

Article

Dynamic Merging for Optimal Onboard Resource Utilization: Innovating Mission Queue Constructing Method in Multi-Satellite Spatial Information Networks

Jun Long ^{1,2,3,4}, Shangpeng Wang ^{1,2} , Yakun Huo ^{1,2}, Limin Liu ^{1,2,4} and Huilong Fan ^{1,2,*} 

¹ School of Computer Science and Engineering, Central South University, Changsha 410075, China; junlong@csu.edu.cn (J.L.); 204701019@csu.edu.cn (S.W.); huokunkun5@gmail.com (Y.H.); liulimin@outlook.com (L.L.)

² Network Resource Management and Trust Evaluation Key Laboratory of Hunan, Changsha 410075, China

³ Big Data Institute, Central South University, Changsha 410075, China

⁴ Joint Laboratory of Space Information System, Changsha 410075, China

* Correspondence: hlfanpro@csu.edu.cn

Abstract: The purpose of constructing onboard observation mission queues is to improve the execution efficiency of onboard tasks and reduce energy consumption, representing a significant challenge in achieving efficient global military reconnaissance and target tracking. Existing research often focuses on the aspect of task scheduling, aiming at optimizing the efficiency of single-task execution, while neglecting the complex dependencies that might exist between multiple tasks and payloads. Moreover, traditional task scheduling schemes are no longer suitable for large-scale tasks. To effectively reduce the number of tasks within the network, we introduce a network aggregation graph model based on multiple satellites and tasks, and propose a task aggregation priority dynamic calculation algorithm based on graph computations. Subsequently, we present a dynamic merging-based method for multi-satellite, multi-task aggregation, a novel approach for constructing onboard mission queues that can dynamically optimize the task queue according to real-time task demands and resource status. Simulation experiments demonstrate that, compared to baseline algorithms, our proposed task aggregation method significantly reduces the task size by approximately 25% and effectively increases the utilization rate of onboard resources.

Keywords: Earth observation satellite; resource management; mission queue construction; graph computation; dynamic merging

MSC: 68Q25; 68W01; 68U03; 94-10



Citation: Long, J.; Wang, S.; Huo, Y.; Liu, L.; Fan, H. Dynamic Merging for Optimal Onboard Resource Utilization: Innovating Mission Queue Constructing Method in Multi-Satellite Spatial Information Networks. *Mathematics* **2024**, *12*, 986. <https://doi.org/10.3390/math12070986>

Academic Editors: Yufeng Yu, Guoxia Xu and Hu Zhu

Received: 3 March 2024

Revised: 23 March 2024

Accepted: 25 March 2024

Published: 26 March 2024



Copyright: © 2024 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

In the architecture of space–Earth integrated information networks, the efficiency limitations of conventional single satellites are insufficient to accommodate the escalating demand for observational data. Consequently, the significance of multi-satellite collaboration becomes paramount in enhancing task scheduling within space information networks [1–6]. Multi-satellite collaboration, with its extensive coverage, precise information collection capabilities, and the potential for ground-based imaging unfettered by geopolitical boundaries, is invaluable in domains such as battlefield reconnaissance, military target identification, resource exploration, disaster monitoring, urban planning, and crop assessment. The employment of mission aggregation technology, which consolidates various observation missions into a singular, unified monitoring effort, is of profound practical importance [7]. It not only reduces the aggregate number of missions but also diminishes the observational duration and attitude adjustment maneuvers of remote sensing equipment.

Despite notable advancements in multi-satellite collaborative technologies and task scheduling algorithms, the efficient allocation and utilization of satellite resources remain an

exigent challenge amid dynamic and multifaceted task environments. Most current mission aggregation algorithms concentrate on observation time windows and sidelobe angles, frequently overlooking the critical aspect of satellite resource utilization before and after aggregation [8]. In the network environment of spatio-temporal dynamic evolution, there are problems [9] such as large task scale, complex execution sequence relationship, dynamic queue growth, strict execution time, and inclusion of tasks. These characteristics make the calculation of the task execution queue particularly complicated. In addition, the traditional fixed task scheduling scheme will no longer adapt to a large-scale task and the dynamic changes of task requirements. The complex dynamic topology and complex constraint relationships of satellite information networks result in distributed scheduling architectures that are not suitable for on-satellite mission scheduling scenarios. The importance of in-depth research on task aggregation algorithms has become increasingly prominent.

At present, the task scheduling method based on spatial information network integration strategy is mainly applied to remote sensing observation tasks. This method interprets the issue as a clustering problem by constructing a task graph model based on clusters. The model is then solved using the appropriate approximation algorithm [10–13]. For example, Soma et al. [10], and improved cluster partitioning algorithm was designed to achieve the clustering of imaging reconnaissance tasks based on the approximate cluster partitioning algorithm proposed by Tseng C. J. combined with two merging criteria proposed by Lu-Feng Zhang [11]; Wang et al. [12] designed a clustering algorithm based on the maximum–minimum ant colony algorithm aiming to effectively accomplish the observation mission clustering of agile imaging satellites; in addition, more common are task aggregation methods based on task insertion. For example, Chen et al. [14] proposed an enhanced single-track optimal cluster partitioning clustering method for remote sensing satellite imaging missions; task aggregation methods based on task insertion are commonly employed. For example, Song et al. [15] introduces a Cluster-Based Genetic Algorithm (C-BGA) using the k-means clustering method to address satellite scheduling issues. By employing a heuristic population initialization strategy and a clustering-based evolutionary strategy, along with the application of a Task Arrangement Algorithm (TAA), the algorithm's adaptability to different scenarios and the likelihood of successful task scheduling have been effectively enhanced.

Compared with the traditional multi-satellite network mission planning methods that lack the intricate dependencies among multiple tasks and the allocation of onboard resources, the new generation of satellites with autonomous online mission planning capabilities can fully utilize the individual intelligence [16]. Under this precondition, our study aims to bridge these gaps by proposing an innovative multi-satellite, multi-task aggregation algorithm that not only addresses the dynamic scheduling of tasks but also optimizes resource utilization across the satellite network. This involves creating a strategy based on a multi-task aggregate graph model. By incorporating the principle of minimum energy consumption from graph theory, this strategy is designed to dynamically optimize the task queue. This optimization is based on current task demands and the status of available resources, ensuring a more effective allocation and utilization of satellite capabilities. While CP-TM stands as a solid foundation for maximizing observation opportunities through task merging criteria, it predominantly focuses on static task aggregation. It does not adequately address the dynamic scheduling of emergency tasks or the real-time optimization of task queues based on fluctuating demands and resource availability [17]. Our method extends these concepts by providing a comprehensive framework for dynamic task scheduling. By leveraging CP-TM as a baseline, we highlight the novel contributions of our approach in enhancing resource efficiency and operational effectiveness in satellite networks. As science and technology advance, the need for swiftly addressing complex space tasks escalates. Optimized task aggregation algorithms enhance the execution rate of observation tasks and enable rapid responses to terrestrial or extraterrestrial environmental changes, thereby significantly boosting the efficacy of space missions.

In this paper, we address the above shortcomings and investigate the graph-based task aggregation problem under multiple observation satellites to minimize satellite energy resources. The innovative work in this paper is as follows.

1. In-depth analysis of the principle of satellite Earth observation and the aggregation constraints of Earth observation missions, culminating in the creation of an advanced multi-satellite, multi-mission aggregation graph model and its corresponding construction algorithm;
2. Leveraging the aforementioned aggregation graph model, this paper devises a dynamic task merging-based multi-satellite, multi-task aggregation algorithm. The task synthesis priority based on the minimum energy consumption of the aggregated tasks and the minimum energy consumption calculation method of the aggregated tasks based on dynamic planning is designed to realize the fast solution of the task synthesis priority;
3. This paper's algorithm undergoes rigorous theoretical and experimental validation, including a comparative analysis with the aggregation algorithm presented in the literature [17]. The findings reveal that our algorithm can decrease mission size by approximately 45% and reduce the energy consumption needed for mission execution by about 50%. These results significantly enhance satellite operational efficiency and resource utilization.

2. Related Work

The construction of multi-satellite mission queues plays a crucial role in the complex scheduling relationships between rapidly growing, intricate user demands and limited on-board resources. In the current environment of multiple satellites and ground stations, the ability to reasonably allocate payload resources, improve resource utilization efficiency, and maximally meet the timeliness requirements of observation missions presents a valuable research point.

At present, most research on multi-satellite mission scheduling is based on the dynamic programming algorithm, greedy algorithm, and heuristic algorithm [18–21]. In recent years, the exploration of multi-satellite mission scheduling has made continuous progress. Wu et al. [20], by designing preprocessing models for different types of targets to decompose complex scenes, and using dynamic task planning algorithms to optimize the resource allocation for multi-satellite cooperative observation, effectively improved the efficiency of dynamic observation by remote sensing satellites. For the dynamic area task scheduling problem, Yin and Clerckx [18] proposed an emergency response satellite scheduling model, which, using dynamic segmentation algorithm and dynamic Greedy state algorithm, can intelligently assign satellite imaging tasks and has a good application prospect. Furthermore, Long et al. [19] proposed a semi-Markov decision process (SMDP) method to solve the satellite task scheduling problem, which reduces the operational complexity and the number of steps of task planning, and achieves higher acceleration in the single target scenario. For complex constraint conflicts, Feng et al. [21] proposes a large-scale task multi-satellite cooperative scheduling method based on a hybrid graph neural network (GNN) and meta-heuristic algorithms. By leveraging the GNN's capability to represent and extract task relationships, combined with the optimization framework of meta-heuristic algorithms, the computational efficiency of large-scale task planning has been substantially improved.

In addition to the classical algorithms and their variants, there exist search methods based on complex neural network architectures. For instance, in the paper by Liu et al. [22], a new satellite mission planning algorithm leveraging symmetric recurrent neural networks is introduced. This algorithm demonstrates the capability to promptly respond to various task requirements, aligning with business needs. On a related note, Han et al. [23] introduced EfficientNet, a novel CNN model scaling method. This approach optimizes model channels in terms of number, depth, and width through neural structure search and composite coefficients. While these approaches effectively enhance resource utilization,

satisfy user demands, and manage both urgent and routine task scheduling, they encounter challenges in adapting to the time-varying characteristics of satellite networks. However, their computational complexity often results in increased time consumption, limiting their applicability and efficiency in real-world scenarios.

In recent years, numerous scholars have delved into the intricate interrelationships among multiple tasks using graph theory, leading to advancements in both theory and practical applications in the field [24–28]. Hao et al. [24] introduces the Multi-Aspect Extended Hypergraph (MAEH) model, which accurately describes the spatiotemporal characteristics and functional attributes of resources. Through a two-stage scheme, they achieve rapid and low-complexity scheduling of cross-domain resources, thereby effectively enhancing resource utilization. To overcome the limitations of traditional Time Expanded Graphs (TEG) in representing observation, energy, and transceiver resources, Li et al. [25] proposes an Enhanced Time Expanded Graph (ETEG). By introducing snapshot graphs to depict the time-varying attributes of resources, along with virtual links and node transformations to manage resource constraints, simulation results have validated the efficiency and effectiveness of the ETEG method. In the realm of energy consumption minimization, Phillips et al. [27] proposes a new model for satellite formation control, highlighting the practicality and effectiveness of graph theory in addressing energy optimization problems in space missions. Their study provides valuable insights for future satellite network design. Shi et al. [28] explores the use of time-expanded graph models to tackle the challenge of energy-efficient delay-constrained multicast in satellite networks. By establishing a time-expanded graph theory model, they optimize energy consumption during data transmission while ensuring compliance with transmission delay constraints. This work not only improved the energy efficiency of satellite networks during large-scale data transmission but also offered an innovative perspective on utilizing graph theory methods to solve problems with real-time and energy dual constraints.

With the emergence of edge computing, Zhang et al. [29] has applied it to multi-satellite task scheduling, enabling better adaptation to specific scenarios and optimizing resource utilization and repeatability. Wang et al. [30] proposes a high dynamic cluster task scheduling algorithm for satellite edge computing, which solves the problems of traditional satellite task processing, such as too long data transmission time and time-sensitive tasks can not respond in time, and improves the utilization efficiency of satellite cluster meter resources and task processing speed. Despite these innovations, challenges persist in addressing the complex dynamics, temporal variations, and multi-resource coordination of satellite information networks.

Given the time-varying adaptability of spatial information network environment, a multi-star task scheduling algorithm that can better adapt to this characteristic is needed. Considering the constraints of airborne resource environment, scheduling complexity and computing time, dynamic real-time problems, and scalability of spatial information network environment, the task is optimized under multiple constraints. A strategy based on multi-task aggregate graph model is proposed, which is further explored by combining the principle of minimum energy consumption in graph theory.

3. System Model and Problem Description

3.1. System Model

In this paper, we adopt the classical Earth observation system model, as shown in Figure 1. The Earth observation satellite orbits the Earth at a particular inclination turns on the onboard remote sensor payload when it flies over the ground target, and performs the observation operation by pushing and sweeping the image. The satellite observation area corresponds to an observation strip on the Earth's surface parallel to the satellite's subsatellite trajectory, and the missions within the coverage of the observation strip are considered to be observed. The satellite performs the observation by turning the side-swing angle of the remote sensor in the direction of the perpendicular satellite orbit to deviate the observation strip from the satellite subsatellite trajectory to cover more missions.

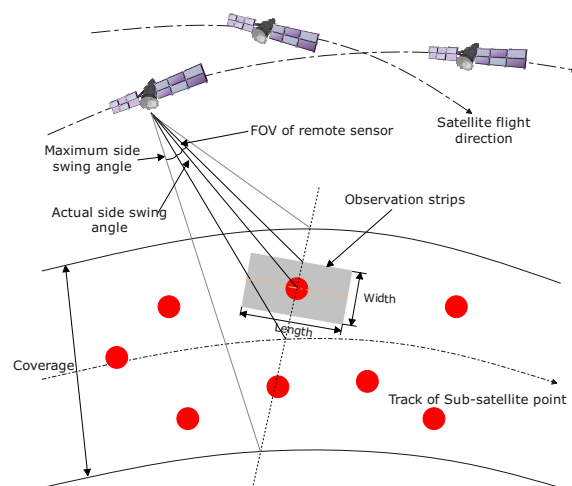


Figure 1. Satellite Earth observation schematic.

Let there be a total of M Earth observation satellites with observation missions in the system, and each satellite carries an imaging remote sensor. N missions to be observed are randomly distributed on the Earth's surface, and the missions are meta-missions that can be observed by a single transit of the satellite, and the following system model is established.

The set of meta-tasks $GT = \{T_1, T_2, \dots, T_N\}$, any meta-task in the set can be described by a triplet $t_i = (lon_i, lat_i, d_i)$, lon_i and lat_i denote the longitude and latitude where the meta-task is located, respectively, and d_i denotes the continuous execution time required for the meta-task T_i .

The set of satellite resources $GS = \{S_1, S_2, \dots, S_M\}$, any satellite resource in the set can be described by a six-tuple $S_j = \{o_j, \sigma_j, \omega_j, at_j, oe_j, se_j\}$, where o_j denotes the orbit information of the satellite, the rest is the information related to the remote sensor carried by the satellite, σ_j , ω_j , and at_j denotes the field of view, side-swing rate and maximum on-time of the remote sensor, respectively. oe_j denotes the energy consumption per unit time inland when the remote sensor performs the observation activity and se_j denotes the energy consumption per unit angle when the remote sensor performs a side-swing activity.

$GTW_{i,j}$ denotes the set of observation time windows of the satellite S_j for the mission T_i during the mission planning cycle, which is calculated from the orbit information of the satellite, the parameter information of the remote sensor carried by the satellite, and the geographical location of the mission target. Any observation time window $tw_{i,j,k} \in GTW_{i,j}$ in the set can be described by a triplet $tw_{i,j,k} = \{ts_{i,j,k}, te_{i,j,k}, \theta_{i,j,k}\}$, $ts_{i,j,k}$ and $te_{i,j,k}$ denotes the earliest start time and the latest end time of the observation performed by the satellite in the observation time window, $\theta_{i,j,k}$ denotes the ideal side-swing angle of the remote sensor when the satellite performs the observation under the observation time window $tw_{i,j,k}$, i.e., the angle that the satellite remote sensor side-swing needs to turn when the center point of the observation strip formed on the ground when the satellite performs the observation coincides with the position of the mission target.

3.2. Problem Description

Based on the above model, the multi-satellite multi-task aggregation problem in this paper can be described as follows: the meta task set GT , the satellite resource set GS , and the observation time window information $GTW_{i,j}$ of any meta task $T_i \in GT$ under the satellite resource $S_j \in GS$ are known, all the meta tasks in GT that satisfy the aggregation constraints with each other are aggregated together to form a new task-aggregation task, and the satellite resources corresponding to the aggregated task are determined as well as the start execution time and the end execution time. The final result of task aggregation is the aggregated task set $GC = \{C_1, C_2, \dots, C_L\}$, where L is the number of aggregated

missions. Any of the aggregated tasks $C_k = (M_k, l_k, st_k, et_k, \delta_k)$ in the set, where $M_k, l_k, st_k, et_k, \delta_k$ denote the set of meta-tasks corresponding to the aggregated task C_k , the satellite resources, the start execution time, the end execution time, and the side-swing angle of the remote sensor during execution, respectively.

In order to clarify the methodology and the satellite Earth observing model, Table 1 summarizes the essential symbols and their corresponding descriptions.

Table 1. Symbol descriptions for satellite operations.

Symbol	Description
GT	Meta-task set $\{T_1, \dots, T_N\}$
d_i	The continuous execution time required for meta-task T_i
GS	Satellite resource set $\{S_1, \dots, S_M\}$
S_j	The six-tuple of satellite resource S_j ($o_j, \sigma_j, \omega_j, at_j, oe_j, se_j$)
o_j	Orbital information for satellite S_j
σ_j	FOV angle of satellite resource S_j
ω_j	Side-wing rate of remote sensor on satellite S_j
at_j	Maximum continuous working time of remote sensor on satellite S_j
oe_j	Energy consumption rate of remote sensor on satellite S_j when observing
se_j	Energy consumption per unit angle of remote sensor on satellite S_j
$GTW_{i,j}$	Observation time windows of meta-task T_i on satellite S_j
$tw_{i,j,k}$	Observation time window tuple ($ts_{i,j,k}, te_{i,j,k}, \theta_{i,j,k}$)
$ts_{i,j,k}$	The earliest observation time of task T_i under window k using satellite S_j
$te_{i,j,k}$	The latest observation time of task T_i under window k using satellite S_j
$\theta_{i,j,k}$	Ideal sensor swing angle for mission T_i under window k using satellite S_j
GC	Aggregated task set $\{C_1, \dots, C_L\}$
C_k	Aggregated task elements ($M_k, l_k, st_k, et_k, \delta_k$)
M_k	The set of meta-tasks contained in task C_k
l_k	Resources required to execute aggregation task C_k
st_k, et_k	Start and end times of task C_k
δ_k	Side-swing angle of the remote sensor during execution aggregation task C_k

The purpose of mission aggregation in this paper is to reduce the energy consumption on the satellite star. Therefore, the aggregation process of the mission mainly considers the energy consumption required by the satellite to perform the task before and after the aggregation, and the final aggregated mission requires the minor energy consumption as a whole. In the actual process of satellite execution of observation tasks, satellite energy consumption mainly comes from the remote sensor to perform observation activities and side pendulum activities, each observation requires two side pendulums (open and close), then the objective function of mission aggregation as shown in Equation (1).

$$\min E = \sum_{k=1}^{|F|} (et_k - st_k) \times oe_j + 2 \times \frac{|\delta_k|}{\omega_j} \times se_j \quad (1)$$

The constraints mainly include satellite resource constraints and observation time window constraints. Assuming that k meta-tasks $GT_p = \{T_{p1}, T_{p2}, \dots, T_{pk}\}$ can be aggregated to obtain aggregated task C_p , the satellite resource constraint between the meta-tasks is shown in Equation (2).

$$\exists S_j \in GS : \forall T_{pi} \in GT_p, GTW_{pi,j} \neq \emptyset \quad (2)$$

The satellite resource constraint indicates the existence of at least one satellite resource S_j capable of performing all the meta-tasks in the aggregated mission. The observation time window between meta-missions is shown in Equation (3).

$$\forall T_{pq} \in GT_p, \exists S_j \in GS, tw_{pq,j,k} \in GTW_{pq,j} \quad (3)$$

s.t.

$$\begin{cases} (a) \max_{T_{pq} \in T_p} \{\theta_{pq,j,k_{pq}}\} - \min_{T_{pq} \in T_p} \{\theta_{pq,j,k_{pq}}\} \leq \sigma_j, \\ (b) \max_{T_{pq} \in T_p} \{te_{pq,j,k_{pq}}\} - \min_{T_{pq} \in T_p} \{ts_{pq,j,k_{pq}}\} \leq at_j, \end{cases} \quad (4)$$

The observation time window constraint indicates that the satellite can perform all meta-tasks in the aggregated mission in a single observation campaign. The observation time window constraint consists of a side-swing angle constraint and a time constraint. (a) in Equation (4) is the side-swing angle constraint, which indicates that the satellite can adjust the side-swing angle of the remote sensor to cover all the tasks in the aggregated mission by the remote sensor field of view when executing the observation activity. (b) in Equation (4) is a time window constraint, which indicates that the satellite can complete the observation activities of all meta-tasks in the aggregation task within the maximum power-on time of the remote sensor, i.e., the observation time length of the aggregation task is less than the maximum power-on time of the remote sensor. Since the observation time interval of the aggregated tasks is dynamically changing during the task aggregation process, the difference between the earliest time and the latest time of the observation time window of all meta-tasks in the aggregated tasks is chosen here as the judgment criterion to ensure that the execution time of the aggregated tasks must be less than the maximum power-on time of the remote sensor.

4. Algorithm Design

The multi-satellite multi-task aggregation problem is a combinatorial optimization problem under multiple constraints, and the solution process is complex and computationally intensive. In this paper, we first construct a Multi-satellite and Multi-task Aggregation Graph (MSMTAG) based on the task aggregation condition better to describe the task-to-task aggregation relationship under multi-satellite resources and design a task aggregation based on dynamic merge priority (DMP-TA) to solve the task aggregation problem based on the MSMTAG model.

4.1. MSMTAG Model

The MSMTAG model is denoted by $G = (V, E)$, where V is the set of vertices and E is the set of edges between vertices. Each vertex $v_p \in V$ represents an aggregation task. An aggregation task can contain only one meta-task, i.e., each meta-task itself can be viewed as an aggregation task containing only itself. For any two vertices v_p and v_q in Graph G , an edge connection relationship exists between vertex v_p and vertex v_q if the meta-task contained in v_p and the meta-task contained in v_q satisfy the satellite resource constraint shown in Equation (2) and the observation time window constraint shown in Equation (3) with each other. There may be multiple edge connection relationships between vertex v_p and vertex v_q , each of which corresponds to one aggregation of the meta-tasks contained in v_p and v_q , as shown in Figure 2.

All edges between vertex v_p and v_q form a set $E_{p,q}$, and the more elements there are in the set $E_{p,q}$, the more aggregation options are available between vertex v_p and vertex v_q . Figure 3 depicts the aggregation graph G containing five vertices and seven edges.

Vertex v_1 indicates an aggregation task containing meta-tasks T_1 and T_2 ; vertex v_2 indicates an aggregation task containing meta-tasks T_3 and T_4 ; vertex v_3 indicates an aggregation task having only meta-task T_5 ; vertex v_4 indicates an aggregation task containing meta-tasks T_6 and T_7 ; vertex v_5 indicates an aggregation task containing only meta-task T_8 ; and edge $E_{1,2}$ indicates that vertices v_1 and v_2 are aggregated in one way under satellite resource S_k ; edge $E_{1,3}$ indicates that vertices v_1 and v_3 are aggregated in one way under satellite resource S_j ; edge $E_{1,5}$ indicates that vertices v_1 and v_5 are aggregated in two ways under satellite resource S_j and in one way under satellite resource S_k ; edge $E_{2,5}$ indicates that vertices v_1 and v_5 are aggregated in three ways under satellite resource S_j ; edge $E_{3,4}$ indicates that vertices v_3 and v_4 are aggregated in two ways under satellite resource S_k ;

edge $E_{3,5}$ indicates that vertices v_3 and v_5 exist in one aggregation mode under satellite resource S_k ; edge $E_{4,5}$ indicates that vertices v_4 and v_5 exist in one aggregation method under satellite resource S_j .

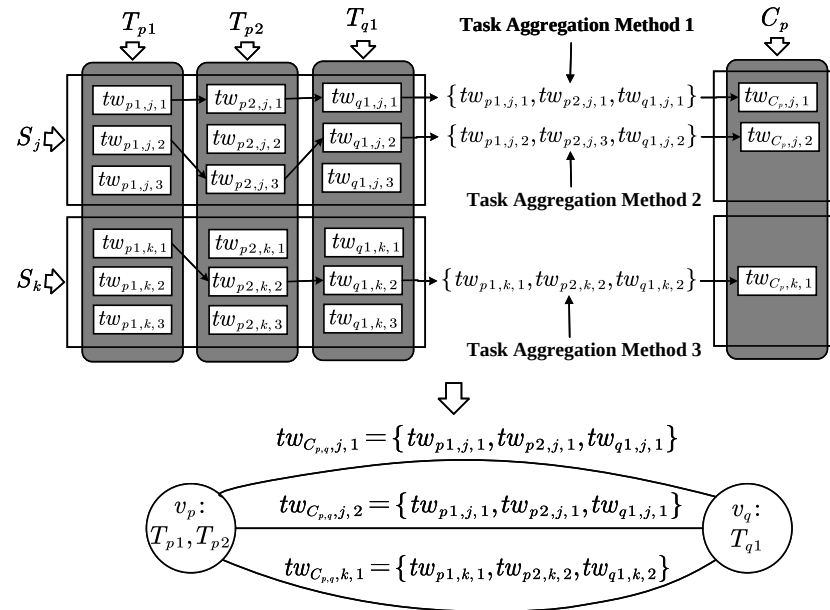


Figure 2. Vertex and aggregation method schematic.

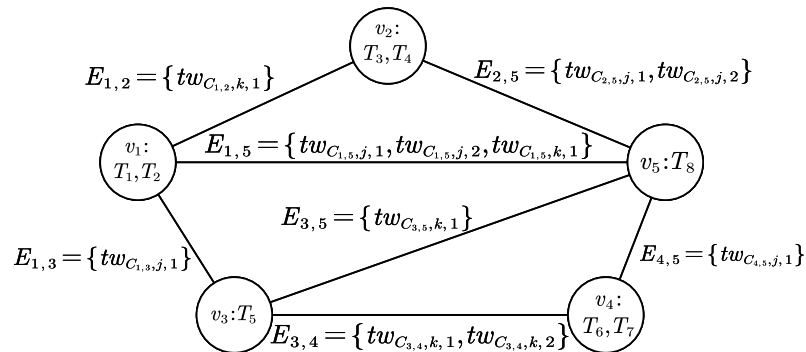


Figure 3. Multi-satellite and multi-task aggregation graph model.

The initial MSMTAG model contains only one meta-task per vertex, and the edges between the vertices are constructed by judging whether the time windows between the vertices' corresponding meta-tasks satisfy the task aggregation constraints shown in Equations (2)–(4), and the specific construction algorithm is shown in Algorithm 1.

4.2. DMP-TA Algorithm

The DMP-TA algorithm solves the task aggregation problem iteratively based on the initial MSMTAG model. The idea of the algorithm is to set a merge priority for each edge in the MSMTAG model and merge the two vertices connected to the edge with the maximum merge priority in each round until no edge exists in the graph.

The calculation of the merge priority at the core of the DMP-TA algorithm. Considering that task aggregation aims to reduce the consumption of satellite energy resources, the algorithm takes the value of the energy that can be consumed less by executing a task after aggregation than by performing a task without aggregation as the basis for the merge priority. The merger priority is defined as follows.

Algorithm 1: Initial MSMTAG model construction algorithm

```

1 Define  $f : GT \rightarrow V : \forall GT, \exists! v_i \in V : f(GT) = v_i$ ;
2 Initialize  $p = 1, q = 2, k = 1$ ;
3 for  $\forall tw_{p,j,kp} \in GTW_{p,j}$  and  $\forall tw_{q,j,kq} \in GTW_{q,j}$  do
4   if  $(\exists S_j \in GS : \forall T_{pi} \in GT_p, GTW_{pi,j} \neq \emptyset)$  and
       $(\forall T_{pq} \in GT_p, \exists S_j \in GS, tw_{pq,j,k} \in GTW_{pq,j})$  then
5      $E_{p,q} = \{tw_{p,j,kp}, tw_{q,j,kq}\}$ ;
6      $E \leftarrow E \cup E_{p,q}$ ;
7      $k = k + 1$ ;
8   while  $p < |GT|$  do
9     if  $q \leq |GT|$  then
10        $q = q + 1$ ;
11     else
12        $p = p + 1$ ;
13        $q = p + 1$ ;
14 return  $G$ 

```

Definition 1. Aggregation Priority.

$$AP(E_{i,j}) = f_{\min}(v_i) + f_{\min}(v_j) - f_{\min}(v_{C_{i,j}}) \quad (5)$$

where $f_{\min}(v_i)$, $f_{\min}(v_j)$ denote the minimum value of energy required for the execution of the aggregation tasks corresponding to vertex v_i and vertex v_j , respectively, and $f_{\min}(v_{C_{i,j}})$ denotes the minimum value of energy needed for the execution of the new aggregation tasks formed by the aggregation of tasks corresponding to vertex v_i and vertex v_j . Since the MSMTAG model G changes with each task merge performed by the algorithm, the merge priority between tasks is also dynamically changing. The algorithm will dynamically calculate and update the merge priority of the edges during each round of iteration.

The essence of calculating merge priority lies in determining the minimal amount of energy required for the execution of aggregation tasks. As shown in Figure 2, for any edge $E_{p,q}$, the more elements contained in $E_{p,q}$ of the aggregation graph, the more aggregation methods are available between vertices v_p and v_q . Although the aggregation tasks generated under different aggregation methods include the same meta-tasks, the corresponding satellite resources and execution times may be different. Thus the energy consumption required for the execution of the aggregated tasks obtained after aggregation is also different. Therefore, solving the minimum energy consumption required for executing the aggregated task requires calculating the minimum energy consumption value for generating this aggregated task under different aggregation methods.

This paper designs a dynamic programming-based computational method to quickly solve the minimum energy consumption of an aggregation task generated under a certain aggregation method. Let one aggregation method of the aggregation task $C_p = \{T_{p1}, T_{p2}, \dots, T_{pl}\}$ be $\{tw_{p1,j,k1}, tw_{p2,j,k2}, \dots, tw_{pl,j,kl}\}$, the algorithm for solving C_p the minimum value of energy consumption ME_p is shown in Algorithm 2.

In summary, the critical steps of the DMP-TA algorithm can be described as follows: starting from the initial task aggregation graph G constructed in Algorithm 1, the edge $E_{p,q}$ with the maximum merge priority is selected in each round, and the two vertices v_p and v_q connected to this edge are merged to form a new aggregation task $v_{C_{pq}}$, and the minimum energy required to execute the aggregation task $v_{C_{pq}}$ is calculated and updated according to Algorithm 2. Next, the edges of vertices v_p and v_q and the edges connected to vertices v_p and v_q are deleted, and the edges of vertex $v_{C_{pq}}$ and the edges connected to vertex $v_{C_{pq}}$ are added, and the merging priority of the newly added edges is calculated, and

the next round of computation is performed. The algorithm ends when the set of edges E in the aggregation graph G is empty, and the DMP-TA algorithm is described in Algorithm 3 as follows.

Algorithm 2: Dynamic planning-based algorithm for minimum energy calculation for aggregated tasks

```

1 Sort  $GTW$  by end time to get  $TWL = \{tw_{x1,j,k1}, \dots\}$  for  $TL = \{T_{x1}, \dots\}$ . Initialize  $b_i, f_i$  for min energy start and end times for the first  $i$  tasks.
2 if  $(te_{x1} - \max\{ts_{x1}, ts_{x2}\}) \geq \max\{d_{x1}, d_{x2}\}$  then
3    $f_2 = te_{x1}; b_2 = te_{x1} - \max\{d_{x1}, d_{x2}\};$ 
4 else
5    $f_2 = \max\{te_{x1}, d_{x2} + ts_{x2}\}; b_2 = \min\{te_{x1} - d_{x1}, ts_{x2}\};$ 
6 for  $i = 3$  to  $l$  do
7   if  $(\min\{f_{i-1}, te_{xi}\} - \max\{b_{i-1}, ts_{xi}\}) \geq d_{xi}$  then
8      $f_i = f_{i-1}; b_i = b_{i-1};$ 
9   else
10     $f_i = ts_{xi} + d_{xi}; b_i = b_{i-1};$ 
11  $\delta_{C_p} = \frac{\sum \theta_{xn,j,kn}}{l};$ 
12  $ME_p = (f_l - b_l) \times oe_j + 2 \frac{|\delta_{C_p}|}{\omega_j} \times se_j.$ 

```

The time complexity of the DMP-TA algorithm is $O(n^3)$, where n is the number of meta-tasks.

Algorithm 3: DMP-TA Algorithm

```

1 Initialize MSMTAG Model
2 foreach  $v \in V$  do
3   Compute  $E_{\min}(v);$  // Minimum energy for executing the task at  $v$ 
4    $AP(E_{ij}) = f_{\min}(v_i) + f_{\min}(v_j) - f_{\min}(v_{C_{ij}});$ 
5 while  $E \neq \emptyset$  do
6    $E_{\max} = \arg \max AP(E_{ij});$  // Select the edge with the highest AP.
7    $E \leftarrow (E \setminus E_{ij}) \cup \{E_{\max}\};$ 
8    $v_p = \text{Merge}(v_i, v_j);$  // Merge vertices to a new vertex  $v_p$ .
9   Recompute  $E_{\min}(v_p);$ 
10  foreach  $v_k \in V, v_k \rightarrow v_p$  do
11    if Conditions of Equation (2) and Equation (3) are satisfied then
12       $E \leftarrow E \cup E_{k,p};$  // Update edges
13      Update  $AP(E_{k,p});$  // Recalculate AP for new edges
14 return  $G$ 

```

Proof. The time complexity of constructing the initial MSMTAG model is $O(n^2)$ (Line 1). The time complexity of computing the minimum execution energy of the corresponding aggregated tasks for all vertices in the initial MSMTAG model G is $O(n^2)$ (Lines 2–3). The task aggregation graph with n vertices has at most C_n^2 edges, and it takes $\frac{1}{2} \times n \times (n-1)$ calculations to compute the maximum merge priority of all edges, in that the time complexity of computing the edge with the maximum merge priority in the algorithm is $O(n^2)$ (Line 4), and similarly, the time complexity of selecting the edge with the maximum merge priority is $O(n^2)$ (Line 6). The time complexity of deleting edges is $O(n)$ (Line 7). Calculating the vertex contains at most n meta-tasks, the time complexity of computing the minimum energy

consumption of the aggregation task corresponding to the vertex is $O(n)$, and the time complexity of merging a vertex is $O(n)$ (Line 8). After creating a new vertex, it is necessary to regenerate the edges connected to it and calculate the merging priority of the edges, which require a time complexity of $O(n)$ (Line 12) and $O(n^2)$ (Line 13), respectively. Among them, Lines 6–13 are the operations in the loop body, and the number of tasks is reduced by one for each task synthesis in the algorithm, and the loop is terminated at most $n - 1$ times, so the time complexity of the loop body is $O(n^3)$. In summary, the overall time complexity of the algorithm is $O(n) + O(n^2) + O(n^3)$, i.e., $O(n^3)$. $\square$

5. Results

In this paper, we test the performance of the DMP-TA algorithm through extensive simulation experiments and compare it with the CP-TM algorithm proposed in the literature [17]. The models and algorithms used in the experiments are implemented on an Intel(R) Core(TM) i7-9750H 2.59 GHz CPU and a computer with 8 GB RAM. The selected comparison algorithm, the CP-TM algorithm, is also a time complexity of $O(n^3)$. This algorithm only considers aggregation between tasks that have an ordinary time window under the same satellite resource and the ordinary time window is larger than the required execution time of the task in the process of aggregation and prioritizes the merging of the two aggregated tasks with the most common time windows in each aggregation process.

5.1. Test Methods and Parameters

Three Earth observation satellites and six sets of Earth observation missions are set up in the experiment. The orbital parameters of the Earth observation satellites and the onboard remote sensors are referred to the famous foreign IKONOS-2 satellite, SPOT-5 satellite, and the domestic Gaofen series satellites. Table 2 shows the information of the satellite orbit parameters and the parameters of the remote sensors carried by them. The target sizes of the six groups of Earth observation missions are 100, 200, 300, 400, 500, and 600, and all the missions are distributed in the latitude $[-30^\circ, 60^\circ]$ and longitude $[0^\circ, 150^\circ]$ ranges. Three distribution methods are used: dense distribution, average distribution, and mixed distribution (i.e., half are uniformly distributed and half are densely distributed). The time window of the satellite to the mission was calculated using Satellite Tool Kit(STK) version 11.6 x64 software.

Table 2. Satellite orbit parameters.

Satellite Parameters	Satellite S_1	Satellite S_2	Satellite S_3
Semi-major axis (km)	7201	7052	7480
Orbital eccentricity	0.0134947	0.0008895	0.0015076
Orbit inclination ($^\circ$)	98.2597	98.15	98.0036
True anomaly ($^\circ$)	191.1477	246.0537	7480
Ascending node equinox ($^\circ$)	7.0557	73.3937	7480
Argument of periapsis ($^\circ$)	169.2664	114.1612	7480
Field of view ($^\circ$)	2.1	0.931	7480
Lateral swing rate/(s)	1	1	7480
Maximum swing angle ($^\circ$)	± 45	± 45	± 45
Maximum operating time (s)	80	80	80
Observation of energy consumption (km/s)	200	200	200
Side pendulum energy consumption (km/ $^\circ$)	500	500	500

5.2. Algorithm Performance

In this section, the effectiveness of the algorithm is first verified. In the above experimental scenario, six sets of experiments are set up, each with task sizes of 100, 200, 300, 400, 500, and 600, and the tasks are distributed in a mixed distribution. In the experiments, the running time of the DMP-TA algorithm, the number of aggregated tasks obtained after aggregation, and the sum of the energy required to perform the aggregated tasks after aggregation is verified. The sum of the energy consumption for the execution of the

aggregated tasks solved by the algorithm is compared with the sum of the energy consumption for the performance of the tasks without aggregation and the sum of the energy consumption for the execution of the aggregated tasks solved by the CP-TM algorithm. To improve the reliability of the results, the two algorithms were run 15 times, and the average of the relevant results was taken. The obtained experimental results are shown in Figures 4–6.

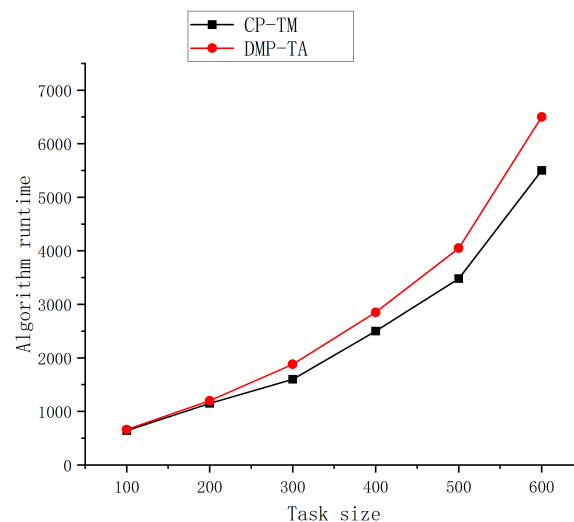


Figure 4. Comparison of algorithm running time.

Figure 4 indicates that the execution time of the DMP-TA and CP-TM algorithms increases with the increase of task size. The execution time of the DMP-TA algorithm is slightly higher than that of the CP-TM algorithm for the same task size, and the difference in the execution time of the algorithms increases with the increase of the task size. This phenomenon is because the merging priority used by the DMP-TA algorithm in considering task merging needs to calculate the minimum energy consumption of the aggregated tasks, and this part of the calculation consumes some time. However, from the proof in Section 4.2, the time complexity of the DMP-TA algorithm is the same as that of the CP-TM algorithm, so it can be theoretically speculated that the difference in execution time between the DMP-TA algorithm and the CP-TM algorithm will not be significant regardless of the increase in task size.

Figure 5 shows that both the DMP-TA and CP-TM algorithms have relatively significant aggregation effects. The task size can be reduced by about 25% after aggregation using the CP-TM algorithm. The task size can be reduced by about 50% after aggregation using the DMP-TA algorithm, about 25% higher than that of the CP-TM algorithm. Overall, the aggregation capability of the DMP-TA algorithm is significantly better than that of the CP-TM algorithm because the DMP-TA algorithm takes into account the extended start-up time of the remote sensor in the process of aggregation. Tasks with observation time windows at the maximum start-up time of the remote sensor have the opportunity to be aggregated. In contrast, the CP-TM algorithm only considers tasks with common time windows, and the length of the common time window is more significant than that required for task execution. The aggregation condition of the CP-TM algorithm is more severe than that of the DMP-TA algorithm.

Figure 6 shows that both the DMP-TA algorithm and the CP-TM algorithm effectively reduce the energy consumption required for task execution. The aggregation using the CP-TM algorithm reduces the time required for task execution by about 20%, while the aggregation using the DMP-TA algorithm reduces the time required for task execution by about 50%, which is about 30% higher than that of the CP-TM algorithm. The reason is that the algorithm calculates the overall energy reduction before and after each edge aggregation and prioritizes the merging of the two clustering tasks corresponding to the edges with the most considerable energy reduction. In contrast, the CP-TM algorithm considers the

number of remaining observation opportunities in each round of task merging. In addition, the aggregation condition of the CP-TM algorithm is more stringent than that of the DMP-TA algorithm, such that the number of tasks that are not aggregated after aggregation using the CP-TM algorithm will be greater compared to that using the DMP-TA algorithm, resulting in the CP-TM algorithm being inferior to the DMP-TA algorithm in terms of reducing the sum of energy consumption required for task execution.

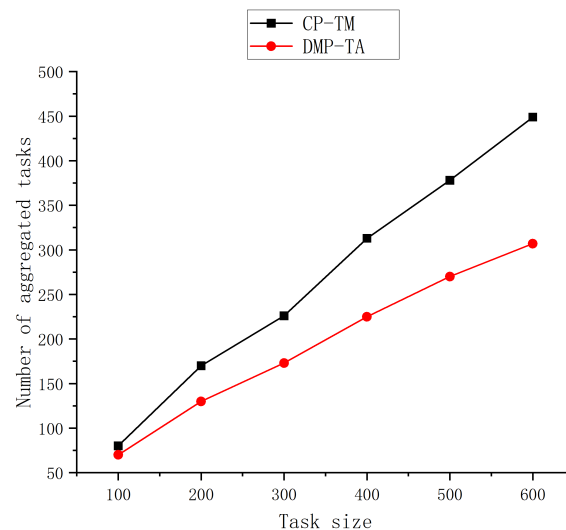


Figure 5. Comparison of the number of aggregation tasks.

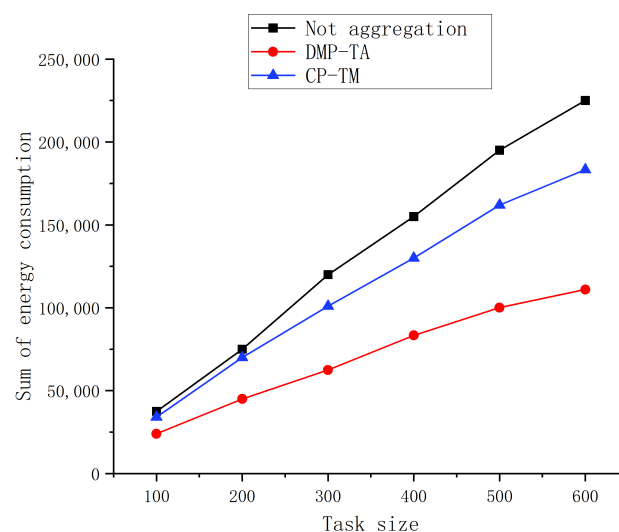


Figure 6. Comparison of the sum of energy requirements for aggregation task execution.

5.3. Algorithm Performance

Considering the possible large influence of the distribution density of tasks on the algorithm, this paper generates 18 sets of observation tasks in the range of latitude $[-30^\circ, 60^\circ]$ and longitude $[0^\circ, 150^\circ]$, respectively (groups 1–6 are observation tasks with task sizes of 100, 200, 300, 400, 500, 600 under dense distribution, groups 7–12 are observation tasks with task sizes of 100, 200, 300, 400, 500, 600 under the uniform distribution, respectively 100, 200, 300, 400, 500, 600, and 13–18 groups of 100, 200, 300, 400, 500, 600 under mixed distribution), and the DMP-TA algorithm is experimentally validated under these 18 groups of data. The experimentally obtained results are shown in Figures 7 and 8.

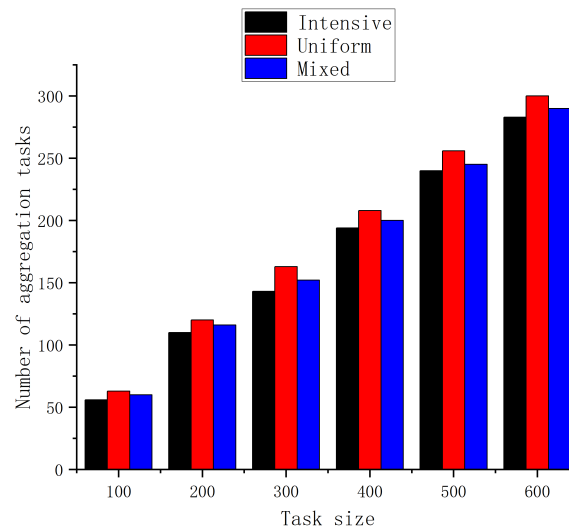


Figure 7. Comparison of the number of aggregated tasks under different distribution densities.

From Figure 7, it can be seen that the DMP-TA algorithm has a strong task aggregation capability in all three distributions: dense, uniform, and mixed distributions. Among them, the DMP-TA algorithm obtains the lowest number of aggregated missions when they are densely distributed and the highest number of aggregated missions when they are uniformly distributed. When the tasks are densely distributed, the number of tasks that can be covered simultaneously by a single satellite transit is higher. Thus the chance of task aggregation is greater.

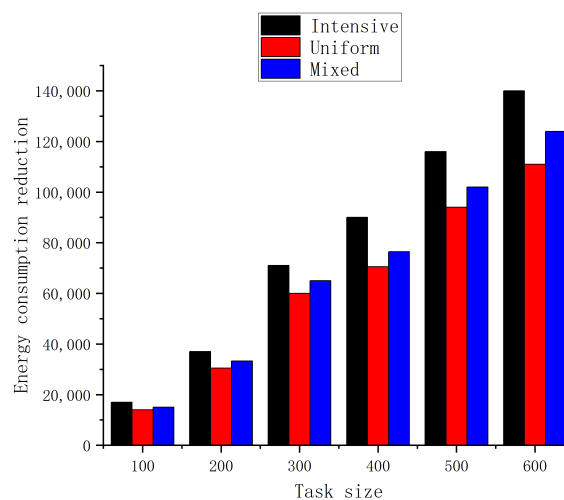


Figure 8. Comparison of reduced task execution energy consumption at different distribution densities.

From Figure 8, it can be seen that when the missions are densely distributed, the satellite resource consumption can be reduced the most by using the DMP-TA algorithm for aggregation, and the satellite resource consumption can be reduced the least by using the DMP-TA algorithm for aggregation when the missions are uniformly distributed. This phenomenon may be due to the higher overlap rate of observation time windows between missions when missions are densely distributed and the lower energy consumption of aggregated missions resulting from mission aggregation when the overlap rate of mission observation time windows is higher. Also, fewer task aggregations are generated in the case of dense distribution, which leads to the minor overall energy consumption of the observation tasks.

6. Conclusions

This paper addresses the problem of limited resources of Earth observation satellites in space information networks and the difficulty of satisfying many observation tasks. The aggregation method of observation tasks under multi-satellite cooperative observation is studied to improve the utilization rate of satellite resources. An in-depth analysis of the principle of satellite Earth observation and the aggregation constraints of Earth observation tasks is presented, a multi-satellite multi-task aggregation graph model is constructed, and a model construction algorithm is given. Based on the multi-satellite multi-task aggregation graph model, a multi-satellite multi-task aggregation algorithm is designed based on the dynamic merging of tasks. The task merging strategy in the aggregation algorithm is studied in depth. The task synthesis priority based on the minimum energy consumption of the aggregated tasks and the minimum energy consumption calculation method of the aggregated tasks based on dynamic planning is designed to realize the fast solution of the task synthesis priority. The paper concludes with a complete theoretical and experimental validation of the algorithm and a comparative analysis with the aggregation algorithm proposed in the literature [17]. The experiments show that the task size can be reduced by about 45%. The energy required for task execution can be reduced by about 50% after aggregation by the algorithm in this paper. The algorithm in this paper is slightly higher than the aggregation algorithm proposed in [17] in terms of running time. Still, the reduction in aggregated task size obtained using the algorithm in this paper is 25% more than using the algorithm in [17]. The sum of the energy consumption required for aggregated task execution is 20% less.

The observation task aggregation method proposed in this paper has some practical significance. However, there is room for improvement. To better match the work of observation satellites, the next step is to study the real-time task aggregation scheme during the execution of observation tasks.

Author Contributions: Conceptualization, J.L.; methodology, S.W. and L.L.; software, S.W.; validation, Y.H., J.L. and L.L.; formal analysis, S.W. and Y.H.; investigation, H.F. and J.L.; resources, S.W. and L.L.; data curation, S.W.; writing—original draft preparation, S.W.; writing—review and editing, H.F. and J.L.; visualization, S.W.; supervision, L.L.; project administration, J.L.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant U2003208, in part by the Science and Technology Plan of Hunan under Grant No. 2016TP1003, in part by the Key Technology R&D Program of Hunan Province under Grant No. 2018GK2052, in part by the Foundation of Preview of Equipment (No. 6141A020227), in part by the Theory and Method of Multi-Satellite Collaborative Application (No. 2019-JCJQ-ZD-342-00) and in part by the National Natural Science Foundation of China (No. 61972398).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Fan, H.; Yang, Z.; Zhang, X.; Wu, S.; Long, J.; Liu, L. A novel multi-satellite and multi-task scheduling method based on task network graph aggregation. *Expert Syst. Appl.* **2022**, *205*, 117565. [\[CrossRef\]](#)
2. Zhao, M.; Ye, N.; Ouyang, Q.; Jin, Y.; Jin, Y.; Zhao, L. Multi-satellite cooperative communication: Exploiting time asynchrony in non-orthogonal transmissions. *IEEE Trans. Veh. Technol.* **2023**, *72*, 6868–6873. [\[CrossRef\]](#)
3. Li, F.; Wan, Q.; Wen, F.; Zou, Y.; He, Q.; Li, D.; Zhong, X. Multi-Satellite Imaging Task Planning for Large Regional Coverage: A Heuristic Algorithm Based on Triple Grids Method. *Remote Sens.* **2024**, *16*, 194. [\[CrossRef\]](#)
4. Wen, Z.; Li, L.; Song, J.; Zhang, S.; Hu, H. Scheduling single-satellite observation and transmission tasks by using hybrid Actor-Critic reinforcement learning. *Adv. Space Res.* **2023**, *71*, 3883–3896. [\[CrossRef\]](#)
5. Zhang, X.; Sun, S.; Tao, M.; Huang, Q.; Tang, X. Multi-Satellite Cooperative Networks: Joint Hybrid Beamforming and User Scheduling Design. *IEEE Trans. Wirel. Commun.* **2024**, early access. [\[CrossRef\]](#)
6. Li, X.; Yang, J.; Fan, H. Dynamic Network Resource Autonomy Management and Task Scheduling Method. *Mathematics* **2023**, *11*, 1232. [\[CrossRef\]](#)

7. Chen, M.; Wen, J.; Song, Y.J.; Xing, L.; Chen, Y. A population perturbation and elimination strategy based genetic algorithm for multi-satellite TT&C scheduling problem. *Swarm Evol. Comput.* **2021**, *65*, 100912.
8. Qi, W.; Yang, W.; Xing, L.; Yao, F. Modeling and Solving for Multi-Satellite Cooperative Task Allocation Problem Based on Genetic Programming Method. *Mathematics* **2022**, *10*, 3608. [\[CrossRef\]](#)
9. Chen, Y.; Song, Y.; Du, Y.; Wang, M.; Zong, R.; Gong, C. A knowledge-based scheduling method for multi-satellite range system. In *Proceedings of the International Conference on Knowledge Science, Engineering and Management*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 388–396.
10. Soma, P.; Venkateswarlu, S.; Santhalakshmi, S.; Bagchi, T.; Kumar, S. Multi-satellite scheduling using genetic algorithms. In *Proceedings of the Space OPS 2004 Conference*, Montreal, QC, Canada, 17–21 May 2004; p. 515.
11. Cui, K.; Song, J.; Zhang, L.; Tao, Y.; Liu, W.; Shi, D. Event-Triggered Deep Reinforcement Learning for Dynamic Task Scheduling in Multi-Satellite Resource Allocation. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *59*, 3766–3777. [\[CrossRef\]](#)
12. Wang, Z.; Hu, X.; Ma, H.; Xia, W. Learning multi-satellite scheduling policy with heterogeneous graph neural network. *Adv. Space Res.* **2024**, *73*, 2921–2938. [\[CrossRef\]](#)
13. Yin, X.; Bai, M.; Li, Z. Clustering-Scheduling Methods for Oversubscribed Short-Term Tasks of Astronomical Satellites. *Trans. Nanjing Univ. Aeronaut. Astronaut.* **2023**, *40*, 307.
14. Chen, X.; Reinelt, G.; Dai, G.; Wang, M. Priority-based and conflict-avoidance heuristics for multi-satellite scheduling. *Appl. Soft Comput.* **2018**, *69*, 177–191. [\[CrossRef\]](#)
15. Song, Y.; Ou, J.; Wu, J.; Wu, Y.; Xing, L.; Chen, Y. A cluster-based genetic optimization method for satellite range scheduling system. *Swarm Evol. Comput.* **2023**, *79*, 101316. [\[CrossRef\]](#)
16. Song, Y.J.; Ma, X.; Li, X.J.; Xing, L.N.; Wang, P. Learning-guided nondominated sorting genetic algorithm II for multi-objective satellite range scheduling problem. *Swarm Evol. Comput.* **2019**, *49*, 194–205. [\[CrossRef\]](#)
17. Wang, J.; Zhu, X.; Qiu, D.; Yang, L.T. Dynamic scheduling for emergency tasks on distributed imaging satellites with task merging. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *25*, 2275–2285. [\[CrossRef\]](#)
18. Yin, L.; Clerckx, B. Rate-splitting multiple access for satellite-terrestrial integrated networks: Benefits of coordination and cooperation. *IEEE Trans. Wirel. Commun.* **2022**, *22*, 317–332. [\[CrossRef\]](#)
19. Long, X.; Wu, S.; Wu, X.; Huang, Y.; Mu, Z. A GA-SA Hybrid Planning Algorithm Combined with Improved Clustering for LEO Observation Satellite Missions. *Algorithms* **2019**, *12*, 231. [\[CrossRef\]](#)
20. Wu, Q.; Pan, J.; Wang, M. Dynamic Task Planning Method for Multi-Source Remote Sensing Satellite Cooperative Observation in Complex Scenarios. *Remote Sens.* **2024**, *16*, 657. [\[CrossRef\]](#)
21. Feng, X.; Li, Y.; Xu, M. Multi-satellite cooperative scheduling method for large-scale tasks based on hybrid graph neural network and metaheuristic algorithm. *Adv. Eng. Inform.* **2024**, *60*, 102362. [\[CrossRef\]](#)
22. Liu, W.; Li, Z.; Chen, L.; Zhang, D.; Shao, X. Design and simulation in SAR satellites' task planning system using genetic algorithm with entropy operator. *Aircr. Eng. Aerosp. Technol.* **2021**, *93*, 1236–1242. [\[CrossRef\]](#)
23. Han, J.; Wang, H.; Wu, S.; Wei, J.; Yan, L. Task Scheduling of High Dynamic Edge Cluster in Satellite Edge Computing. In *Proceedings of the 2020 IEEE World Congress on Services (Services)*, Beijing, China, 18–23 October 2020; IEEE: New York, NY, USA, 2020; pp. 288–294. [\[CrossRef\]](#)
24. Hao, Q.; Sheng, M.; Zhou, D.; Shi, Y. A multi-aspect expanded hypergraph enabled cross-domain resource management in satellite networks. *IEEE Trans. Commun.* **2022**, *70*, 4687–4701. [\[CrossRef\]](#)
25. Li, J.; Wang, P.; Li, H.; Shi, K. Enhanced time-expanded graph for space information network modeling. *Sci. China Inf. Sci.* **2022**, *65*, 192301. [\[CrossRef\]](#)
26. Casadesus-Vila, G.; Ruiz-de Azua, J.A.; Alarcon, E. Toward autonomous cooperation in heterogeneous nanosatellite constellations using dynamic graph neural networks. *arXiv* **2024**, arXiv:2403.00692.
27. Phillips, S.; Petersen, C.; Fierro, R. Robust, resilient, and energy-efficient satellite formation control. In *Intelligent Control and Smart Energy Management: Renewable Resources and Transportation*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 223–251.
28. Shi, K.; Zhang, X.; Zhang, S.; Li, H. Time-expanded graph based energy-efficient delay-bounded multicast over satellite networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 10380–10384. [\[CrossRef\]](#)
29. Zhang, G.; Li, X.; Hu, G.; Li, Y.; Wang, X.; Zhang, Z. MARL-Based Multi-Satellite Intelligent Task Planning Method. *IEEE Access* **2023**, *11*, 135517–135528. [\[CrossRef\]](#)
30. Wang, F.; Jiang, D.; Qi, S.; Qiao, C.; Shi, L. A dynamic resource scheduling scheme in edge computing satellite networks. *Mob. Netw. Appl.* **2021**, *26*, 597–608. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.