network of 500 satellites and a super-scale network of 900 satellites to confirm the computational efficiency and solution quality of our algorithm. It is worth noting that the topology of the constellation network in all the cases is from our previous work [45]. However, to ensure the authenticity of all experiments, the datasets and codes designed for this experiment can be accessed on 1 October 2022 at https://github.com/Sean-Ming/inter-satellite-routing-algorithm.

5.1. Illustrative Cases

In the subsection, we aim to provide an explanatory illustration of the proposed model and algorithm. Hence, support that the node and link capacities in the network are limited to constant respectively, the priority of all tasks is defined as one, and all tasks have four upload windows and eight downlink windows. Initially, we test our algorithm on the 20×10 time-space state network shown in Figure 3, including a total of six cases, the specific descriptions of which are shown in Table 7. Table 8 shows partial task-related information for different scenarios.

Table 7. Scenario Description.

Scenario	Specific Instructions
Scenario I	A total of 2 tasks are required, with network node and link capacities capped at 2 and 1. The 2 tasks have the same start and end nodes, the start and end time windows are overlapping, and the inter-satellite network transmission costs are different.
Scenario II	A total of 2 tasks are required, with network node and link capacities capped at 2 and 1. The 2 tasks are randomly generated task start and end nodes, with overlapping start and end time windows and identical inter-satellite network transmission costs.
Scenario III	A total of 2 tasks are required, with network node and link capacities capped at 2 and 1. The 2 tasks have the same starting and ending nodes, overlapping starting and ending time windows, and identical inter-satellite network transmission costs.
Scenario IV	A total of 18 tasks are required, with network node and link capacities capped at 3 and 2. Priority of all tasks is 1, and task repetition is approximately 30%. Task repetition is the percentage of all tasks in a task set that have the same time window or the same network cost.
Scenario V	A total of 18 tasks are required, with network node and link capacities capped at 3 and 2. Priority of all tasks is 1, and task repetition is approximately 60%.
Scenario VI	A total of 18 tasks are required, with network node and link capacities capped at 3 and 2. Priority of all tasks is 1, and task repetition is approximately 100%.

Table 8. Description of some tasks in the scenarios.

Scenario	Number of Tasks	Task 1				Task 2			
		Start Time Window	End Time Window	Start Node Set	Terminate Node Set	Start Time Window	End Time Window	Start Node Set	Terminate Node Set
Scenario I	2	[3,5]	[7,15]	{2,5}	{2,3,4,5,6,7,8}	[3,5]	[7,15]	{2,5}	{2,3,4,5,6,7,8}
Scenario II	2	[4,6]	[8,16]	{2,1}	{2,3,4,5,6,7}	[4,6]	[8,16]	{2,6}	{2,3,4,5,6,7,8}
Scenario III	2	[4,6]	[8,16]	{2,6}	{2,3,4,5,6,7,8}	[4,6]	[8,16]	{2,6}	{2,3,4,5,6,7,8}
Scenario IV	18	[3,5]	[7,15]	{2,5}	{2,3,4,5,6,7,8}	[1,3]	[5,13]	{4,9}	{1,2,3,4,5,7,8}
Scenario V	18	[3,5]	[7,15]	{2,5}	{2,3,4,5,6,7,8}	[5,7]	[9,17]	{4,9}	{1,2,3,4,5,7,8}
Scenario VI	18	[3,5]	[7,15]	{2,5}	{2,3,4,5,6,7,8}	[8,10]	[12,19]	{4,9}	{1,2,3,4,5,7,8}

As shown in Figure 6, all task transmission routes for the six scenarios are shown. We set up Scenarios I–III to test the algorithm's ability to handle tasks with similar attributes or the same attributes. In Scenarios I and II, we find that the proposed algorithm can handle tasks with the same time attributes or with the same cost attribute very well. Additionally, our algorithm can handle "mirror tasks" with the same time and cost attributes in Scenario III because of the proposed dual routes in the reduced search space approach, which can solve the same tasks well. Then, we set up Scenarios IV–VI to test the ability of the proposed algorithm to handle some concentrated tasks. From the experimental results, it is concluded that the proposed algorithm is able to obtain better solutions under capacity constraints when dealing with high-density task sets. It is worth noting that no task in the task transmission route scheme is transmitted three times by nodes with the same satellite attribute (the experimental congestion threshold is 3), demonstrating the ability of the algorithm to avoid packet loss due to congestion. The specific experimental results are present in Table 9. From the planning results of the six scenarios, we find that the gap between the upper and lower bounds of the solutions is small in each scenario. This phenomenon does not indicate that the proposed algorithm is invalid, but proves that our proposed method for the construction of feasible solutions is very effective and is able to obtain an excellent upper bound value at the early stage of algorithm planning, which provides more information for the subsequent algorithm convergence process.

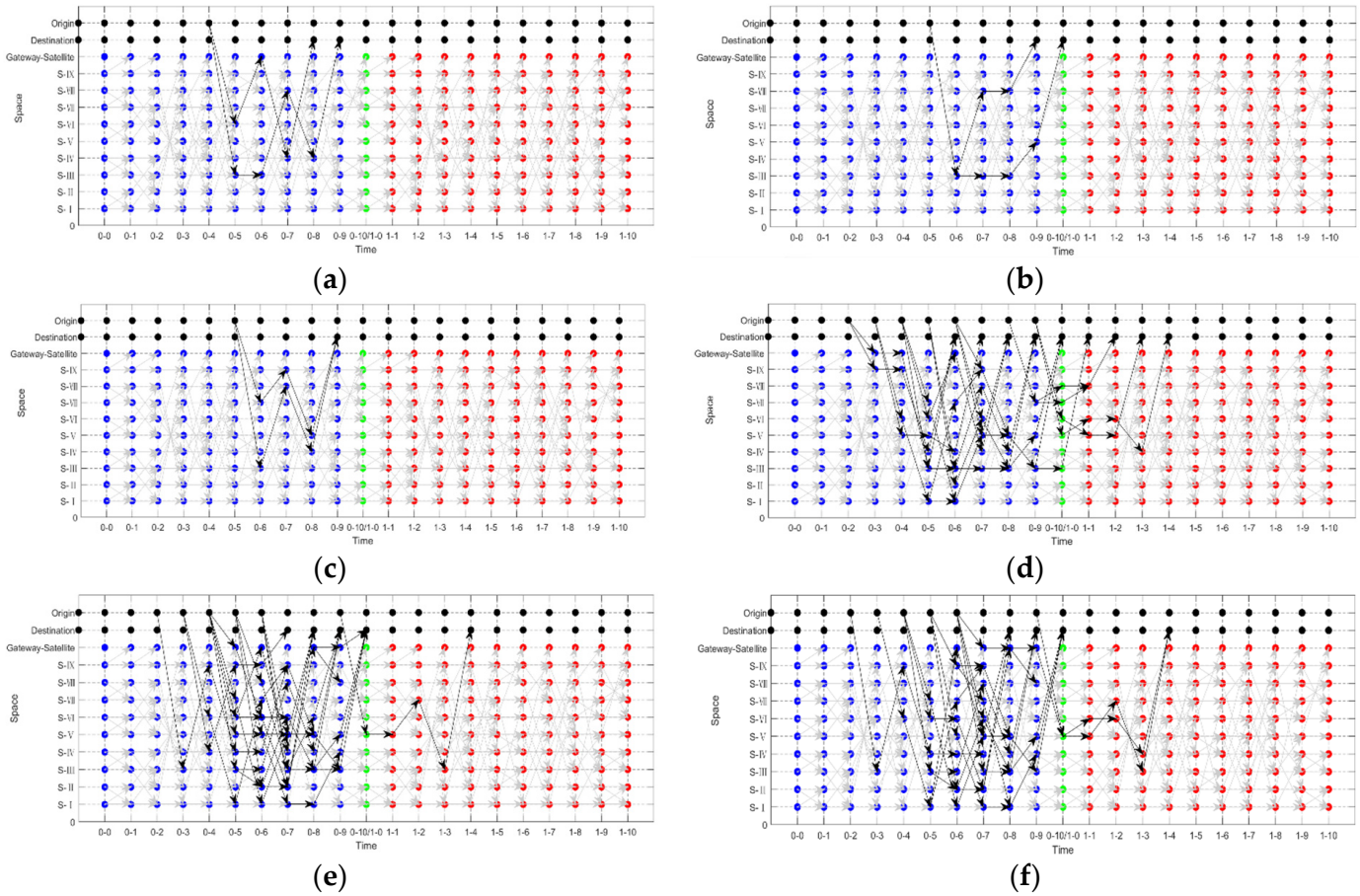


Figure 6. Experimental results for illustrative cases. (**a**) Scenario I. (**b**) Scenario II. (**c**) Scenario III. (**d**) Scenario IV. (**e**) Scenario V. (**f**) Scenario VI.

Table 9. Experimental results of illustrative cases.

Number of Iterations k	LB^*	UB^*	Gap%	Violated Node Constraints	Violated Link Constraints
Scenario I: A total of 2 tasks, network node, and link capacity is upper limit 2 and 1. The time attribute of tasks is the same.					
1	8.87	10.36	14.38%	Null	{[55,69], [69,73]}
2	8.87	10.36	14.38%	Null	{[55,69], [69,73]}
4	10.36	10.36	0%	Null	Null
Scenario II: A total of 2 tasks, network node, and link capacity is upper limit 2 and 1. The tasks have different time attributes and the same cost attribute.					
1	9.38	10.06	6.76%	Null	{[62,77], [77,83]}
2	9.78	10.06	2.78%	Null	{[62,77]}
3	10.06	10.06	0%	Null	Null
Scenario III: A total of 2 tasks, network node, and link capacity are upper limits 2 and 1. The tasks have the same time attribute and cost attribute.					
1	7.32	8.35	12.31%	Null	{[66,78], [78,84]}
2	8.35	8.35	0%	Null	Null
Scenario IV: 18 tasks in total. Network node and link capacities are capped at 3 and 2. Task repetition is about 30%.					
1	65.3	76.89	15.07%	{55,69,73}	{[55,69], [69,73], [96,106]}
2	70.52	74.56	5.42%	{55,69,73}	Null
5	72.87	73.57	0.95%	{74}	{[50,60], [60,74]}
9	73.57	73.57	0%	Null	Null
Scenario V: 18 tasks in total. Network node and link capacities are capped at 3 and 2. Task repetition is about 60%.					
1	71.75	88.83	19.23%	{84,73,69,55,74}	{[55,69], [69,73], [74,84], [84,92]}
6	80.71	87.84	8.12%	{60,74}	{[45,50], [50,60], [60,74], [70,80], [52,62]}
8	81.5	85.07	4.19%	Null	{[52,62], [70,80]}
15	81.5	84.74	3.82%	Null	{[52,62], [70,80]}
20	83.48	84.74	1.49%	Null	{[70,80]}
Scenario VI: 18 tasks in total. Network node and link capacities are capped at 3 and 2. Task repetition is about 100%.					
1	72.28	88.54	18.37%	{82,74,84,73,55,69,92}	{[55,69], [69,73], [74,84], [84,92], [75,82]}
3	77.96	86.23	10.01%	{73,55,69}	{[55,69], [69,73], [70,80], [80,93]}
8	79.02	84.86	6.88%	Null	{[55,69], [69,73], [70,80], [80,93], [52,62]}
11	81.79	84.86	3.61%	Null	{[70,80], [80,93], [52,62]}
18	82.88	84.86	2.33%	Null	{[52,62]}

5.2. Testing Lagrangian Relaxation Framework

We further tested it in a randomly generated case. Specifically, we set up 20×30, 20×100 and 20×200 networks to verify the computational efficiency and solution capability of the algorithm in each scenario, in the case of handling a sparse, regular and overloaded number of tasks (due to the small size of the 20×30 network, a sparse task scenario is not necessary), respectively. The parameters are present in Table 10.

Table 10. Random scenarios parameters.

Scenario	Number of Timeslots	Number of Satellites	Number of Tasks	Node Capacity	Link Capacity
Random_I	20	30	50	2	1
Random_II	20	30	80	2	2
Random_III	20	100	80	2	1
Random_IV	20	100	120	2	2
Random_V	20	100	150	2	2
Random_VI	20	100	180	2	2
Random_VII	20	200	150	2	1
Random_VIII	20	200	180	2	2
Random_IX	20	200	240	2	2
Random_X	20	200	300	2	2

We dynamically adjust the capacity of nodes and links according to the number of tasks and set the proposed algorithm to terminate when the gap value of the optimal upper bound UB^* and the optimal lower bound LB^* is less than 1%. In the small-scale cases, we keep increasing the number of tasks to test the computational efficiency and solution quality of the algorithm, in which the ratio of the number of satellites to the number of tasks for Random_II and Random_VI has reached 3:8 and 1:1.8, and the experimental results demonstrate that the proposed algorithm obtains the optimal solution in an acceptable time. In addition, we find that the running time in Random_X is much longer than in other scenarios due to the excessive number of nodes and links violating the constraints caused by these tasks.

There are 104 nodes and links that violate the constraints at the initial iteration, but our algorithm can still obtain a solution close to the optimal solution. Therefore, the experimental results illustrate that the proposed algorithm still works well in handling overloaded tasks. The specific results are present in Table 11.

Table 11. Experimental results of random scenarios.

Scenario	LB^*	UB^*	Gap%	Number of Iterations	CPU Running Time (s)	Violated Node Constraints	Violated Link Constraints
Random_I	276.55	276.55	0%	10	95.812	Null	Null
Random_II	456.43	459.67	0.75%	20	277.538	{264}	Nullx
Random_III	553.90	553.90	0%	10	119.954	Null	Null
Random_IV	828.34	828.34	0%	27	401.272	Null	Null
Random_V	1055.13	1058.63	0.33%	29	482.773	{606}	Null
Random_VI	1272.36	1276.90	0.36%	30	506.777	{606}	Null
Random_VII	1361.89	1364.34	0.18%	20	305.205	Null	{[1655,1993], [1993,2188]}
Random_VIII	1623.88	1623.88	0%	14	259.703	Null	Null
Random_IX	2146.54	2146.54	0%	20	587.444	Null	Null
Random_X	2757.69	2761.23	0.13%	47	1811.168	{2825,2238}	Null

5.3. Results from Medium-Scale and Large-Scale Transportation Networks

The real-world cases are primarily present and adopted from the Starlink constellation of publicly available datasets. We use the simulation platform to calculate the set of inter-satellite visible relationships in the constellation and construct a 20×500 mesoscale and 20×900 large-scale inter-satellite network where there are 500 satellites and 20,000 links in the mesoscale scenario and 900 satellites and 36,000 links in the large-scale scenario. Meanwhile, the sparse task scenario has been proven to be well solved by the proposed algorithm, so the regular and dense task scenarios are set up separately, as shown in Table 12. In addition, we set the upload time window to 4 and the downlink time window to 8 for each task. The node and link capacity in the network is defined according to the number of tasks. The termination iteration condition of the algorithm is that the LB^* of the subproblems and the UB^* of the original problem reach a small gap. The experimental results are presented in Table 13, where the proposed algorithm can still be solved efficiently by verifying the dense task demand scenario with a large-scale population of satellites. In particular, our algorithm is able to give a near-optimal solution quickly when the tasks are more uniformly distributed. It is worth noting that we set the priority of task requirements to be the same in all experimental cases. However, when the task priorities are different, the degree of differentiation of task attributes will be more obvious and more favorable to the solution.

Table 12. Real scenario parameters.

Scenario	Number of Timeslots	Number of Satellites	Number of Tasks	Node Capacity	Link Capacity
MediumScale_I	20	500	500	2	2
MediumScale_II	20	500	900	3	2
LargeScale_I	20	900	800	2	2
LargeScale_II	20	900	1200	3	2

Table 13. Real-world case experiment results.

Scenario	LB^*	UB^*	Gap%	Number of Iterations	CPU Running Time (s)	Violated Node Constraints	Violated Link Constraints
MediumScale_I	5492.61	5493.40	0%	31	1658.687	{4832}	Null
MediumScale_II	8420.76	8422.13	0%	30	2069.711	{5357}	Null
LargeScale_I	9030.03	9040.10	0.11%	32	3886.633	{6214,7537,4726}	Null
LargeScale_II	13,066.90	13,074.37	0%	25	4372.879	{7943,9731}	{[8444,9483], [8994,9894]}

6. Conclusions

In the context of satellite-ground integration, we investigate the inter-satellite routing problem for low-orbit mega-constellations to design an exact solution to solve it. Thus, we propose a multi-task large-scale inter-satellite routing problem with time windows and capacity constraints (ISRPTWC) and design a Lagrange relaxation solution framework combined with the dynamic programming approach in a time-space state network to solve the ISRPTWC. Frankly, multi-task ISRPTWC is actually a minimum-cost route planning problem considering task time window constraints and inter-satellite network capacity constraints.

In this study, we first build an integer linear programming model of the problem to solve it. However, the problem is almost impossible to solve directly using the solver. Therefore, we reformulate this problem in the constructed time-space state network and propose a time-discrete multi-commodity flow model which can well explain the time, space, and state properties of a task. Additionally, we find that multi-task ISRPTWC is a set of single-task ISRPTWC if capacity constraints are not considered because the single-task ISRPTWC can use a dynamic programming approach to solve it. Therefore, we further propose a Lagrangian relaxation framework to relax the capacity constraint into the objective function by Lagrangian multipliers and design subproblem-solving methods and feasible solution construction methods. Notably, to obtain a high-quality feasible solution, we propose a new idea to consider a feasible solution construction process as a new combinatorial optimization problem and design an integer linear programming model and heuristic rules to combine to solve it. Finally, we validate the proposed algorithm using illustrative cases, randomly generated cases, and real-world, large-scale network cases. The computational results show that our algorithm can solve the optimal solution of the multi-task ISRPTWC.

In conclusion, this algorithm provides a new idea for solving multi-mission ISRPTWC, which is simple and reliable and can be applied to more complex inter-satellite routing problems. Our future research will focus on the following three areas: (1) considering more complex inter-satellite routing problems in satellite networks, such as multi-layer satellite networks and heterogeneous satellite networks; (2) considering more complex task requirement constraints, such as certain tasks can only be transmitted between specific satellites; (3) applying the idea of feasible solution construction to solve other types of transportation problems in combination with related exact solutions.

Author Contributions: Conceptualization, J.L. and L.X.; methodology, J.L.; software, W.Q.; validation, J.L. and G.Z.; resources, Y.C.; data curation, G.Z.; writing—original draft preparation, J.L.; writing—review and editing, L.X. and W.Q.; visualization, J.L.; supervision, Y.C.; funding acquisition, L.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (61773120), The General Program of the National Natural Science Foundation of Hunan Province (2022JJ30671), and the Hunan Postgraduate Research Innovation Project (CX20210083).

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

References

1. Bogdan-Martin, D. *Measuring Digital Development: Facts and Figures*; Technical Report; International Telecommunications Union (ITU): Geneva, Switzerland, 2019.
2. Del Portillo, I.; Eiskowitz, S.; Crawley, E.F. Connecting the other half: Exploring options for the 50% of the population unconnected to the internet. *Telecommun. Policy* **2021**, *45*, 102092. [CrossRef]
3. Chen, Q.; Chen, X.; Yang, L. A distributed congestion avoidance routing algorithm in mega-constellation network with multi-gateway. *Acta Astronaut.* **2019**, *162*, 376–387. [CrossRef]
4. Chen, Q.; Giambene, G.; Yang, L.; Fan, C.; Chen, X. Analysis of inter-satellite link route s for LEO mega-constellation networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 2743–2755. [CrossRef]
5. Nishiyama, H.; Tada, Y.; Kato, N.; Toyoshima, M. Toward optimized traffic distribution for efficient network capacity utilization in two-layered satellite networks. *IEEE Trans. Veh. Technol.* **2012**, *62*, 1303–1313. [CrossRef]
6. Fu, Y.; Hou, Y.; Wang, Z.; Wu, X.; Gao, K.; Wang, L. Distributed Scheduling Problems in Intelligent Manufacturing Systems. *Tsinghua Sci. Technol.* **2021**, *26*, 625–645. [CrossRef]
7. Peng, Q.; Wu, H.S.; Xue, R.S. Review of Dynamic Task Allocation Methods for UAV Swarms Oriented to Ground Targets. *Complex Syst. Model. Simul.* **2021**, *1*, 163–175. [CrossRef]
8. Pan, Y.; Gao, K.; Li, Z.; Wu, N. Solving Biobjective Distributed Flow-Shop Scheduling Problems with Lot-Streaming Using an Improved Jaya Algorithm. *IEEE Trans. Cybern.* **2022**, 1–11. [CrossRef]
9. Gao, K.Z.; Huang, Y.; Sadollah, A.; Wang, L. A review of energy-efficient scheduling in intelligent production systems. *Complex Intell. Syst.* **2020**, *6*, 237–249. [CrossRef]
10. Du, P.; Nazari, S.; Mena, J.; Fan, R.; Gerla, M.; Gupta, R. Multiroute TCP in SDN-enabled LEO satellite networks. In Proceedings of the MILCOM 2016-2016 IEEE Military Communications Conference, Baltimore, MD, USA, 1–3 November 2016; pp. 354–359.
11. Roth, M.; Brandt, H.; Bischl, H. Implementation of a geographical routing scheme for low Earth orbiting satellite constellations using intersatellite links. *Int. J. Satell. Commun. Netw.* **2021**, *39*, 92–107. [CrossRef]
12. Liu, J.; Zhao, B.; Xin, Q.; Su, J.; Ou, W. DRL-ER: An intelligent energy-aware routing protocol with guaranteed delay bounds in satellite mega-constellations. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 2872–2884. [CrossRef]
13. Wu, S.; Chen, X.; Yang, L.; Fan, C.; Zhao, Y. Dynamic and static controller placement in software-defined satellite networking. *Acta Astronaut.* **2018**, *152*, 49–58. [CrossRef]
14. Ekici, E.; Akyildiz, I.F.; Bender, M.D. A distributed routing algorithm for datagram traffic in LEO satellite networks. *IEEE/ACM Trans. Netw.* **2001**, *9*, 137–147. [CrossRef]
15. Papapetrou, E.; Karapantazis, S.; Pavlidou, F.N. Distributed on-demand routing for LEO satellite systems. *Comput. Netw.* **2007**, *51*, 4356–4376. [CrossRef]
16. Anderson, J.; Cardin, M.A.; Grogan, P. Design and analysis of flexible multi-layer staged deployment for satellite mega-constellations under demand uncertainty. *Acta Astronaut.* **2022**, *198*, 179–193. [CrossRef]
17. Chang, H.S.; Kim, B.W.; Lee, C.G.; Min, S.L.; Choi, Y.; Yang, H.S.; Kim, D.N.; Kim, C.S. FSA-based link assignment and routing in low-earth orbit satellite networks. *IEEE Trans. Veh. Technol.* **1998**, *47*, 1037–1048. [CrossRef]
18. Soret, B.; Smith, D. Autonomous routing for LEO satellite constellations with minimum use of inter-plane links. In Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
19. Li, J.; Lu, H.; Xue, K.; Zhang, Y. Temporal netgrid model-based dynamic routing in large-scale small satellite networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6009–6021. [CrossRef]
20. Jia, M.; Zhu, S.; Wang, L.; Guo, Q.; Wang, H.; Liu, Z. Routing algorithm with virtual topology toward to huge numbers of LEO mobile satellite network based on SDN. *Mob. Netw. Appl.* **2018**, *23*, 285–300. [CrossRef]
21. Liu, P.; Chen, H.; Wei, S.; Li, L.; Zhu, Z. Hybrid-traffic-detour based load balancing for onboard routing in LEO satellite networks. *China Commun.* **2018**, *15*, 28–41. [CrossRef]
22. Taleb, T.; Mashimo, D.; Jamalipour, A.; Kato, N.; Nemoto, Y. Explicit load balancing technique for NGEO satellite IP networks with on-board processing capabilities. *IEEE/ACM Trans. Netw.* **2008**, *17*, 281–293. [CrossRef]
23. Wang, L.; Pan, Z.X.; Wang, J.J. A Review of Reinforcement Learning Based Intelligent Optimization for Manufacturing Scheduling. *Complex Syst. Model. Simul.* **2021**, *1*, 257–270. [CrossRef]
24. Desaulniers, G.; Desrosiers, J.; Erdmann, A.; Solomon, M.M.; Soumis, F. VRP with Pickup and Delivery. *Veh. Routing Probl.* **2002**, *9*, 225–242.

25. Toth, P.; Vigo, D. (Eds.) *The Vehicle Routing Problem*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2002.
26. Cordeau, J.F.; Laporte, G.; Potvin, J.Y.; Savelsbergh, M.W. Transportation on demand. *Handb. Oper. Res. Manag. Sci.* **2007**, *14*, 429–466.
27. Psaraftis, H.N.; Orlin, J.B.; Bienstock, D.; Thompson, P.M. *Analysis and Solution Algorithms of Sealift Routing and Scheduling Problems*; Massachusetts Institute of Technology, Alfred P. Sloan School of Management: Cambridge, MA, USA, 1985.
28. Fisher, M.L.; Rosenwein, M.B. An interactive optimization system for bulk-cargo ship scheduling. *Nav. Res. Logist. (NRL)* **1989**, *36*, 27–42. [CrossRef]
29. Solomon, M.; Chalifour, A.; Desrosiers, J.; Boisvert, J. An application of vehicle-routing methodology to large-scale larvicide control programs. *Interfaces* **1992**, *22*, 88–99. [CrossRef]
30. Wang, X.; Regan, A.C. Local truckload pickup and delivery with hard time window constraints. *Transp. Res. Part B Methodol.* **2002**, *36*, 97–112. [CrossRef]
31. Wang, L.; Lu, J. A memetic algorithm with competition for the capacitated green vehicle routing problem. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 516–526. [CrossRef]
32. Shang, P.; Yang, L.; Zeng, Z. Solving school bus routing problem with mixed-load allowance for multiple schools. *Comput. Ind. Eng.* **2021**, *151*, 106916. [CrossRef]
33. Rios, B.H.O.; Xavier, E.C.; Miyazawa, F.K. Recent dynamic vehicle routing problems: A survey. *Comput. Ind. Eng.* **2021**, *160*, 107604. [CrossRef]
34. Wang, X.; Dessouky, M.; Ordonez, F. A pickup and delivery problem for ridesharing considering congestion. *Transp. Lett.* **2016**, *8*, 259–269. [CrossRef]
35. Psaraftis, H.N. An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. *Transp. Sci.* **1983**, *17*, 351–357. [CrossRef]
36. Sexton, T.R.; Bodin, L.D. Optimizing single vehicle many-to-many operations with desired delivery times: I. Scheduling. *Transp. Sci.* **1985**, *19*, 378–410. [CrossRef]
37. Savelsbergh, M.; Sol, M. Drive: Dynamic routing of independent vehicles. *Oper. Res.* **1998**, *46*, 474–490. [CrossRef]
38. Lu, Q.; Dessouky, M. An exact algorithm for the multiple vehicle pickup and delivery problem. *Transp. Sci.* **2004**, *38*, 503–514. [CrossRef]
39. Ropke, S.; Cordeau, J.F. Branch and cut and price for the pickup and delivery problem with time windows. *Transp. Sci.* **2009**, *43*, 267–286. [CrossRef]
40. Kok, A.L.; Hans, E.W.; Schutten, J.M. Vehicle routing under time-dependent travel times: The impact of congestion avoidance. *Comput. Oper. Res.* **2012**, *39*, 910–918. [CrossRef]
41. Gromicho, J.; van Hoorn, J.J.; Kok, A.L.; Schutten, J.M. Restricted dynamic programming: A flexible framework for solving realistic VRPs. *Comput. Oper. Res.* **2012**, *39*, 902–909. [CrossRef]
42. Mahmoudi, M.; Zhou, X. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations. *Transp. Res. Part B Methodol.* **2016**, *89*, 19–42. [CrossRef]
43. Boland, N.; Hewitt, M.; Marshall, L.; Savelsbergh, M. The continuous-time service network design problem. *Oper. Res.* **2017**, *65*, 1303–1321. [CrossRef]
44. Tong, L.C.; Zhou, L.; Liu, J.; Zhou, X. Customized bus service design for jointly optimizing passenger-to-vehicle assignment and vehicle routing. *Transp. Res. Part C: Emerg. Technol.* **2017**, *85*, 451–475. [CrossRef]
45. Liu, J.; Xing, L.; Wang, L.; Du, Y.; Yan, J.; Chen, Y. A Data-driven Parallel Adaptive Large Neighborhood Search Algorithm for a Large-scale Inter-satellite Link Scheduling Problem. *Swarm Evol. Comput.* **2022**, *74*, 101124. [CrossRef]