remote sensing

MDPI

Article

Distributed Coordination of Space–Ground Multiresources for Remote Sensing Missions

Runzi Liu [1], Xu Ding [1], Weihua Wu [2] and Wei Guo [3,*]

[1] School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China
[2] School of Physics and Information Technology, Shaanxi Normal University, Xi'an 710119, China
[3] School of Telecommunications Engineering, Xidian University, Xi'an 710071, China
* Correspondence: w.guo@xidian.edu.cn

Abstract: As data relay satellites (DRSs) play an increasingly important supporting role in remote sensing missions, efficient coordination across space–ground multiresources becomes a significant problem. Owing to the implementation problem of the centralized coordinate methods, this paper studies a distributed coordinate resource scheduling method which is realizable in the current space network structure. To be specific, we first formulate the multiple resource coordination problem into an MILP problem based on a modified time-expanded graph. Then, the problem is transferred and decomposed into subproblems for remote sensing satellite (RSS) systems and DRS systems to solve distributedly. Afterwards, we propose a distributed iterative scheme for the RSS systems and DRS systems based on alternating direction method of multipliers (ADMM), in which only the schedule information of the inter-satellite links are required to exchange between RSS systems and DRS systems. Simulation results are provided to validate the effectiveness of our distributed coordinated resource scheduling algorithm.

Keywords: remote sensing; space–ground resource coordination; resource scheduling

Citation: Liu, R.; Ding, X.; Wu, W.; Guo, W. Distributed Coordination of Space–Ground Multiresources for Remote Sensing Missions. *Remote Sens.* **2023**, *15*, 3362. https://doi.org/10.3390/rs15133362

Academic Editors: Michele Nappi, Chen Chen, Stefano Berretti, Shaohua Wan and Ying Ju

Received: 6 May 2023
Revised: 18 June 2023
Accepted: 20 June 2023
Published: 30 June 2023

1. Introduction

Benefiting from the inherent wide coverage and potential overflight capability, remote sensing satellites (RSSs) are widely employed to observe specific targets for diverse missions, e.g., mapping, environmental monitoring, disaster surveillance, and national defense [1,2]. With the rapid growth of the information acquiring capability of the sensors on RSSs, how to download the huge amount of onboard data to the ground in time has become a great challenge for remote sensing missions. The length of time windows between RSSs and ground stations are usually very short (i.e., 5 min–10 min). To tackle this problem, data relay satellites (DRSs) have been deployed in geosynchronous Earth orbit to provide high-bandwidth, near continuous communication support to low-orbiting RSSs in recent years [3]. To coordinate the space and ground communication resources, the RSS systems and DRS systems are integrated to be space–ground coordination networks [4].

However, the resource scheduling problem of RSS systems [5–8] and the resource scheduling problem of DRS systems [9–11] have been studied independently for a long time. With these independent methods, the resources of RSS and DRS systems are scheduled separately. This may lead to resource underutilization due to the absence of synergy in RSS and DRS systems. To cater for this issue, some studies have been carried out on the integrated resource scheduling of RSS and DRS systems. For instance, in [12], the joint observation and transmission scheduling problem is modeled as a flow problem on a time-expanded graph, and a heuristic resource scheduling algorithm is proposed. In [13], a planning strategy based on resource interchange scheme for remote sensing missions of space information networks consisting of DRSs and RSSs is proposed. Zhu et al. design an integrated observation mission scheduling method based on genetic annealing algorithm which realized the cooperation of the observation and data transmission phase [14].

Although the work in [12–14] improves the resource utilization of the whole networks through jointly scheduling the resources of RSSs and DRSs in a centralized manner, the implementation encounters immense obstacles in practice. Specially, these works assume all the resources in the space network are integrated and managed centrally. However, the resources belong to different RSS/DRS systems and are currently managed by their operation control centers. Many RSS and DRS systems are managed by different government departments or armies, each with their own interests [15], which cannot provide their local mission/resource information to or be managed by other departments. Therefore, it is urgent to develop distributed coordinate resource scheduling strategies which do not need global information and uniform management.

The technical challenges of designing efficient distributed coordinated resource scheduling mechanism are threefold: (1) How to design a distributed coordinated framework among the RSS and DRS systems based on the current network architecture? (2) How to reduce the amount of local mission/resource information required to be exchanged among the satellite systems? (3) How to realize the coordination not only between RSS system and DRS system but also among RSS systems?

To address the above technical challenges, this paper studies a distributed resource scheduling method to realize the resource coordination between RSS systems and DRS systems. The contributions are outlined as follows:

- By modifying the time-expanded graph, we formulate a joint optimization framework of multiple resource scheduling of the RSS systems and DRS systems and decompose it in terms of each satellite system.
- Based on the alternating direction method of multipliers (ADMM), a distributed coordinated space–ground multiresource scheduling method is developed for remote sensing missions. Compared with the centralized counterparts, it not only does not require the introduction of network entities into the current network but also avoids any information exchange outside the schedule information of the intersatellite link between RSSs and DRSs. Therefore, the proposed method is much more practical than the centralized methods.
- Simulation results shows that the performance of the proposed method is very close to that of the centralized method and is much better than the noncoordination method.

The rest of the paper is organized as follows. Section 2 presents the network model as well as the optimization formulation of the distributed coordinated multiresource scheduling problem of the integrated network. Section 3 transfers and decomposes the optimization problem into subproblems to be solved by the RSS systems and DRS systems distributedly. In Section 4, the ADMM-based distributed coordinated multiresource scheduling algorithm is developed for the remote sensing missions in integrated networks. Simulation results and conclusion are given in Sections 5 and 6, respectively.

2. System Model

This section introduces the system model employed in the distributed coordinated scheduling of space–ground multiresources for remote sensing missions. Firstly, the space–ground coordination network scenario and the problem of multiresources coordinated scheduling are introduced. Subsequently, we modified the time-expanded graph model to capture the dynamics of the network. Finally, the multiresources resource coordinated scheduling problem is formulated into an MILP problem based on the modified time-expanded graph.

2.1. Scenario and Problem Description

Consider a space–ground coordination network (SGCN) for remote sensing missions which consists of a DRS system and N RSS systems, as depicted in Figure 1. The RSSs observe the observation objects for the users of their system, and the acquired data are sent back to the ground either by the ground stations belonging to the same RSS system or through the DRS system. The operation control center (OCC) of each RSS system and the operation control center of DRS system are connected through the terrestrial network to exchange data relay requirements and coordinated scheduling information.

The set of DRSs in the space network is denoted by $R = \{r_1, r_2, ..., r_i, ...\}$, where r_i denotes the i-th DRS. The set of RSS systems is represented as $\mathcal{N} = \{1, 2, ..., n, ..., N\}$, where symbol n denotes the n-th RSS system. The RSS system n consists of an operation control center, several RSSs, and ground stations. The set of satellites in the n-th RSS system is denoted by $S_n = \{s_{n,1}, s_{n,2}, ..., s_{n,i}, ...\}$, where $s_{n,i}$ denotes the i-th RSS of n. Similarly, $G_n = \{g_{n,1}, g_{n,2}, ..., g_{n,i}, ...\}$ denotes the set of ground stations in the n-th RSS system, where $g_{n,i}$ represents the i-th ground station of n. The payload of each RSS includes an imager, a solid-state memory, and two transceivers which are used to communicate with ground stations and the DRSs, respectively.

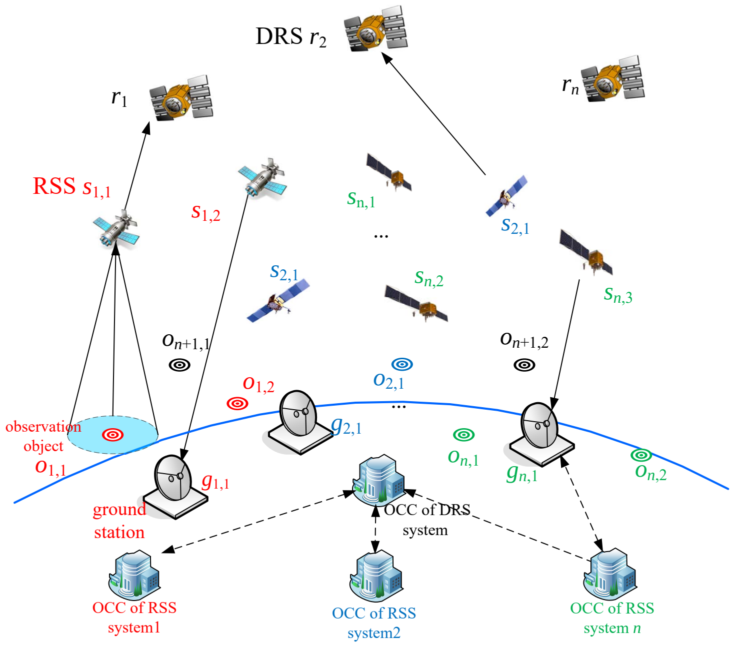

Figure 1. Space–ground coordination network.

The satellite systems adopt the "preapplication and offline scheduling" paradigm [16,17]. That is, the remote sensing mission requests are submitted to the operation control centers of RSS systems in advance, and then the operation control centers of RSS and DRS systems schedule space–ground resources coordinately for the scheduling horizon offline. Let T denote the scheduling horizon. There are a number of remote sensing missions requests for each RSS system to complete during T. The mission set of the n-th RSS system in the scheduling horizon is denoted by $OM_n = \{om_{n,1}, om_{n,2}, \ldots, om_{n,i}, \ldots\}$. Each mission is described by a 4-tuple. Take the i-th mission of RSS system n as an example. $om_{n,i} = [o_{n,i}, b_{n,i}, st_{n,i}, et_{n,i}]$, where $o_{n,i}$ denote the observation object of mission $om_{n,i}$, and $b_{n,i}$ is the amount of data that needs to be acquired and transmitted back to the ground. $[st_{n,i}, et_{n,i}]$ is the feasible scheduling window of mission $om_{n,i}$, i.e., $st_{n,i}$ and $et_{n,i}$, respectively, denote the earliest start time and latest end time of $om_{n,i}$.

In the mission scheduling stage, the operation control centers of RSS system cooperatively schedule their resources in parallel by interacting with the operation control center of DRS systems for relay requirements and coordinated scheduling information. The outputted resource schedules assign the RSS, DRS (or ground station), and the observation and data transmission window for each mission. The aim of the distributed space–ground resource coordinated scheduling strategy is to maximize the number of successfully scheduled missions of the entire SGCN.

2.2. Network Model

In order to represent the impact of dynamic topology on execution process of remote sensing missions effectively, we modified the time-expanded graph (TEG) [18,19] by adding virtual vertices and virtual arcs, which simplifies the model of the mission execution process. The scheduling horizon is divided into K time slots of equal length. It is assumed that the network topology is static in each time slot and only changes at the time of time slot switching [19,20]. As shown in Figure 2, the time-expanded graph $G_K(V, A)$ is a K-layered directed graph, where each layer corresponds to a time slot in the network. V and A represent the set of vertices and arcs of the graph, respectively. There are two classes of vertices, namely ordinary vertices and virtual vertices. The vertex set is denoted by $V = V_{OD} \cup V_{VT}$, where V_{OD} and V_{VT} represent the set of ordinary vertices and virtual vertices, respectively. Similarly, the arc set A contains two types of arcs, namely ordinary arcs and virtual arcs. The arc set is denoted by $A = A_{OD} \cup A_{VT}$. Hereinafter, we introduce the ordinary vertices, ordinary arcs, virtual vertices, and virtual arcs in TEG.

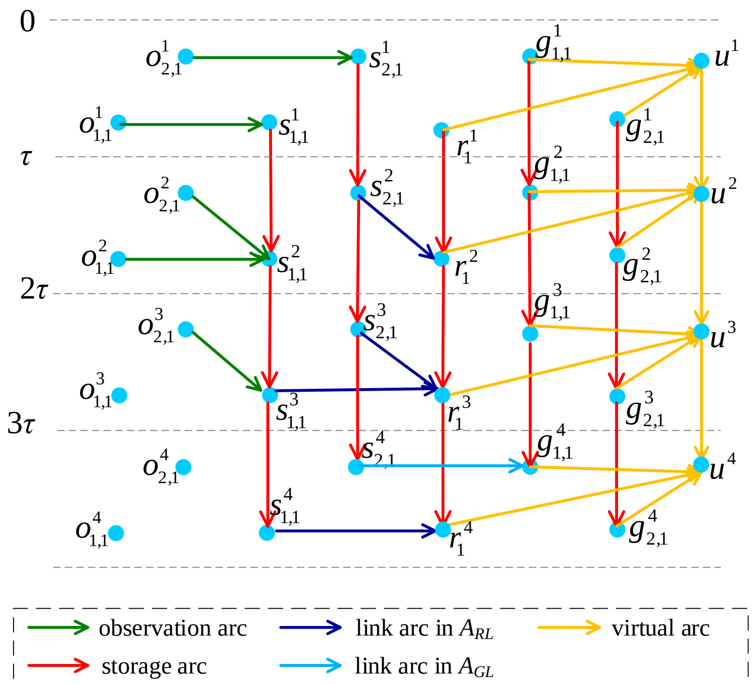

Figure 2. Time-expanded graph.

2.2.1. Ordinary Vertices

The ordinary vertices of TEG represent the temporal replicas of the observation objects, RSSs, DRSs, and ground stations in each slot of SGCN, of which the set is denoted by $V_{OD} = V_o \cup V_s \cup V_r \cup V_g$, where V_o, V_s, V_r, and V_g, respectively, represent the set of vertices representing the temporal replicas of the observation objects, RSSs, DRSs, and ground stations in each time slot [18,19]. For example, $V_o = \{o_{n,i}^k | 1 \leq k \leq K, 1 \leq i \leq |OM_n|, 1 \leq n \leq N\}$ denotes the set of observation object vertices in TEG, where $o_{n,i}^k$ is the temporal replica of the observation object $o_{n,i}$ in the k-th time slot. Similarly, $s_{n,i}^k$, r_i^k, and $g_{n,i}^k$ represent the temporal replica of RSS $s_{n,i}$, DRS r_i, and ground station $g_{n,i}$ in the k-th slot, respectively.

2.2.2. Ordinary Arcs

The ordinary arcs in TEG are divided into three categories, namely, link arcs, storage arcs, and observation arcs. The set of ordinary arcs is denoted by $A_{OD} = A_L \cup A_S \cup A_O$, where A_L, A_S, and A_O are the set of link arcs, storage arcs, and observation arcs, respectively. The link arcs represent the transmission ability of each time slot in the network. The link arc set is denoted by $A_L = A_{RL} \cup A_{GL}$, where A_{RL} represents the set of link arcs corresponding to the links between RSSs and DRSs, and A_{GL} denotes the set of link arcs corresponding to the links between the RSSs and ground stations of the same RSS system. We use $sr^k_{n,i,j}$ to denote the link arc from vertex $s^k_{n,i}$ to vertex r^k_j in TEG, which represents the link from RSS $s_{n,i}$ to DRS r_j in the k-th slot. Similarly, $sg^k_{n,i,j}$ denote the link arc from $s^k_{n,i}$ to $g^k_{n,j}$. Then, the link arc set A_{GL} and A_{RL} can be expressed as

$$\begin{aligned} A_{RL} &= \left\{ sr^k_{n,i,j} | lc^k(r_j) \in R^k_C(s_{n,i}), 1 \le k \le K, 1 \le i \le |S_n|, 1 \le j \le |R|, 1 \le n \le N \right\}, \\ A_{GL} &= \left\{ sg^k_{n,i,j} | lc(g_{n,j}) \in R^k_C(s_{n,i}), 1 \le k \le K, 1 \le i \le |S_n|, 1 \le j \le |G_n|, 1 \le n \le N \right\}, \end{aligned} \tag{1}$$

where $lc^k(r_j)$ represents the position of r_j in the k-th slot, and $R^k_C(s_{n,i})$ denotes the communication range of $s_{n,i}$ in the k-th slot. Similarly, $lc(g_{n,j})$ is the position of $g_{n,j}$.

The storage arcs represent the storage ability of the nodes in SGCN. Let $ss^k_{n,i}$ denote the storage arc from vertex $s^k_{n,i}$ to vertex $s^{k+1}_{n,i}$ in TEG, which represents the storage capability of RSS $s_{n,i}$ in the k-th slot. Similarly, rr^k_i denotes the storage arc from r^k_i to r^{k+1}_i, and $gg^k_{n,i}$ denotes the storage arc from $g^k_{n,i}$ to $g^{k+1}_{n,i}$. The set of storage arcs A_S is expressed as

$$\begin{aligned} A_S = &\left\{ ss^k_{n,i} | 1 \le k \le K-1, 1 \le i \le |S_n|, 1 \le n \le N \right\} \\ &\cup \left\{ rr^k_i | 1 \le k \le K-1, 1 \le i \le |R| \right\} \\ &\cup \left\{ gg^k_{n,i} | 1 \le k \le K-1, 1 \le i \le |G_n|, 1 \le n \le N \right\}. \end{aligned} \tag{2}$$

The observation arcs model the opportunities for RSSs to observe the observation objects. We use $os^k_{n,i,j}$ to denote the observation arc from vertex $o^k_{n,i}$ to vertex $s^k_{n,j}$ in TEG, which represents the opportunity for RSS $s_{n,j}$ to observe the observation object $o_{n,i}$ in the k-th slot. The set of observation arcs A_O is expressed as

$$A_O = \left\{ os^k_{n,i,j} \middle| lc^k(o_{n,i}) \in R^k_O(s_{n,j}), 1 \le k \le K, 1 \le i \le |OM_n|, 1 \le j \le |S_n|, 1 \le n \le N \right\}, \tag{3}$$

where $lc^k(o_{n,i})$ represents the location of observation object $o_{n,i}$, and $R^k_O(s_{n,j})$ denotes the visible range of $s_{n,j}$ in the k-th slot. The capacity of a link/storage/observation arc represents the maximum amount of data that the corresponding communication/storage/observation resource can transmit/storage/observe in a slot, which is denoted by $C(a)$, $\forall a \in A$. For more details of arc capacity, please refer to reference [18,19].

2.2.3. Virtual Vertices and Virtual Arcs

With the ordinary vertices and ordinary arcs in TEG, the execution process of missions in SGCN can be represented as a flow originated from an observation object vertex and destinated to a ground station or DRS vertex (DRSs are stationary on the ground and generally adopt a transparent forwarding mode [21,22]. That is, DRSs can forward the mission data to the ground immediately after receiving them with no need for considering a visible relationship with the ground station. Therefore, to simplify the coordinated resource scheduling model, the process of data transmission from DRS to the ground is ignored in this paper) [18]. Therefore, the coordinated resource scheduling problem can be modeled as a flow optimization problem in TEG. However, the number of feasible flows for each mission is proportionate to the sum of the number of ground stations and DRSs, which raises the complexity of the flow problem. Inspired by the fact that the mission requests do not specify which ground station or DRS to receive or relay the mission data to the ground, we simplify the mission model by adding virtual vertices as the end point of the mission flow in the traditional TEG, which directly connect to the ground station and DRS vertices through virtual arcs, as shown in Figure 2. With the introduction of virtual vertices and virtual arcs in TEG, the number of feasible mission flows corresponding to a given mission request can be evidently decreased.

Virtual vertices represent virtual sinks of mission data, of which the set is denoted by $V_{VT} = \{u^k | 1 \leq k \leq K\}$, where u^k represents the virtual sink of the k-th slot. The virtual arcs represent the virtual links between the DRSs or ground stations and the virtual sink. Let ru_i^k denote the virtual arc from vetex r_i^k to vertex u^k in TEG. Similarly, $gu_{n,i}^k$ represents the virtual arc from $g_{n,i}^k$ to u^k, and uu^k represents the virtual arc from u^k to u^{k+1}. The set of virtual arcs A_{VT} is expressed as

$$\begin{aligned} A_{VT} = &\left\{ru_i^k | 1 \leq i \leq |R|, 1 \leq k \leq K\right\} \\ &\cup \left\{gu_{n,i}^k | 1 \leq i \leq |G_n|, 1 \leq n \leq N, 1 \leq k \leq K\right\} \cup \left\{uu^k | 1 \leq k \leq K-1\right\}. \end{aligned} \tag{4}$$

Because the virtual vertex represents the virtual sink of mission data received by the DRSs and ground stations, it is assumed that the capacity of the virtual arcs is infinite.

2.3. Problem Formulation

With the addition of virtual vertices and virtual arcs in TEG, the mission execution process in SGCN can be modeled as the flow originated from an observation object vertex and the destinated virtual vertex in TEG. Specifically, the implementing process of mission $om_{n,i}$ is modeled by the flow set $F_{n,i}$ in TEG, which is expressed as

$$F_{n,i} = \{o_{n,i}^{sk} \to u^{ek} \Big| \left\lceil \frac{st}{\tau} \right\rceil \leq sk \leq \left\lceil \frac{et}{\tau} \right\rceil, ek = \left\lceil \frac{et}{\tau} \right\rceil \}. \tag{5}$$

The set of flows corresponding to all the missions in the network is given by

$$\mathcal{F} = \bigcup_{1 \leq n \leq N} F_n = \bigcup_{1 \leq n \leq N} \bigcup_{1 \leq i \leq |OM_n|} F_{n,i}, \tag{6}$$

where F_n is the flow set corresponding to the missions of the n-th RSS system.

By modeling the missions as the flows of TEG, the coordinated resource scheduling problem can be transformed into the multicommodity flow problem. Let the Boolean variable $z_{n,i} \in 0, 1$ represent whether the mission $om_{n,i}$ is successfully scheduled. As coordinated resource scheduling aims at maximizing the number of successfully scheduled missions, the target of the corresponding multicommodity flow problem is expressed follows:

$$\max \sum_{1 \leq n \leq N} \sum_{1 \leq i \leq |OM_n|} z_{n,i} \tag{7}$$

Let $x(f)$ denote the value of flow $f \in \mathcal{F}$ on TEG. Let $x(a, f) > 0$ denote the amount of flow f on arc $a \in A$. The flows in set $\mathcal{F}$ should satisfy the following constraints:

(1) Data Volume Constraint

The data volume constraint guarantees that for a successfully scheduled mission $om_{n,i}$, the total amount of data acquired and transmitted by RSSs within the scheduling horizon satisfies mission demands $b_{n,i}$, which is given by

$$\sum_{f \in F_{n,i}} x(f) = z_{n,i} \cdot b_{n,i}, \forall 1 \leq i \leq |OM_n|, 1 \leq n \leq N \tag{8}$$

Equation (8) imposes that if the mission $om_{n,i}$ is successfully executed, the sum volume of the corresponding flow in TEG is $b_{n,i}$, and 0 otherwise.

(2) Flow Conservation Constraint

For each flow $f \in \mathcal{F}$ of TEG, the value entering any vertex except its source and sink equals the value out of this vertex. That is,

$$\sum_{h(a)=v} x(a,f) - \sum_{t(a)=v} x(a,f) = \begin{cases} -x(f) & v = s(f), f \in \mathcal{F} \\ 0 & v \in V - \{s(f), d(f)\}, f \in \mathcal{F} \\ x(f) & v = d(f), f \in \mathcal{F} \end{cases} \tag{9}$$

where $h(a)$ and $t(a)$ denote the head and tail of arc a in TEG, respectively. $s(f)$ and $d(f)$, respectively, represent the source and destination vertex of the flow f in TEG.

(3) Conflict Constraints

Due to the limited service capability of antenna/imager and the restriction of satellite platform attitude, there are conflicts between the schedule of the same observation and communication resources in SGCN. For example, each RSS can only observe one target in each time slot. To model this kind of conflict, we introduce a set of Boolean variables $y(os^k_{n,i,j}) \in \{0,1\}$ to indicate whether the observation resources of $s_{n,j}$ are scheduled to observe $o_{n,i}$ in the k-th time slot. Specifically, $y(o^k_{n,j}, s^k_{n,i}) = 1$ means that the observation resources of $s_{n,j}$ are scheduled to observe $o_{n,i}$ in the k-th time slot, and 0 otherwise. The conflicts of observation resources can be expressed as

$$\sum_{i:os^k_{n,i,j} \in A_O} y(os^k_{n,i,j}) \leq 1, \quad \forall 1 \leq k \leq K, 1 \leq j \leq |S_n|, 1 \leq n \leq N. \tag{10}$$

To avoid the waste of observation resources, it is restricted that each observation object can only be observed once during the scheduling horizon, which can be formulated as

$$\sum_{j,k:os^k_{n,i,j} \in A_O} y(os^k_{n,i,j}) \leq 1, \quad \forall 1 \leq i \leq |OM_n|, 1 \leq n \leq N. \tag{11}$$

Each RSS has two transceivers to communicate with the ground stations and the DRSs, respectively. Because of using single-access antenna, an RSS can establish a communication link with at most one DRS in a time slot. Let Boolean variables $y(sr^k_{n,i,j}) \in \{0,1\}$ denote whether link $(s_{n,i}, r_j)$ are scheduled in the k-th time slot. Then, the conflict of communication resources of RSSs can be expressed as

$$\sum_{j:sr^k_{n,i,j} \in A_{RL}} y(sr^k_{n,i,j}) \leq 1, \quad \forall 1 \leq k \leq K, 1 \leq i \leq |S_n|, 1 \leq n \leq N. \tag{12}$$

Likewise, an RSS can only communicate with at most one ground station in a time slot, even if there are multiple ground stations in its coverage range, which is formulated as

$$\sum_{j:sg^k_{n,i,j} \in A_{GL}} y(sg^k_{n,i,j}) \leq 1, \quad \forall 1 \leq k \leq K, 1 \leq i \leq |S_n|, 1 \leq n \leq N, \tag{13}$$

where the Boolean variable $y(sg^k_{n,i,j}) \in \{0,1\}$ denotes whether link $(s_{n,i}, g_{n,j})$ is scheduled in the k-th time slot. Similarly, a DRS/ground station can communicate with only one RSS in the same time slot, which can be expressed as

$$\sum_{n,i:sr^k_{n,i,j} \in A_{RL}} y(sr^k_{n,i,j}) \leq 1, \quad \forall 1 \leq k \leq K, 1 \leq j \leq |R|, \tag{14}$$

$$\sum_{i:sg^k_{n,i,j}\in A_{GL}} y(sg^k_{n,i,j}) \le 1, \quad \forall 1 \le k \le K, 1 \le j \le |G_n|, 1 \le n \le N. \tag{15}$$

(4) Capacity Constraints

The capacity constraint models the constraints of the capacity of observation, communication, and storage resources on the mission execution process by restricting the flow volume on the arcs in TEG, which are given by

$$\sum_{f\in F_n} x(ss^k_{n,i}, f) \le C(ss^k_{n,i}), \quad \forall ss^k_{n,i} \in A_S, \tag{16}$$

$$\sum_{f\in F_{n,i}} x(os^k_{n,i,j}, f) \le C(os^k_{n,i,j}) \cdot y(os^k_{n,i,j}), \quad \forall os^k_{n,i,j} \in A_O, \tag{17}$$

$$\sum_{f\in F_n} x(sg^k_{n,i,j}, f) \le C(sg^k_{n,i,j}) \cdot y(sg^k_{n,i,j}), \quad \forall sg^k_{n,i,j} \in A_{GL}, \tag{18}$$

$$\sum_{f\in F_n} x(sr^k_{n,i,j}, f) \le C(sr^k_{n,i,j}) \cdot y(sr^k_{n,i,j}), \quad \forall sr^k_{n,i,j} \in A_{RL}. \tag{19}$$

To be specific, the capacity constraint of the storage arc in Equation (16) imposes that the amount of data cached in RSSs in any time slot does not exceed its storage volume. The capacity constraint of the observation arc in Equation (17) shows that if the observation resource $os^k_{n,i,j}$ is scheduled, the total volume of the flows on arc $os^k_{n,i,j}$ does not exceed its capacity; otherwise, the flow volume of all flows on arc $os^k_{n,i,j}$ is zero. This constraint ensures that the observation resource of RSS observes the target only in the slots being scheduled, and the amount of data acquired in each slot does not exceed the capacity of the corresponding observation resource. Similarly, Equations (18) and (19) enforce that the links between RSS to ground stations/DRSs transmit only in the slots being scheduled, and the amount of data transmitted in each slot does not exceed the capacity of the corresponding link.

In summary, the coordinated resource scheduling problem can be formulated as the following optimization problem:

$$\begin{aligned} \text{P1}: \quad & \max \sum_{1\le n\le N} \sum_{1\le i\le |OM_n|} z_{n,i} \\ \text{s.t.} \quad & \text{Equations } (4) \sim (15) \end{aligned}$$

It can be observed from P1 that the objective and all the constraints satisfy linear conditions. $x(f)$ and $x(a, f)$ are continuous variables, and $z_{n,i}, y(os^k_{n,i,j}), y(sg^k_{n,i,j})$, and $y(sr^k_{n,i,j})$ are integer variables. Therefore, problem P1 is a mixed-integer linear programming problem [23]. Moreover, P1 is a centralized optimization model; directly solving P1 not only requires global mission and resource information but also suffers from high complexity. In order to distributedly solve the resource scheduling in polynomial time under the coordination of the RSS and DRS operation control center, P1 is transformed and decomposed in the next section.

3. Problem Transformation and Decomposition

This section presents a distributed solving approach for coordinated space–ground multiresource scheduling for remote sensing missions. Firstly, the mixed-integer linear programming problem P1 is converted into a linear programming problem through variable relaxation. Then, the relaxed problem is decomposed into independent subproblems by duplicating the global variables and establishing local variables for each system.

3.1. Problem Transformation

The mixed-integer linear programming problem P1 is generally NP-hard [24] and of large size. In order to solve P1 in polynomial time, we relax its integer variables, i.e.,

$$\begin{aligned} &0 \le z_{n,i} \le 1, \quad \forall 1 \le i \le |OM_n|, 1 \le n \le N, \\ &0 \le y(a) \le 1, \quad \forall a \in A_O \cup A_L \end{aligned} \tag{20}$$

Thus, the original problem P1 is transformed into

$$\begin{aligned} \text{P2}: \quad \max \sum_{1 \le n \le N} \sum_{1 \le i \le |OM_n|} z_{n,i} \\ \text{s.t.} \quad \text{Equations } (4) \sim (16) \end{aligned}$$

It can be seen that P2 is a linear programming problem. To enable N RSS systems and the DRS system to complete missions in coordination, P2 is decomposed into $N+1$ subproblems that can be solved in parallel in the next subsection.

3.2. Problem Decomposition

The objective of P2 can be directly separated with respect to N RSS systems. However, due to the constraints of Equations (12), (14) and (19), the variables $y(sr^k_{n,i,j})(\forall sr^k_{n,i,j} \in A_{RL})$ become global variables between the DRS system and RSS systems and thus cannot be classified as local variables of any system. To handle this problem, we introduce the local copies of global variable $y(sr^k_{n,i,j})$ as [25,26], denoted by $\hat{y}_r(sr^k_{n,i,j})$ and $\hat{y}_n(sr^k_{n,i,j})$, which belongs to the DRS system and RSS system n, respectively. By substituting the local copies with the global variables into P2, an equivalent form can be obtained

$$\begin{aligned} \text{P3}: \quad & \max \sum_{1 \le n \le N} \sum_{1 \le i \le |OM_n|} z_{n,i} \\ \text{s.t.} \quad & \text{Equations } (4) \sim (7), (9), (11) \sim (14) \text{ and } (16) \end{aligned}$$

$$\sum_{j:sr^k_{n,i,j} \in A_{RL}} \hat{y}_n(sr^k_{n,i,j}) \le 1, \quad \forall 1 \le k \le K, 1 \le i \le |S_n|, 1 \le n \le N, \tag{21}$$

$$\sum_{n,i:sr^k_{n,i,j} \in A_{RL}} \hat{y}_r(sr^k_{n,i,j}) \le 1, \quad \forall 1 \le k \le K, 1 \le j \le |R|, \tag{22}$$

$$\sum_{f \in F_n} x(sr^k_{n,i,j}, f) \le C(sr^k_{n,i,j}) \cdot \hat{y}_n(sr^k_{n,i,j}), \quad \forall sr^k_{n,i,j} \in A_{RL} \tag{23}$$

$$\hat{y}_r(sr^k_{n,i,j}) = \hat{y}_n(sr^k_{n,i,j}) = y(sr^k_{n,i,j}), \quad \forall sr^k_{n,i,j} \in A_{RL} \tag{24}$$

The consensus constraint Equation (24) imposes that the local copy variables $\hat{y}_r(sr^k_{n,i,j})$, $\hat{y}_n(sr^k_{n,i,j})$ be consistent with the corresponding global variable $y(sr^k_{n,i,j})$. All variables in problem P3 except the global variables $\boldsymbol{y}^C = \{y(sr^k_{n,i,j})\}_{sr^k_{n,i,j} \in A_{RL}}$ can be divided into individual satellite systems as local variables. Thus, the constraints of P3, except the consistency constraint, can be separated into each RSS system and the DRS system. Note that since DRS systems and RSS systems play different roles in remote sensing missions, their local variables and constraints must be discussed separately. Specifically, the local variables of the RSS system n are

$$\begin{cases} \boldsymbol{z}_n = \{z_{n,i}\}_{1 \le i \le |OM_n|} \\ \boldsymbol{x}_n = \left(\{x(f)\}_{f \in F_n}, \{x(a,f)\}_{a \in A, f \in F_n}\right) \\ \boldsymbol{y}^L_n = \left(\{y(os^k_{n,i,j})\}_{os^k_{n,i,j} \in A^n_O}, \{y(sg^k_{n,i,j})\}_{sg^k_{n,i,j} \in A^n_{GL}}\right) \\ \hat{\boldsymbol{y}}^C_n = \{\hat{y}_n(sr^k_{n,i,j})\}_{sr^k_{n,i,j} \in A^n_{RL}} \end{cases}, \tag{25}$$

where A^n_O, A^n_{GL}, and A^n_{RL} represent the set of observation arcs, ground–satellite link arcs, and intersatellite link arcs belong to the RSS system n, respectively. For example, A^n_O is defined as

$$A^n_O = \left\{ os^k_{n,i,j} \middle| lc^k(o_{n,i}) \in R^k_O(s_{n,j}), 1 \le k \le K, 1 \le i \le |OM_n|, 1 \le j \le |S_n| \right\}. \tag{26}$$

The local variable of the DRS system is $\hat{\boldsymbol{y}}^C_r = \{\hat{y}_r(sr^k_{n,i,j})\}_{sr^k_{n,i,j} \in A_{RL}}$.

Based on the local constraints, the set of feasible solutions for the local variables of RSS system n is given by

$$\xi_n = \left\{ \boldsymbol{z}_n, \boldsymbol{x}_n, \boldsymbol{y}^L_n, \hat{\boldsymbol{y}}^C_n | \mathrm{C1}, \mathrm{C2}, \mathrm{C3}, \mathrm{C4}, \mathrm{C5}, \mathrm{C6}, \mathrm{C7}, \mathrm{C8}, \mathrm{C9}, \mathrm{C10}, \mathrm{C11} \right\} \tag{27}$$

wherein,

$$\begin{aligned}
&\mathrm{C1}: x(f) = z_{n,i} \cdot b_{n,i}, \quad \forall 1 \le i \le |OM_n| \\
&\mathrm{C2}: \sum_{h(a)=v} x(a,f) - \sum_{t(a)=v} x(a,f) = \begin{cases} -x(f) & v = s(f), f \in F_n \\ 0 & v \in V - \{s(f), d(f)\}, f \in F_n \\ x(f) & v = d(f), f \in F_n \end{cases} \\
&\mathrm{C3}: \sum_{f \in F_n} x(ss^k_{n,i}, f) \le C(ss^k_{n,i}), \quad \forall ss^k_{n,i} \in A^n_S \\
&\mathrm{C4}: \sum_{f \in F_n} x(sr^k_{n,i,j}, f) \le C(sr^k_{n,i,j}) \cdot \hat{y}_n(sr^k_{n,i,j}), \quad \forall sr^k_{n,i,j} \in A^n_{RL} \\
&\mathrm{C5}: \sum_{f \in F_{n,i}} x(os^k_{n,i,j}, f) \le C(os^k_{n,i,j}) \cdot y(os^k_{n,i,j}), \quad \forall os^k_{n,i,j} \in A^n_O \\
&\mathrm{C6}: \sum_{f \in F_n} x(sg^k_{n,i,j}, f) \le C(sg^k_{n,i,j}) \cdot y(sg^k_{n,i,j}), \quad \forall sg^k_{n,i,j} \in A^n_{GL} \\
&\mathrm{C7}: \sum_{j: sr^k_{n,i,j} \in A^n_{RL}} \hat{y}_n(sr^k_{n,i,j}) \le 1, \quad \forall 1 \le k \le K, 1 \le i \le |S_n| \\
&\mathrm{C8}: \sum_{j,k: os^k_{n,i,j} \in A^n_O} y(os^k_{n,i,j}) \le 1, \quad \forall 1 \le i \le |OM_n| \\
&\mathrm{C9}: \sum_{i: os^k_{n,i,j} \in A^n_O} y(os^k_{n,i,j}) \le 1, \quad \forall 1 \le k \le K, 1 \le j \le |S_n| \\
&\mathrm{C10}: \sum_{i: sg^k_{n,i,j} \in A^n_{GL}} y(sg^k_{n,i,j}) \le 1, \quad \forall 1 \le k \le K, 1 \le j \le |G_n| \\
&\mathrm{C11}: \sum_{j: sg^k_{n,i,j} \in A^n_{GL}} y(sg^k_{n,i,j}) \le 1, \quad \forall 1 \le k \le K, 1 \le i \le |S_n|
\end{aligned}$$

Similarly, the feasible solution set of the DRS system can be obtained

$$\xi_r = \left\{ \hat{\boldsymbol{y}}^C_r \middle| \begin{array}{l} \sum\limits_{n,i: sr^k_{n,i,j} \in A_{RL}} \hat{y}_r(sr^k_{n,i,j}) \le 1, \quad \forall 1 \le k \le K, 1 \le j \le |R|, \\ 0 \le \hat{y}_r(sr^k_{n,i,j}) \le 1, \quad \forall sr^k_{n,i,j} \in A_{RL} \end{array} \right\}. \tag{28}$$

In order to further divide the coordinated resource scheduling problem with respect to the local variables, we define the local cost function for the resource scheduling of RSS system n as

$$v_n(\boldsymbol{z}_n, \boldsymbol{x}_n, \boldsymbol{y}^L_n, \hat{\boldsymbol{y}}^C_n) = \begin{cases} -\sum\limits_{1 \le i \le |OM_n|} z_{n,i}, & \{\boldsymbol{z}_n, \boldsymbol{x}_n, \boldsymbol{y}^L_n, \hat{\boldsymbol{y}}^C_n\} \in \xi_n \\ +\infty, & \text{otherwise} \end{cases} \tag{29}$$

Similarly, the local cost function for the DRS system is given by

$$v_r(\hat{\boldsymbol{y}}_r^C) = \begin{cases} 0, & \hat{\boldsymbol{y}}_r^C \in \xi_r \\ +\infty, & \text{otherwise} \end{cases} \tag{30}$$

Based on Equations (27)–(30), P3 can be compactly rewritten as

$$\begin{aligned} \text{P4}: \quad & \min \sum_{1\le n\le N} v_n(\boldsymbol{z}_n, \boldsymbol{x}_n, \boldsymbol{y}_n^L, \hat{\boldsymbol{y}}_n^C) + v_r(\hat{\boldsymbol{y}}_r^C) \\ \text{s.t.} \quad & \hat{y}_n(sr_{n,i,j}^k) = y(sr_{n,i,j}^k), \quad \forall sr_{n,i,j}^k \in A_{RL} \\ & \hat{y}_r(sr_{n,i,j}^k) = y(sr_{n,i,j}^k), \quad \forall sr_{n,i,j}^k \in A_{RL} \end{aligned}$$

It can be seen that the objective of P4 is separable across the RSS and DRS systems, so that each system can handle the corresponding subproblem independently. Meanwhile, the consensus constraints guarantee the consistency of the local copies held in the RSS and DRS systems. As P4 is a typical global consistency optimization problem, it is solved in a distributed manner with the Alternating Direction Multiplier Method (ADMM) [25] in the next section.

4. Distributed Coordinated Resource Scheduling Algorithm Design Based on ADMM

This section presents a distributed iterative method based on the Alternating Direction Method of Multipliers (ADMM). Firstly, we tackle the consistency constraints of the problem P4 by constructing an augmented Lagrangian formulation, enabling the complete decomposition of the problem into subproblems that can be independently solved by each satellite system. Subsequently, we develop a distributed coordinated resource scheduling algorithm based on the ADMM which effectively solves the problem in parallel.

4.1. Augmented Lagrangian and ADMM Sequential Iterations

Although the objective of P4 is in a separable form, its different parts are still coupled through the consistency constraint. In order to remove this obstacle to completely splitting P4, an augmented Lagrangian form is constructed for P4 [25,26], which is given as follows

$$\begin{aligned} & L_\rho\left(\{\boldsymbol{z}_n, \boldsymbol{x}_n, \boldsymbol{y}_n^L, \hat{\boldsymbol{y}}_n^C\}_{1\le n\le N}, \{\hat{\boldsymbol{y}}_r^C\}, \boldsymbol{y}^C, \{\boldsymbol{\lambda}_n\}_{1\le n\le N}, \boldsymbol{\lambda}_r\right) \\ & = \sum_{1\le n\le N} v_n(\boldsymbol{z}_n, \boldsymbol{x}_n, \boldsymbol{y}_n^L, \hat{\boldsymbol{y}}_n^C) + v_r(\hat{\boldsymbol{y}}_r) + \sum_{sr_{n,i,j}^k \in A_{RL}} \lambda_n(sr_{n,i,j}^k)\left(\hat{y}_n(sr_{n,i,j}^k) - y(sr_{n,i,j}^k)\right) \\ & \quad + \sum_{sr_{n,i,j}^k \in A_{RL}} \lambda_r(sr_{n,i,j}^k)\left(\hat{y}_r(sr_{n,i,j}^k) - y(sr_{n,i,j}^k)\right) + \frac{\rho}{2} \sum_{sr_{n,i,j}^k \in A_{RL}} \left(\hat{y}_n(sr_{n,i,j}^k) - y(sr_{n,i,j}^k)\right)^2 \\ & \quad + \frac{\rho}{2} \sum_{sr_{n,i,j}^k \in A_{RL}} \left(\hat{y}_r(sr_{n,i,j}^k) - y(sr_{n,i,j}^k)\right)^2 \end{aligned} \tag{31}$$

where $\boldsymbol{\lambda}_n = \left(\{\lambda_n(sr_{n,i,j}^k)\}_{sr_{n,i,j}^k \in A_{RL}^n}\right)$ and $\boldsymbol{\lambda}_r = \left(\{\lambda_r(sr_{n,i,j}^k)\}_{sr_{n,i,j}^k \in A_{RL}}\right)$ are the Lagrange multipliers with respect to consistency constraints, and $\rho \in \mathbb{R}_{++}$ is the penalty term, which is a constant parameter intended for adjusting the convergence speed of the ADMM [27]. Compared with the standard Lagrangian form, the additional quadratic regularization term in augmented Lagrangian can improve the convergence efficiency of the iterative method. Moreover, the additional quadratic regularization term is equal to zero for any feasible solution, i.e., the optimal solution of Equation (31) is equivalent to that of P4. As Equation (31) can be completely decomposed into $N+1$ independent parts, P4 can be solved through sequential iterative local optimization for the RSS and DRS systems with the application of the ADMM. Let t denote the iteration index. The specific iteration steps are as follows:

Step 1. Updating Local Variables:

In each iteration, local variables are update by minimizing the augmented Lagrangian in Equation (31). Specifically, the RSS system n updates its local variables by

$$\left\{\boldsymbol{z}_n^{t+1}, \boldsymbol{x}_n^{t+1}, \boldsymbol{y}_n^{L,t+1}, \hat{\boldsymbol{y}}_n^{C,t+1}\right\}_{1\le n\le N} = \underset{\{\boldsymbol{z}_n,\boldsymbol{x}_n,\boldsymbol{y}_n^L,\hat{\boldsymbol{y}}_n^C\}}{\arg\min} \left\{ \begin{array}{l} v_n(\boldsymbol{z}_n, \boldsymbol{x}_n, \boldsymbol{y}_n^L, \hat{\boldsymbol{y}}_n^C) + \sum\limits_{sr_{n,i,j}^k \in A_{RL}^n} \lambda_n^t(sr_{n,i,j}^k)\left(\hat{y}_n(sr_{n,i,j}^k) - y^t(sr_{n,i,j}^k)\right) \\ +\frac{\rho}{2} \sum\limits_{sr_{n,i,j}^k \in A_{RL}^n} \left(\hat{y}_n(sr_{n,i,j}^k) - y^t(sr_{n,i,j}^k)\right)^2 \end{array} \right\} \tag{32}$$

Moreover, the DRS system updates its local variables via

$$\hat{\boldsymbol{y}}_r^{C,t+1} = \underset{\hat{\boldsymbol{y}}_r^C}{\arg\min} \left\{ \begin{array}{l} v_n(\hat{\boldsymbol{y}}_r^C) + \sum\limits_{sr_{n,i,j}^k \in A_{RL}} \lambda_r^t(sr_{n,i,j}^k)\left(\hat{y}_r(sr_{n,i,j}^k) - y^t(sr_{n,i,j}^k)\right) \\ +\frac{\rho}{2} \sum\limits_{sr_{n,i,j}^k \in A_{RL}} \left(\hat{y}_r(sr_{n,i,j}^k) - y^t(sr_{n,i,j}^k)\right)^2 \end{array} \right\} \tag{33}$$

It can be seen from Equations (32) and (33) that step 1 can be solved by the N RSS systems and the DRS system in parallel. For RSS system n, by dropping the constant term in Equation (32), which does not affect the solution, we can obtain the subproblem to be solved by the RSS system $n \in N$ in the $(t+1)$th iteration

$$\begin{array}{ll} \text{P5}: \min - \sum\limits_{1\le i\le |OM_n|} z_{n,i} + \sum\limits_{sr_{n,i,j}^k \in A_{RL}^n} \left(\lambda_n^t(sr_{n,i,j}^k)\hat{y}_n(sr_{n,i,j}^k) + \frac{\rho}{2}\left(\hat{y}_n(sr_{n,i,j}^k) - y^t(sr_{n,i,j}^k)\right)^2\right) \\ \quad\text{s.t.} \ \left\{\boldsymbol{z}_n, \boldsymbol{x}_n, \boldsymbol{y}_n^L, \hat{\boldsymbol{y}}_n^C\right\} \in \xi_n \end{array}$$

Similarly, the subproblem to be solved by the DRS system in the $t+1$ iteration is given by

$$\begin{array}{ll} \text{P6}: & \min \sum\limits_{sr_{n,i,j}^k \in A_{RL}} \left(\lambda_r^t(sr_{n,i,j}^k)\hat{y}_r(sr_{n,i,j}^k) + \frac{\rho}{2}\left(\hat{y}_r(sr_{n,i,j}^k) - y^t(sr_{n,i,j}^k)\right)^2\right) \\ \text{s.t.} & \hat{\boldsymbol{y}}_r^C \in \xi_r \end{array}$$

On account of the quadratic objective and convex feasible set of problems P5 and P6, both the problems are convex optimization problems that can be solved directly by applying classical methods such as the primal-dual interior-point method [28] or optimization software such as CVX.

Step 2. Updating Global Variables:

After the update of local variables, the global variables are updated by minimizing the augmented Lagrangian in each iteration, which is given as follows

$$\boldsymbol{y}^{C,t+1} = \underset{\boldsymbol{y}^C}{\arg\min} \left\{ \begin{array}{l} \sum\limits_{sr_{n,i,j}^k \in A_{RL}} \lambda_n^t(sr_{n,i,j}^k)\left(\hat{y}_n^{t+1}(sr_{n,i,j}^k) - y(sr_{n,i,j}^k)\right) \\ + \sum\limits_{sr_{n,i,j}^k \in A_{RL}} \lambda_r^t(sr_{n,i,j}^k)\left(\hat{y}_r^{t+1}(sr_{n,i,j}^k) - y(sr_{n,i,j}^k)\right) \\ +\frac{\rho}{2} \sum\limits_{sr_{n,i,j}^k \in A_{RL}} \left(\hat{y}_n^{t+1}(sr_{n,i,j}^k) - y(sr_{n,i,j}^k)\right)^2 \\ +\frac{\rho}{2} \sum\limits_{sr_{n,i,j}^k \in A_{RL}} \left(\hat{y}_r^{t+1}(sr_{n,i,j}^k) - y(sr_{n,i,j}^k)\right)^2 \end{array} \right\} \tag{34}$$

Since Equation (34) is an unconstrained quadratic programming problem, which is strictly convex owing to the additional quadratic regularization terms, the global variable $\boldsymbol{y}^{C,t+1}$ can be solved by setting its gradient to 0, i.e.,

$$\rho\left(\hat{y}_n^{t+1}(sr_{n,i,j}^k) + \hat{y}_r^{t+1}(sr_{n,i,j}^k) - 2y(sr_{n,i,j}^k)\right) + \lambda_n^t(sr_{n,i,j}^k) + \lambda_r^t(sr_{n,i,j}^k) = 0, \ \forall sr_{n,i,j}^k \in A_{RL} \tag{35}$$

Then we can obtain

$$y^{t+1}(sr^k_{n,i,j}) = \frac{1}{2\rho}\left(\lambda^t_n(sr^k_{n,i,j}) + \lambda^t_r(sr^k_{n,i,j})\right) + \frac{1}{2}\left(\hat{y}^{t+1}_n(sr^k_{n,i,j}) + \hat{y}^{t+1}_r(sr^k_{n,i,j})\right), \ \forall sr^k_{n,i,j} \in A_{RL} \tag{36}$$

By initializing the Lagrange multiplier as zero, the above equation can be reduced to

$$y^{t+1}(sr^k_{n,i,j}) = \frac{1}{2}\left(\hat{y}^{t+1}_n(sr^k_{n,i,j}) + \hat{y}^{t+1}_r(sr^k_{n,i,j})\right), \quad \forall sr^k_{n,i,j} \in A_{RL} \tag{37}$$

That is, in each iteration, the global variable $y^{t+1}(sr^k_{n,i,j})$ can be obtained by averaging the corresponding local copies of RSS system n and the DRS system.

Step 3. Updating Lagrange multiplier:

Applying the penalty parameter ρ as step size, the Lagrange multipliers are updated as

$$\left\{\boldsymbol{\lambda}^{t+1}_n\right\}_{1\le n\le N} = \boldsymbol{\lambda}^t_n + \rho(\hat{\boldsymbol{y}}^{C,t+1}_n - \boldsymbol{y}^{C,t+1}_n) \tag{38}$$

$$\boldsymbol{\lambda}^{t+1}_r = \boldsymbol{\lambda}^t_r + \rho(\hat{\boldsymbol{y}}^{C,t+1}_r - \boldsymbol{y}^{C,t+1}_r) \tag{39}$$

Equations (38) and (39) imply that step 3 can be executed by N RSS systems and the DRS systems in parallel. More specifically, each RSS system updates its associate Lagrange multipliers by Equation (38), and the DRS system updates its associate Lagrange multipliers by Equation (39).

In the above iterative solving process, steps 1 and 3 can be divided into $N + 1$ independent parts to be executed in parallel by each satellite system, while step 2 can be executed by the DRS system. In other words, through exchanging the global variables and their local copies between RSS systems and the relay satellite system, the iterative steps can be implemented in a distributed manner. Upon the completion of the iterative steps, the relaxed problem P2 is solved. Note that P2 is obtained by relaxing binary variables $\boldsymbol{y}$ and $\boldsymbol{z}$ of the original problem P1. Therefore, the relaxed variables should be recovered by the RSS and DRS systems after the sequential iterative optimization steps, so that the solution of coordinated resource scheduling problem can be obtained. In the next subsection, a distributed coordinated resource scheduling algorithm based on the ADMM is developed for the RSS and DRS systems to solve P1 coordinately in a distributed manner, and two relaxed variable recovery algorithms are proposed for the DRS system and RSS systems to recover the global and local relaxed variables, respectively.

4.2. Algorithm Implementation

Algorithm 1 summarizes the ADMM-based distributed coordinated resource scheduling process of SGCN. To be specific, lines 3–7 are the implementation of iteration steps 1 to 3 and the associated information exchange between the RSS system and the DRS system. Line 10 indicates the termination criterion of the iteration, i.e., the DRS system informs each RSS system of the termination of the iteration when the global variables converge, where ε is a small positive number. In lines 11–12, the global variables $\boldsymbol{y}^C$ and local variables $\boldsymbol{y}^L_n$, $\boldsymbol{z}_n$ are, respectively, recovered by the DRS system and RSS systems by employing Algorithms 2 and 3.

Algorithm 2 is the recovery algorithm of the relaxed global variable $\boldsymbol{y}^C$ for the DRS system. The main idea is to fix the global relaxed variables sequentially in descending order while avoiding conflicts with the already fixed variables. $AC(sr^k_{n,i,j})$ denotes the set of link arcs in conflict with link arc $sr^k_{n,i,j}$, which is defined as

$$AC(sr^k_{n,i,j}) = \left\{sr^k_{n,i,q} | sr^k_{n,i,q} \in A_{RL}\right\} \cup \left\{sr^k_{m,p,j} | sr^k_{m,p,j} \in A_{RL}\right\} - \left\{sr^k_{n,i,j}\right\} \tag{40}$$

Algorithm 1 Distributed coordinated resource scheduling algorithm based on the ADMM

Input: The mission requests and resource information of each satellite system.
Output: Optimal resource scheduling results of each RSS system $\{\boldsymbol{x}_n^*, \boldsymbol{y}_n^*, \boldsymbol{z}_n^*\}_{1 \le n \le N}$.
Initialize: $t \leftarrow 0, \boldsymbol{y}^{C,t} \leftarrow \boldsymbol{0}, \boldsymbol{\lambda}^t \leftarrow \boldsymbol{0}$;
repeat
  Each RSS system solves problem P5 to update the local variables $\{\boldsymbol{z}_n^{t+1}, \boldsymbol{x}_n^{t+1}, \boldsymbol{y}_n^{L,t+1}, \hat{\boldsymbol{y}}_n^{C,t+1}\}$;
  DRS system solves problem P6 update local variables $\hat{\boldsymbol{y}}_r^{C,t+1}$;
  Each RSS system sends local copy variables $\hat{\boldsymbol{y}}_n^{C,t+1}$ to the DRS system;
  The DRS system updates the global variable $\boldsymbol{y}^{C,t+1}$ through Equation (30), and distributes them to the corresponding RSS system;
  Each RSS system updates the Lagrange multiplier $\boldsymbol{\lambda}_n^{t+1}$ by Equation (31);
  The DRS system updates the Lagrange multiplier $\boldsymbol{\lambda}_r^{t+1}$ through Equation (32);
  $t \leftarrow t+1$;
until $\|\boldsymbol{y}^{C,t+1} - \boldsymbol{y}^{C,t}\|_2 \le \varepsilon$
DRS system executes Algorithm 2 to recover the relaxed variables $\boldsymbol{y}^C$ and sends $(\boldsymbol{y}_n^C)^*$ to each RSS system n;
Each RSS system recovers relaxed variables $\boldsymbol{y}_n^L$ and $\boldsymbol{z}_n$ via Algorithm 3 .

Algorithm 2 Recovery of global relaxed variables in the DRS system

Input: relaxed variable $\boldsymbol{y}^C$.
Output: $\{(\boldsymbol{y}_n^C)^*\}_{1 \le n \le N}$.
Initialize: $A_X \leftarrow A_{RL}$;
while $A_X \ne \varnothing$ is true **do**
  $sr_{n,i,j}^k \leftarrow \arg\max_{sr_{m,p,q}^l \in A_X} y(sr_{m,p,q}^l)$;
  $y^*(sr_{n,i,j}^k) \leftarrow 1$;
  $A_X \leftarrow A_X - \left\{sr_{n,i,j}^k\right\}$;
  for $sr_{m,p,q}^l \in \left(A_X \cap AC(sr_{n,i,j}^k)\right)$ **do**
    $y^*(sr_{m,p,q}^l) \leftarrow 0$;
    $A_X \leftarrow A_X - \left\{sr_{m,p,q}^l\right\}$;
  end for
end while

Algorithm 3 is the recovery algorithm of the local relaxed variables $\boldsymbol{y}_n^L$ and $\boldsymbol{z}_n$ for each RSS system n. Its main idea is to select a mission and allocate observation and transmission resources alternatively. Specifically, it first fixes the mission scheduling variable $z_{n,e}$ with the largest relaxed value. Then, it allocates observation and transmission resources to the fixed mission by finding a path on graph $G_R(V_R, A_R)$, which is a subgraph of G_K related to RSS system n. The value of the local relaxed variables associated with the observation arcs and the links arcs on the path can be fixed. Finally, update $G_R(V_R, A_R)$ based on the resource allocation results and substitute the recovered variables into problem P5 and solve it again. The above process is repeated until all mission planning variables have been determined. The initial values of V_R and A_R are, respectively, defined as

$$
\begin{aligned}
V_R &= V_o^n \cup V_s^n \cup V_g^n \cup V_r \cup V_{VT}, \\
A_R &= A_L^n \cup A_O^n \cup A_S^n \cup A_{VT}^n,
\end{aligned} \tag{41}
$$

where V_o^n, V_s^n, and V_g^n denote the temporal replicas of the observation objects, RSSs, and ground stations of RSS system n, respectively. For instance, $V_o^n = \{o_{n,i}^k | 1 \le k \le K, 1 \le i \le |OM_n|\}$. A_L^n, A_O^n, A_S^n, and A_{VT}^n represent the set of link arcs, observation arcs, storage arcs, and virtual arcs related to the RSS system n, respectively.

Algorithm 3 Recovery of local relaxed variable in RSS system n

Input: $(\boldsymbol{y}_n^C)^*$,relaxed variable $\boldsymbol{y}^C$.
Output: $\{\boldsymbol{x}_n^*, \boldsymbol{y}_n^*, \boldsymbol{z}_n^*\}_{1 \le n \le N}$.
Initialize:$G_R(V_R, A_R)$, $OM_R \leftarrow OM_n$;
while $OM_R \neq \varnothing$ is true **do**
  $e \leftarrow \arg\max_{i:om_{n,i} \in OM_R} z_{n,i}$;
  if There exists a path $p_0 = o_{n,e}^{sk} \rightarrow u^{ek}$ in G_R satisfying $\left\lceil \frac{st_{n,e}}{\tau} \right\rceil \le sk \le \left\lceil \frac{et_{n,e}}{\tau} \right\rceil, ek = \left\lceil \frac{et_{n,e}}{\tau} \right\rceil$ and the capacity is not less than $b_{n,e}$ **then**
    $z_{n,e}^* \leftarrow 1,\ OM_R \leftarrow OM_R - \{z_{n,e}\}$;
    for $a \in p_0 \cap (A_L^n \cup A_O^n)$ **do**
      $y^*(a) \leftarrow 1$;
      for $a' \in A_R \cap AC(a)$ **do**
        $y^*(a') \leftarrow 0,\ A_R \leftarrow A_R - \{a'\}$;
      end for
    end for
    for $a \in p_0$ **do**
      $C(a) \leftarrow C(a) - b_{n,e}$;
    end for
  else
    $z_{n,e}^* \leftarrow 0,\ OM_R \leftarrow OM_R - \{z_{n,e}\}$;
  end if
  Substitute the recovered variables into problem P5 and solve the problem again;
end while

It can be seen from Algorithms 1–3 that the information interchange among different satellite systems is very limited in the whole distributed resource scheduling process. The only information that needs to be exchanged is intersatellite link scheduling variables (i.e., $\boldsymbol{y}^C$ and $\hat{\boldsymbol{y}}_n^C$). More specifically, the DRS system receives the local intersatellite link scheduling variables $\hat{\boldsymbol{y}}_n^C$ from the RSS systems and feeds back corresponding global intersatellite link scheduling variables $\boldsymbol{y}^C$ to them. Other local information, such as mission requests and ground–satellite scheduling states of each RSS system, is well-preserved. Therefore, the proposed distributed coordinated resource scheduling algorithm can protect the local privacy information better than the traditional methods [12–14].

5. Simulations

5.1. Simulation Setup and Results Description

We conducted simulations via STK and MATLAB to evaluate the performance of the proposed distributed coordinated resource scheduling (DCRS) algorithm. A simulation scenario comprising 3 RSS systems and a DRS system is considered. Specifically, each RSS system consists four RSSs and two ground stations. The DRS system has three DRS on the geosynchronous orbit. The RSSs are distributed in sun-synchronous orbits with a height from 619.6 km to 778 km and inclination from 97.8° to 98.5°. The transmission rate of the links from RSSs to ground stations and DRSs are 50 Mbps. The storage volume of each RSS and DRS is 200 Gbit. The scheduling horizon is 1 day (from 13 May 2022 04:00:00 to 14 May 2022 04:00:00). We randomly generate 50–200 missions for each RSS system. The observation objects are randomly distributed on Earth with uniform distribution. The amount of data required to be acquired follows a uniform distribution from 5 Gbit to 15 Gbit. The earliest start time follows a uniform distribution from 0 to 86,400 s.

To validate the performance of the proposed DCRS algorithm, we consider the following three scheduling schemes for comparison:

1. No Relay Resource Scheduling (NRRS): The DRS system does not provide relay service for the remote sensing missions, i.e., the observed mission data can only be transmitted to the ground through the ground stations of the belonging RSS system.

2. Non-Coordinated Resource Scheduling (NCRS): The remote sensing missions are scheduled in two stages. In the first stage, each RSS system allocates observation resources and local communication resources to their missions and then sends relay request to the DRS system for the missions lacking communication resources. In the second stage, the DRS system assigns the relay resources.
3. Centralized Coordinated Resource Scheduling (CCRS): There exists a central server to schedule the missions of all the RSS systems with global network information in a centralized manner. Note that CCRS is employed as a baseline algorithm because it is a centralized coordinated method with the ideal condition.

The proposed DCRS algorithm and the above three scheduling schemes are implemented through MATLAB programming. By testing them in our simulation scenario, we compare the proposed DCRS algorithm with other three scheduling schemes in terms of the following three metrics [10,13]:

1. Number of successfully scheduled missions: The number of missions which have been successfully scheduled after employing the proposed mission schedule algorithm or the comparing algorithm.
2. Total working time of the ground stations: The sum time that all the ground stations used to receive data from RSSs in the scheduling horizon.
3. Total working time of the DRSs: The sum time that all the DRSs used to receive data from RSSs in the scheduling horizon.

We conducted two groups of simulation experiments. The first group of simulation experiments compares the performance of different schemes with increasing mission numbers. The results of the first group of simulation experiments are shown in Figures 3–5, which, respectively, investigate the number of successfully scheduled missions, total working time of the ground stations, and total working time of the DRSs. The first group of simulation experiments compares the performance of different schemes with increasing mission numbers. The second group of simulation experiments compares of different schemes with varying the length of feasible scheduling window (i.e., the difference value between the earliest start time and latest end time of missions). Specifically, we fix the number of mission requests to be 200 and set the feasible scheduling windows to be 2, 6, 10, and 14 h. The results of the second group of simulation experiments are shown in Figures 6–8, which, respectively, investigate the number of successfully scheduled missions, total working time of the ground stations, and total working time of the DRSs.

5.2. Results Analysis and Discussion

We can observe from Figure 3 that the proposed DCRS algorithm performs much better than the NRRS and NCRS schemes and close to the CCRS scheme in terms of the number of completed missions. To be specific, the maximum performance between the proposed algorithm and the CCRS scheme is no more than 6%. That is to say, the proposed algorithm performs closed to the ideal centralized benchmark scheme via a distributed manner. What is more, there is an obvious gap between the NRRS scheme and the other three schemes with the support of DRS system, which verifies the importance of the relay resource. Moreover, our algorithm can achieve about 24% performance gain from NCRS, which mainly comes from the advantages of coordinated scheduling of the resources of RSS systems and the DRS system.

To further discuss the source of the performance gain of the proposed algorithm, the total working time of the ground stations and DRSs of different schemes are investigated from Figures 4 and 5. Through comparing the two figures, it can be found that most (more than 80%) of the observed data are downloaded through DRSs for the proposed DCRS scheme, and NCRS and CCRS are nearly so.

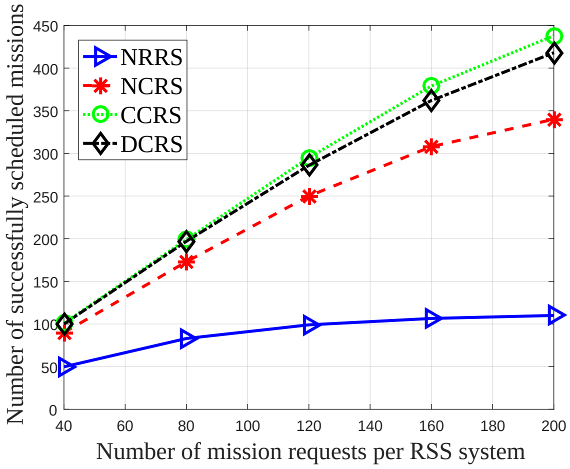

Figure 3. Number of successfully scheduled missions versus the number of mission requests per RSS system.

More specifically, the NRRS has the largest ground station working time, while the NCRS has the least, as shown in Figure 4. Moreover, the total working time of ground stations under the CCRS and DCRS is very close. This is because under the NRRS scheme, only ground stations can be used to download acquired data. NCRS always overestimate the relay resource that can be used due to lack of coordination among RSS systems and the DRS system. As we can observe from Figure 5, the total DRS working time of proposed DCRS algorithm is 20% larger than NCRS scheme, and it is close to that of CCRS. This is because in the second stage of NCRS, collision of the relay resources may be raised among the RSS systems due to the lack of coordination. Therefore, compared with NCRS, the proposed DCRS gains the resource utilization improvements of both ground stations and DRSs from the advantage of coordination.

Figure 6 depicts the number of successfully scheduled missions with increasing length of feasible scheduling window. It can be observed that the performance of the proposed DCRS scheme is very close to the CCRS scheme for the missions with either short or long feasible scheduling windows. The gain of the proposed DCRS scheme compared to the NCRS scheme increases with feasible scheduling window length. Moreover, compared with the nearly linear increase in the NRRS scheme, the growth rate of DCRS slows down when the length of the feasible scheduling window becomes large.

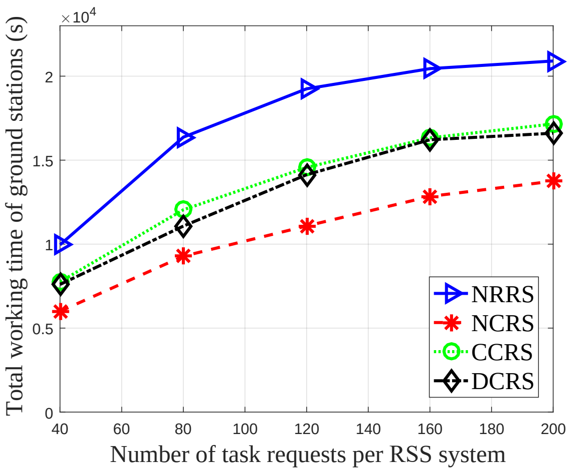

Figure 4. Total working time of the ground stations versus the number of mission requests per RSS system.

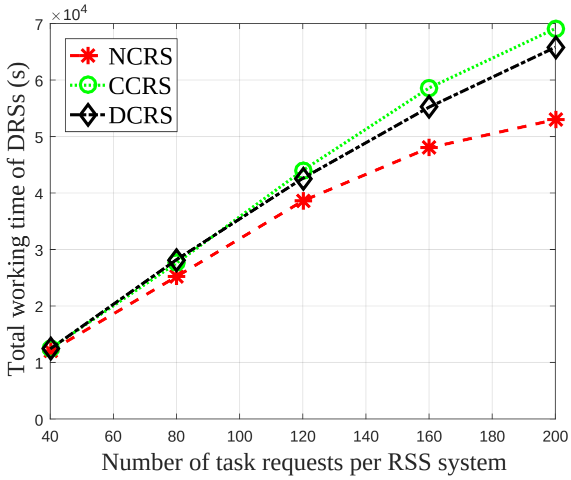

Figure 5. Total working time of the DRSs versus the number of mission requests per RSS system.

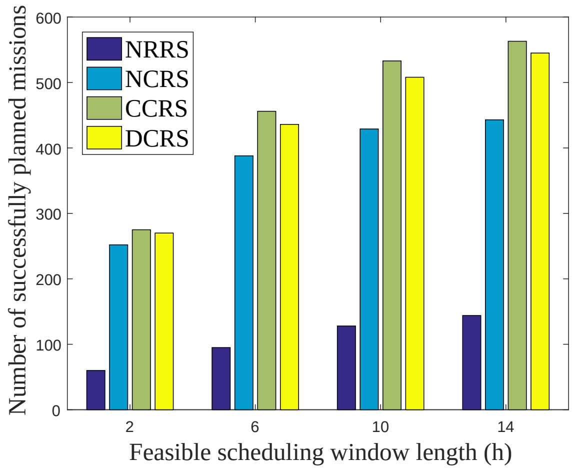

Figure 6. Number of successfully scheduled missions versus the length of feasible scheduling window.

To further dig up the philosophy behind the performance of different lengths of feasible scheduling windows, we investigate the total working time of the ground stations and DRSs with varying lengths of feasible scheduling windows, which are depicted in Figures 7 and 8. As we can see, the working time of ground stations increases approximately linearly with the length of feasible scheduling windows, while the growth working time of DRSs is small, especially when the feasible scheduling window is long. This is because for the missions downloaded through ground stations, both observation and transmission may need to wait a long time for the RSS to fly above the observation object/ground station. In comparison, for the missions downloaded through DRSs, only the observation process may require a long time, because an RSS is visible to at least one DRS most of the time. Moreover, we can observe that the improvement of the proposed algorithm compared to NCRS increases with the feasible scheduling window length in both Figures 7 and 8. This is because the longer the feasible scheduling window, the larger the degree of freedom for resource allocation. With the coordination of the DRS systems and RSS systems, the DCRS scheme can well utilize the degree of freedom to avoid the collisions of relay resources, while NCRS cannot due to the lack of global information.

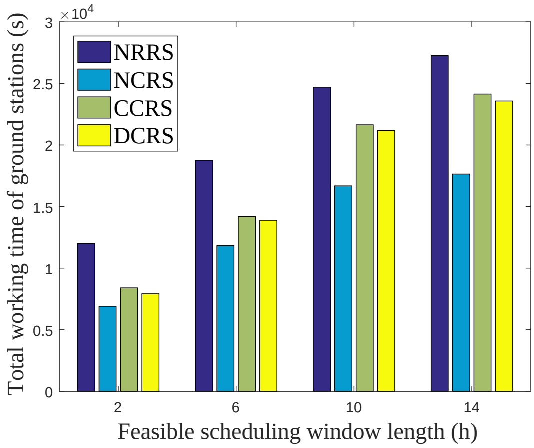

Figure 7. Total working time of the ground stations versus the length of feasible scheduling windows.

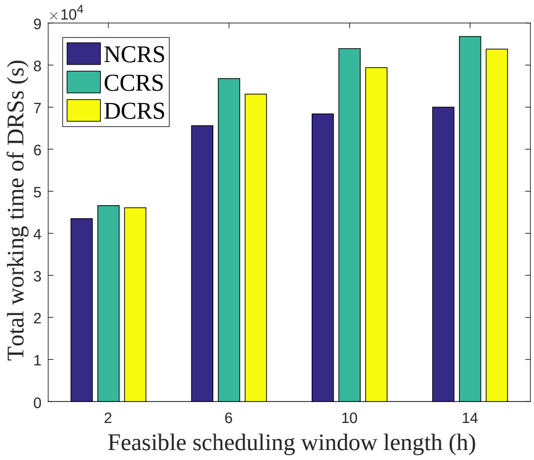

Figure 8. Total working time of the DRSs versus the length of feasible scheduling windows.

6. Conclusions

This paper studies a distributed resource scheduling method to realize the coordination of space–ground multiresources for remote sensing missions. Compared with the centralized resource scheduling method, our method requires no extra coordination center added in the heterogeneous satellite systems. Moreover, it only requires very limited link scheduling information interchange among different satellite systems. Other local information is well-preserved. Therefore, the proposed method is much easier to apply to real systems than centralized coordinated scheduling methods. Simulation results shows that the number of completed missions of the proposed algorithm is improved about 24% compared with NCRS, and it achieves no less than 96% of the centralized coordinated scheduling algorithm. However, as the proposed algorithm is realized by iteration, it requires more computation time and times of information exchanging among the RRS and DRS systems than the noncoordinated scheduling algorithm and centralized coordinated scheduling algorithm. Therefore, it is only suitable for the offline schedule of the nonemergency missions, which can be planned in advance by the OCCs on the ground, and cannot be employed in online schedules for emergency missions. In our future work, we will study the distributed online coordinated scheduling algorithm for emergency remote sensing missions.

Author Contributions: Conceptualization, R.L.; methodology, R.L. and W.W.; simulation, X.D.; validation, W.G.; formal analysis, R.L. and W.W.; writing—original draft preparation, X.D.; writing—review and editing, R.L., W.W. and W.G.; visualization, X.D.; supervision, W.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 62001349), Natural Science Basic Research Program of Shaanxi (No. 2023-JC-YB-566, No. 2023-JC-YB-542), China Postdoctoral Science Foundation (No. 2018M633462), Key Research and Development Program in Shaanxi Province of China (No. 2021GY-066), Young Talent fund of University Association for Science and Technology in Shaanxi, China (No. 20200112), Postdoctoral Foundation in Shaanxi Province of China (No. 2018BSHEDZZ47).

Data Availability Statement: No new data were created or analyzed in this study.

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

References

1. Wu, X.; Yang, Y.; Sun, Y.; Xie, Y.; Song, X.; Huang, B. Dynamic regional splitting planning of remote sensing satellite swarm using parallel genetic PSO algorithm. *Acta Astronaut.* **2023**, *204*, 531–551. [CrossRef]
2. Liu, R.; Wu, W.; Yang, Q.; Zhou, D.; Zhang, W. Exploring the information capacity of remote sensing satellite networks. *IEEE Access* **2020**, *8*, 34056–34070. [CrossRef]
3. Katona, Z.; Gräßlin, M.; Donner, A.; Kranich, N.; Brandt, H.; Bischl, H.; Brück, M. A flexible LEO satellite modem with Ka-band RF frontend for a data relay satellite system. *Int. J. Satell. Commun. Netw.* **2020**, *38*, 301–313. [CrossRef]
4. Zhou, D.; Sheng, M.; Li, J.; Han, Z. Aerospace Integrated Networks Innovation for Empowering 6G: A Survey and Future Challenges. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 975–1019. [CrossRef]
5. Wu, G.; Liu, J.; Ma, M.; Qiu, D. A two-phase scheduling method with the consideration of task clustering for earth observing satellites. *Comput. Oper. Res.* **2013**, *40*, 1884–1894. [CrossRef]
6. Wang, P.; Reinelt, G.; Gao, P.; Tan, Y. A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation. *Comput. Ind. Eng.* **2011**, *61*, 322–335. [CrossRef]
7. Du, Y.; Wang, T.; Xin, B.; Wang, L.; Chen, Y.; Xing, L. A data-driven parallel scheduling approach for multiple agile earth observation satellites. *IEEE Trans. Evol. Comput.* **2019**, *24*, 679–693. [CrossRef]
8. Zhu, W.; Hu, X.; Xia, W.; Sun, H. A three-phase solution method for the scheduling problem of using earth observation satellites to observe polygon requests. *Comput. Ind. Eng.* **2019**, *130*, 97–107. [CrossRef]
9. Zhou, D.; Sheng, M.; Liu, R.; Wang, Y.; Li, J. Channel-aware mission scheduling in broadband data relay satellite networks. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 1052–1064. [CrossRef]
10. Chen, X.; Li, X.; Wang, X.; Luo, Q.; Wu, G. Task scheduling method for data relay satellite network considering breakpoint transmission. *IEEE Trans. Veh. Technol.* **2020**, *70*, 844–857. [CrossRef]
11. Song, Y.; Xing, L.; Wang, M.; Yi, Y.; Xiang, W.; Zhang, Z. A knowledge-based evolutionary algorithm for relay satellite system mission scheduling problem. *Comput. Ind. Eng.* **2020**, *150*, 106830. [CrossRef]
12. Wang, Y.; Sheng, M.; Zhuang, W.; Zhang, S.; Zhang, N.; Liu, R.; Li, J. Multi-resource coordinate scheduling for earth observation in space information networks. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 268–279. [CrossRef]
13. Liu, R.; Li, J.; Ji, X.; Wu, W.; Zhou, D.; Zhang, Y. Two-Stage Task Planning Based on Resource Interchange in Space Information Networks. In Proceedings of the International Conference on Communications and Networking in China, Hangzhou, China, 20–21 November 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 644–659.
14. Waiming, Z.; Xiaoxuan, H.; Wei, X.; Peng, J. A two-phase genetic annealing method for integrated Earth observation satellite scheduling problems. *Soft Comput.* **2019**, *23*, 181–196. [CrossRef]
15. Meng, X.; Wu, L.; Yu, S. Research on resource allocation method of space information networks based on deep reinforcement learning. *Remote Sens.* **2019**, *11*, 448. [CrossRef]
16. Kaslow, D. Planning and scheduling of earth observing satellites. In Proceedings of the 2007 IEEE Aerospace Conference, Big Sky, MT, USA, 3–10 March 2007; IEEE: New York, NY, USA, 2007; pp. 1–12.
17. Klemich, K.S.; Cerrone, G.; Heinen, W. Earth observation mission planning. In *The FLP Microsatellite Platform*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 467–480.
18. Wang, P.; Zhang, X.; Zhang, S.; Li, H.; Zhang, T. Time-expanded graph-based resource allocation over the satellite networks. *IEEE Wirel. Commun. Lett.* **2018**, *8*, 360–363. [CrossRef]
19. Liu, R.; Sheng, M.; Lui, K.S.; Wang, X.; Wang, Y.; Zhou, D. An analytical framework for resource-limited small satellite networks. *IEEE Commun. Lett.* **2015**, *20*, 388–391. [CrossRef]
20. Ju, Y.; Chen, Y.; Cao, Z.; Liu, L.; Pei, Q.; Pei, Q.; Xiao, M.; Ota, M.; Dong, M.; Leung, V.C.M. Joint secure offloading and resource allocation for vehicular edge computing network: A multi-agent deep reinforcement learning approach. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 5555–5569. [CrossRef]
21. Wang, L.; Jiang, C.; Kuang, L.; Wu, S.; Huang, H.; Qian, Y. High-efficient resource allocation in data relay satellite systems with users behavior coordination. *IEEE Trans. Veh. Technol.* **2018**, *67*, 12072–12085. [CrossRef]
22. Zhu, Y.; Sheng, M.; Li, J.; Zhou, D.; Han, Z. Modeling and performance analysis for satellite data relay networks using two-dimensional markov-modulated process. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 3894–3907. [CrossRef]
23. Bertsekas, D.P. Nonlinear programming. *J. Oper. Res. Soc.* **1997**, *48*, 334–334. [CrossRef]

24. Garey, M.R.; Johnson, D.S. Computers and intractability: A Guide To The Theory of NP-Completeness; W. H. Freeman, New York, NY, USA, 1979.
25. Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends® Mach. Learn.* **2011**, *3*, 1–122.
26. Leinonen, M.; Codreanu, M.; Juntti, M. Distributed joint resource and routing optimization in wireless sensor networks via alternating direction method of multipliers. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 5454–5467. [CrossRef]
27. Tabrizi, H.; Peleato, B.; Farhadi, G.; Cioffi, J.M.; Aldabbagh, G. Spatial reuse in dense wireless areas: A cross-layer optimization approach via ADMM. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 7083–7095. [CrossRef]
28. Boyd, S.; Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.