algorithms

MDPI

*Article*

# A Distributed Autonomous Mission Planning Method for the Low-Orbit Imaging Constellation

**Qing Yang [1], Bingyu Song [2], Yingguo Chen [1], Lei He [1] and Pei Wang [1,*]**

[1] College of Systems Engineering, National University of Defense Technology, Changsha 410073, China; yangqnudt@nudt.edu.cn (Q.Y.); ygchen@nudt.edu.cn (Y.C.); helei@nudt.edu.cn (L.H.)

[2] The People's Armed Police Command College China, Tianjin 300250, China; sbynudt@outlook.com

[*] Correspondence: peiwang@nudt.edu.cn

**Abstract:** With the improvement of satellite autonomy, multi-satellite cooperative mission planning has become an important application. This requires multiple satellites to interact with each other via inter-satellite links to reach a consistent mission planning scheme. Considering issues such as inter-satellite link failure, external interference, and communication delay, algorithms should minimize communication costs as much as possible. The CBBA algorithm belongs to a fully distributed multi-agent task allocation algorithm, which has been introduced into multi-satellite autonomous task planning scenarios and achieved good planning results. This paper mainly focuses on the communication problem, and proposes an improved algorithm based on it, which is called c-CBBA. The algorithm is designed with task preemption strategy and single-chain strategy to reduce the communication volume. The task preemption strategy is an accelerated convergence mechanism designed for the convergence characteristics of CBBA, while the single-chain strategy is a communication link pruning strategy designed for the information exchange characteristics of satellites. Experiments in various scenarios show that the algorithm can effectively reduce communication volume while ensuring the effectiveness of task planning.

**Keywords:** multi-satellite system; distributed cooperative task planning; communication constraints

check for updates

**Citation:** Yang, Q.; Song, B.; Chen, Y.; He, L.; Wang, P. A Distributed Autonomous Mission Planning Method for the Low-Orbit Imaging Constellation. *Algorithms* **2023**, *16*, 475. https://doi.org/10.3390/a16100475

Academic Editor: Frank Werner

Received: 10 September 2023
Revised: 5 October 2023
Accepted: 8 October 2023
Published: 11 October 2023

## 1. Introduction

With the improvement of satellite autonomy, the Earth Observation Satellite (EOS) has gradually become capable of onboard perception, decision-making, and inter-satellite communication, playing an increasingly important role in the field of earth observation. Along with the new control mode, EOS can perform mission planning autonomously, rather than relying solely on commands from the ground. This enables the satellite to make decisions based on its own accurate state, while responding to changes in the external environment in a more timely manner, greatly reducing the response time to target observation. At the same time, it reduces the dependence on satellite tracking, telemetry and command resources, enhancing the robustness of the overall system. In the future, with the increase in the number of autonomous EOSs, multiple autonomous EOSs together form a constellation. Then the observation mission will not only rely on a single satellite, but also rely on the cooperation of various satellites in the constellation. Determining how to improve the efficiency of the entire constellation to ensure that more tasks are scheduled in a timely, accurate and high-quality manner will be the focus of future autonomous satellite mission planning [1]. Low-orbit satellite constellations have advantages such as low latency, strong signal, global coverage, and low cost, affording them important roles in remote sensing [2,3]. This paper focuses on the research of autonomous mission planning methods for low-orbit imaging constellations with inter-satellite links (ISLs).

In the traditional satellite mission planning mode, the ground obtains the onboard status through telemetry, tracking, etc., generates the mission planning scheme, and sends the command to the satellite. Satellites act only as executors of instructions and do not have autonomous decision-making capabilities. In this control mode, the imaging mission planning problem can be regarded as an optimization problem. There is currently a large amount of research in this field. For example, there are exact solution algorithms such as the branch and price algorithm [4], the dynamic programming algorithm [5], meta-heuristic algorithms such as the genetic algorithm [6], adaptive large-neighborhood search algorithm [7,8], and many heuristic algorithms [9,10]. However, due to the long satellite-ground loop, this mode cannot respond quickly to changes in the onboard environment.

As the autonomy of satellites increased, mission planning began to be combined with the autonomous capabilities of satellites. Collaboration is required in a multi-satellite environment. Two types of multi-satellite collaboration architectures are often studied: the centralized architecture and the distributed architecture. Centralized architectures usually rely on a central node that has access to all child nodes, and then performs global planning based on the collected information. For example, Truszkowski et al. [11] proposed a spacecraft constellation interaction scenario, constructing a fully centralized architecture with high-orbit satellites as master satellites and low-orbit satellites as child satellites. Master satellites receive the status information sent by the low-orbit child satellites and are responsible for the mission planning of the whole constellation. Adopting a centralized architecture has many advantages, such as less communication required and higher quality solutions. However, the centralized architecture requires high computing power of the master satellite and a good communication environment. Considering issues such as ISL failures, external interference, and communication delays, it is difficult for the master satellite to obtain information from all child satellites in a timely manner [12]. In addition, once the master satellite fails, the entire system will fall into paralysis. In order to cope with more complex communication environments and to improve system robustness, the distributed architecture is proposed.

In a distributed architecture, each node in the system has autonomous capabilities. Child nodes not only act as executors of instructions, but also undertake certain computational tasks. The distributed architecture can be further divided into the centralized-distributed architecture and the fully distributed architecture. In a centralized-distributed architecture, there exists one or more master agents that divide the multiple agents into different groups for control, while each agent can make decisions independently. Zhang et al. [13] proposed a distributed broken-chain reconnection algorithm, where the master satellite splits the scheduling task, transmits sub-tasks to the neighboring auxiliary satellites for computation, and then the child satellites return the scheduled results to the master satellite. The approach can break through the limitations of single satellite computing resources and improve in-orbit computing capability. However, the algorithm can only perform task planning for a single satellite. Contract network protocol (CNP) [14] is a classic centralized-distributed algorithm. CNP adopts the market auction mechanism to allocate tasks in sequence. It has been widely studied in the field of satellite mission planning. Zhang et al. [15] divided the satellite mission planning problem into two levels: satellite cluster mission planning and single-satellite autonomous planning, and proposed an improved CNP algorithm. Long et al. [16] proposed a multi-satellite collaborative mission planning architecture based on the improved Shuffled Frog Leaping Algorithm (SFLA). The master satellite is responsible for task allocation, while child satellites are responsible for task scheduling and calculation of fitness values. Although the child nodes in the centralized-distributed architecture take on a certain amount of computational work, it still puts high demands on the master satellite's computational capability, and there is still a problem of poor system robustness. The fully distributed architecture, on the other hand, does not have a central node, and the nodes generate solutions by mutual negotiation. Determining how to achieve consistency in solutions through decentralized communication is a key issue for the architecture. Zheng et al. [17] addressed collaborative planning scenarios for child satellites after the failure of the master satellite. The Utility-based Regret play was proposed as a negotiation mechanism for teams in a centralized-distributed structure, and the Smoke Signal play and the Broadcast-based play were used as negotiation mechanisms for teams in a fully distributed structure. Gao et al. [18] proposed a multi-agent collaborative coevolutionary genetic algorithm, where multiple subpopulations distributed on different satellites evolve in parallel to continuously optimize the combination of satellite schemes. Zheng et al. [19] proposed a distributed algorithm based on the Hybrid Dynamic Mutation Genetic Algorithm (HDMGA) with a local search heuristic module and a distributed global optimization module. However, as the constellation and mission scale increase, the computational efficiency of the metaheuristic algorithm for iterative optimization is greatly affected. Therefore, considering the limitations of on-board computing power and inter-satellite communication conditions, we need to explore a low complexity and low communication cost algorithm to ensure efficient task allocation.

Choi et al. [20] proposed the Consensus-based Bundle Algorithm (CBBA), which employed a market auction mechanism for decentralized task selection and a consensus strategy based on local communication to resolve conflicts. The algorithm can accomplish task planning within polynomial time while ensuring the quality of planning. Song et al. [21] introduced CBBA into the imaging satellite mission planning problem, solved the coupling constraint problem in mission planning, and explored the fitness functions in different scenarios. However, when the scale of satellite missions increases, there will still be a significant communication burden. The paper proposes the c-CBBA algorithm optimized specifically for the communication problem. The task preemption mechanism and the single-chain strategy are designed. The algorithm is able to significantly reduce the communication volume while guaranteeing the optimization quality, which is more suitable for efficient mission scheduling in the context of restricted inter-satellite communication.

The rest of the paper is organized as follows. Section 2 describes the Multi-Satellite On-board Observation Scheduling Problem (MSOOSP). Section 3 describes the basic CBBA algorithm and analyses its drawbacks. Section 4 proposes the improved algorithm c-CBBA with two important strategies: the task preemption strategy and the single-chain strategy. Section 5 describes the design of test cases and the results of experiments in various scenarios. The discussion is provided in Section 6 and the conclusions are given in Section 7.

## 2. Problem Description

The Multi-Satellite On-board Observation Scheduling Problem (MSOOSP) is a task-resource matching problem. The scheme is generated through autonomous communication and decision-making among various satellites. The plan needs to determine the execution satellite of the mission while meeting various constraints. The objective is to maximize the sum of the benefits obtained from performing the missions.

For ease of reference, a summary of the notations used in the remainder of the paper is presented as follows. Table 1 shows the description list of abbreviations. Tables 2 and 3 show the description lists of mathematical indices and symbols.

**Table 1.** Description list of abbreviations.

| Abbreviations | |
|---|---|
| EOS | the Earth Observation Satellite |
| ISL | the Inter-Satellite Link |
| MSOOSP | the Multi-Satellite On-board Observation Scheduling Problem |
| O-VTW | Observation-Visible Time Window between the satellite and mission |
| C-VTW | Communication-Visible Time Window between satellites |
| TPS | the Task Preemption Strategy |
| SCS | the Single-Chain Strategy |
| CNP | the Contract Network Protocol algorithm |

**Table 2.** Description list of mathematical indices.

| **Mathematical Indices** | |
|---|---|
| $i$, $i'$, $i''$ | LEO satellite index, $i = 1, 2 \ldots \lvert S \rvert$ |
| $j, j'$ | mission index, $j = 1, 2 \ldots \lvert T \rvert$ |
| $k$ | the O-VTW index, $k = 1, 2 \ldots \left\lvert W_{ij} \right\rvert$ |

**Table 3.** Description list of mathematical symbols.

| **Mathematical Symbols** | | | |
|---|---|---|---|
| $s_i$ | the satellite with index i | $\lvert S \rvert$ | the number of satellites |
| $M_i$ | the maximum storage capacity of satellite $s_i$ | $t_j$ | the mission with index j |
| $\lvert T \rvert$ | the number of missions | $m_j$ | the estimated consumption of storage for the execution of task $t_j$ |
| $d_j$ | the required observing duration of task $t_j$ | $ts_j$ | the start time of the execution of task $t_j$ |
| $te_j$ | the end time of the execution of task $t_j$ | $tr_{jj,}$ | the required attitude transition time for the satellite to perform two adjacent tasks between $t_j$ and $t_{j,}$ |
| $p_j$ | the priority of the task $t_j$ | $r_j$ | the benefit of task $t_j$ |
| $\lvert W_{ij} \rvert$ | the total number of O-VTWs of satellite $s_i$ for task $t_j$ | $w_{ijk}$ | the kth O-VTW of satellite $s_i$ for task $t_j$ |
| $ws_{ijk}$ | the start time of $w_{ijk}$ | $we_{ijk}$ | the end time of $w_{ijk}$ |
| $x_{ijk}$ | the decision variable that determines whether the task $t_j$ is executed in $w_{ijk}$ | $g_{ijk}$ | the bidding function for the time window $w_{ijk}$ |
| $\alpha$ | the parameter of TPS | $b_i$ | the task bundle constructed by satellite $s_i$ |
| $z_i = \left\{ z_{i1}, z_{i2}, \ldots, z_{i\lvert T \rvert} \right\}$ | the highest bid satellite list of all tasks under the satellite $s_i$'s cognition | | |
| $y_i = \left\{ y_{i1}, y_{i2}, \ldots, y_{i\lvert T \rvert} \right\}$ | the highest bid list of all tasks under the satellite $s_i$'s cognition | | |
| $h_i = \left\{ h_{i1}, h_{i2}, \ldots, h_{i\lvert T \rvert} \right\}$ | the timestamp list of other satellite's information obtained by satellite $s_i$ | | |
| $u_i = \left\{ u_{i1}, u_{i2}, \ldots, u_{i\lvert T \rvert} \right\}$ | the list of task preemption markers under the cognition of satellite $s_i$ | | |
| $v_i = \left\{ v_{i1}, v_{i2}, \ldots, v_{i\lvert T \rvert} \right\}$ | the task preemption timestamp list under the cognition of satellite $s_i$ | | |
| $q_i = \left\{ q_{i1}, q_{i2}, \ldots, q_{i\lvert T \rvert} \right\}$ | the number of communications made by the satellite $s_i$ that consistently considers itself to be the highest bidder | | |

### *2.1. MSOOSP Task Planning Process*

Figure 1 shows the task planning process of MSOOSP, which is divided into the following three parts.

(1) Phase 1. Task broadcast

When the ground station receives a mission request from the user, it first uploads the mission information to the satellite that is able to establish a satellite-to-earth link at the current moment. The satellite then broadcasts the received information through the inter-satellite link so that all satellites in the constellation have access to the mission information.

It should be noted that for smaller constellations, when the satellite receives information transmitted by the ground station, it can broadcast throughout the entire constellation range as described above. With the tremendous progress of small satellite technology, low-orbit large-scale small satellite constellations have become an important development direction, resulting in a large number of research and application plans [22,23]. Adopting fully distributed algorithms for autonomous task planning in large-scale constellations is inefficient and unnecessary. Considering the efficiency and effectiveness of the algorithm, the task broadcast range can be predetermined based on the characteristics of the constellation, and then the task planning algorithm can be carried out within this range.

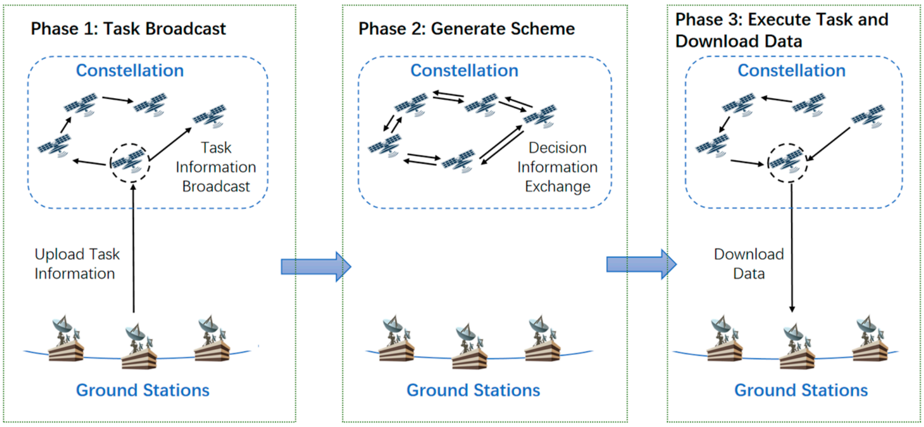


**Figure 1.** MSOOSP task planning process.

In order to ensure the normal operation and long-term control of satellites, ground stations routinely obtain information such as satellite flight orbits and operational status through measurement, control and tracking. Therefore, when the user puts forward the demand, the ground station is able to directly calculate the information such as the visible time window of each satellite for each task using tools such as Satellite Ephemeris. In order to reduce the computational burden of the on-board hardware, the pre-processing procedure of the information required for planning, including mission attributes, observation-visible time window (O-VTW, i.e., the period when the satellite is able to observe the ground target), communication-visible time window (C-VTW, i.e., the period when satellites are able to communicate with each other), satellite attitude, etc., is completed on the ground and uploaded to the satellite via the satellite-to-earth link.

(2) Phase 2. On-board mission scheduling

When the satellite receives task information, the task planning algorithm is triggered. Each satellite runs the scheduling algorithm and generates the final scheduled plan through autonomous decision-making and inter-satellite communication.

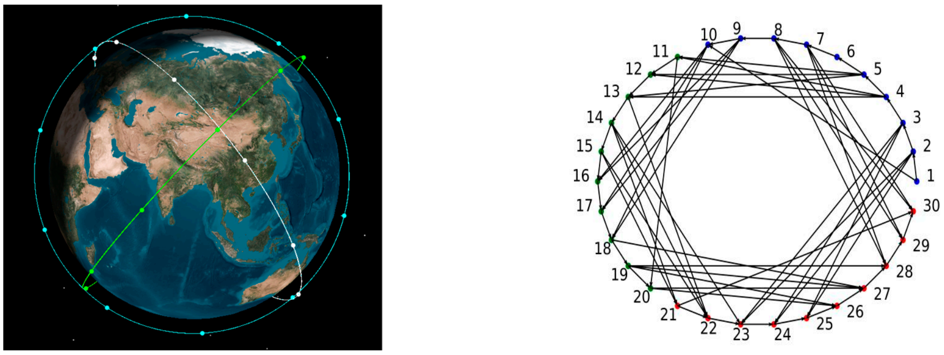
(3) Phase 3. Execute mission planning scheme and download data

Each satellite performs missions according to the scheduled scheme. When the satellite is visible to the ground station, the observation data is transmitted down to the ground station. Alternatively, it can be transmitted down to the ground station through other satellites using inter-satellite links.

This paper focuses on the task planning algorithm for the second part.

### 2.2. *Onboard Communication Topology*

Satellites carry communication payloads, such as radio or laser, which allow the satellite to establish ISLs to transmit data. The establishment of ISLs have many constraints, for example, the distance between the two satellites needs to be within a certain range and the line-of-sight vector between the two satellites needs to be above the earth's surface. Taking the Walker-$\delta$ (30/3/1, 600 km, 60°) constellation (denoted as: Walker constellation, ten satellites uniformly distributed on each orbital plane, a total of three orbital planes, phase factor of 1, orbital inclination of 60°, and orbital altitude of 600 km) as an example. Its constellation configuration is shown in Figure 2a, and the communication topology is shown in Figure 2b. Among them, satellites 1 to satellite 10 belong to the same orbital plane, and their communication topologies are sequentially connected to form a bidirectional closed loop.



(**a**) (**b**)

**Figure 2.** Walker-δ (30/3/1, 600 km, 60°) constellation [21]. (**a**) Three-dimensional graph of satellite constellations; (**b**) communication topology among satellites, a line indicates that an inter-satellite link can be established between two satellites for communication.

When there are a large number of satellites on the same orbital plane, due to the smaller distance between satellites, satellites can not only establish links with adjacent satellites, but also with further satellites. Taking the Walker-δ (20/1/1, 600 km, 60°) constellation as an example, the communication topology at a certain moment is shown in Figure 3.

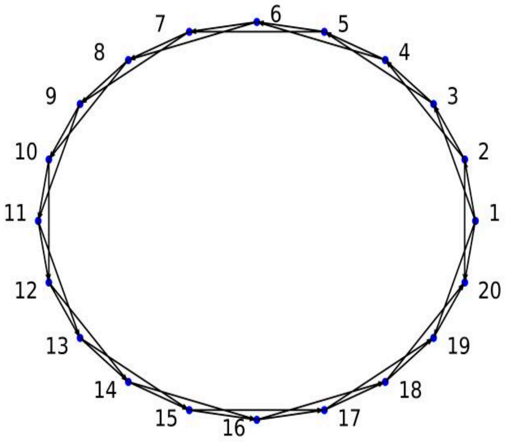


**Figure 3.** Communication topology of the Walker-δ (20/2/1, 600 km, 60°) constellation.

For a certain satellite, the satellite that can be linked to it is called the neighbor satellite. Taking Figure 3 as an example, the neighbors of satellite 1 are $\{s_2, s_3, s_{19}, s_{20}\}$.

### 2.3. Problem Assumption

To highlight the key points of the problem, combined with the actual situation, this paper makes the following assumptions in MSOOSP:

- One satellite can only communicate with one neighbor at a time.
- The communication between two satellites is unidirectional, where one satellite serves as the information sender and the other as the information receiver.
- The paper does not consider the communication costs incurred due to pre-processed information broadcasting.
- The paper does not consider changes in satellite power and storage caused by factors other than mission observations.

### 2.4. Notations

$S = \left\{s_1,\ s_2, \ldots,\ s_{|S|}\right\}$, the set of satellites, $|S|$ represents the number of satellites. $M_i$ represents the maximum storage capacity of satellite $s_i$ during the planning period.

$T = \left\{t_1,\ t_2, \ldots t_{|T|}\right\}$, the set of missions, $|T|$ denotes the number of missions. $m_j$ denotes the estimated consumption of storage for the execution of task $t_j$. $d_j$ is the required observing duration of task $t_j$. $ts_j$ denotes the start time of the execution of task $t_j$, $te_j$ denotes the end time of the execution of task $t_j$. $tr_{jj'}$ represents the required attitude transition time for the satellite to perform two adjacent tasks between $t_j$ and $t_{j'}$. $p_j$ denotes the priority of the task $t_j$. $r_j$ represents the benefit of task $t_j$. Considering the urgency and timeliness of observing tasks, this article introduces a mechanism for task benefits to decay over time. On the basis of the priority $p_j$, the benefit $r_j$ decays with the delay of its execution start time $ts_j$. The calculation formula is shown in Equation (1), λ is the attenuation factor.

$$r_j = p_j e^{-\lambda ts_j}, \tag{1}$$

$W_{ij} = \left\{w_{ij1},\ w_{ij2}, \ldots,\ w_{ij|W_{ij}|}\right\}$, the set of O-VTWs of satellite $s_i$ for task $t_j$, $\left|W_{ij}\right|$ denotes the total number of O-VTWs of satellite $s_i$ for task $t_j$. $w_{ijk}$ denotes the kth O-VTW of satellite $s_i$ for task $t_j$. $ws_{ijk}$ indicates the start time of $w_{ijk}$, $we_{ijk}$ indicates the end time of $w_{ijk}$.

$x_{ijk}$, decision variable. $x_{ijk} = 1$ if task $t_j$ is executed in $w_{ijk}$, $x_{ijk} = 0$ otherwise.

### 2.5. Mathematical Formulation

Define the scheduling model for MSOOSP. The model is:

$$\max \sum_{i=1}^{|S|} \left( \sum_{j=1}^{|T|} \left( \sum_{k=1}^{|W_{ij}|} x_{ijk} r_j \right) \right), \tag{2}$$

subject to:

$$\sum_{j=1}^{|T|} \left( \sum_{k=1}^{|W_{ij}|} x_{ijk} m_j \right) \leq M_i,\ \ \forall i \in S, \tag{3}$$

$$\sum_{i=1}^{|S|} \left( \sum_{k=1}^{|W_{ij}|} x_{ijk} \right) \leq 1,\ \ \forall j \in T, \tag{4}$$

$$ws_{ijk} \leq ts_j < te_j \leq we_{ijk},\ \ if\ x_{ijk} = 1, \tag{5}$$

$$te_j + tr_{jj'} \leq ts_{j'},\ \ if\ te_j < ts_{j'}, \tag{6}$$

$$ts_j + d_j = te_j,\ \ \text{if} \sum_{i=1}^{|S|} \left( \sum_{k=1}^{|W_{ij}|} x_{ijk} \right) = 1,\ \ \forall j \in T, \tag{7}$$

$$x_{ijk} \in \{0,\ 1\},\ \ \forall (i, j, k) \in S \times T \times W, \tag{8}$$

Equation (2) represents the common objective function of all satellites, maximizing the sum of benefits of successfully assigned tasks. Equation (3) indicates that during the planning cycle, the storage occupied by the satellite to perform its tasks cannot reach its maximum storage capacity; Equation (4) indicates that each task is executed at most once; Equation (5) indicates that the actual execution time of a task must be within its O-VTW; Equation (6) indicates that there must be a certain attitude transition time between two adjacent observation tasks of the satellite; Equation (7) indicates that mission observations need to last for a certain amount of time; Equation (8) defines the range of values for the decision variables.

## 3. The Basic CBBA

### 3.1. Structure of the CBBA

CBBA is an auction-based distributed algorithm, where each satellite has the capacity for autonomous decision-making. Each satellite continuously obtains planning information from other satellites through local communication, ultimately reaching a consensus scheme. During the running time, each satellite needs to maintain two lists to record the global bidding information under the satellite's cognition, which are: the highest bid satellite and its bid corresponding to each task, denoted by $z_i = \left\{z_{i1}, z_{i2}, \ldots, z_{i|T|}\right\}$ and $y_i = \left\{y_{i1}, y_{i2}, \ldots, y_{i|T|}\right\}$, respectively. Additionally, it is necessary to record the timestamp list of other satellite's information obtained by the satellite, represented by $h_i = \left\{h_{i1}, h_{i2}, \ldots, h_{i|T|}\right\}$. At the same time, each satellite is required to maintain a task set of its choice, denoted by $b_i$.

The CBBA consists of two phases. The first phase is the task bundle construction phase. In this phase, each agent independently and greedily selects the task with the highest bid one-by-one from the task set from which they believe they have bid higher than the current global highest bid. The second phase is the consensus check phase. At this stage, each agent updates its knowledge of global information by interacting with neighboring agents, which is then used to guide the updating and construction of the task bundle. After communicating with neighbors, for the task in $b_i$, when there is a satellite bidding higher, the satellite $s_i$ needs to give up the task. Since tasks are added sequentially to the task bundle in the first phase, each addition will have a cascading effect on the remaining tasks to be scheduled (e.g., the tasks in the bundle set need to satisfy the constraint of Equation (6), and if there is an overlap in the O-VTW of the two tasks, the addition of one task will result in the loss of bidding eligibility for the other task due to the failure to satisfy the constraint). So, if a task is abandoned, all subsequent tasks added to the task bundle will be invalidated. Therefore, we need to discard the task and all subsequent tasks in the task bundle, and initialize the information of all tasks after that task in $y_i$ and $z_i$. This step is called conflict resolution. With the continuous iteration of the two stages, more and more tasks have had the final winner of the bid determined, and a consensus on the task planning has been ultimately reached.

The process of CBBA from the perspective of single satellite $s_i$ is shown in Figure 4. The inside part of the dotted box represents the satellite's built-in operating program, and the outside part of the dotted box is the satellite's information interaction with the environment.

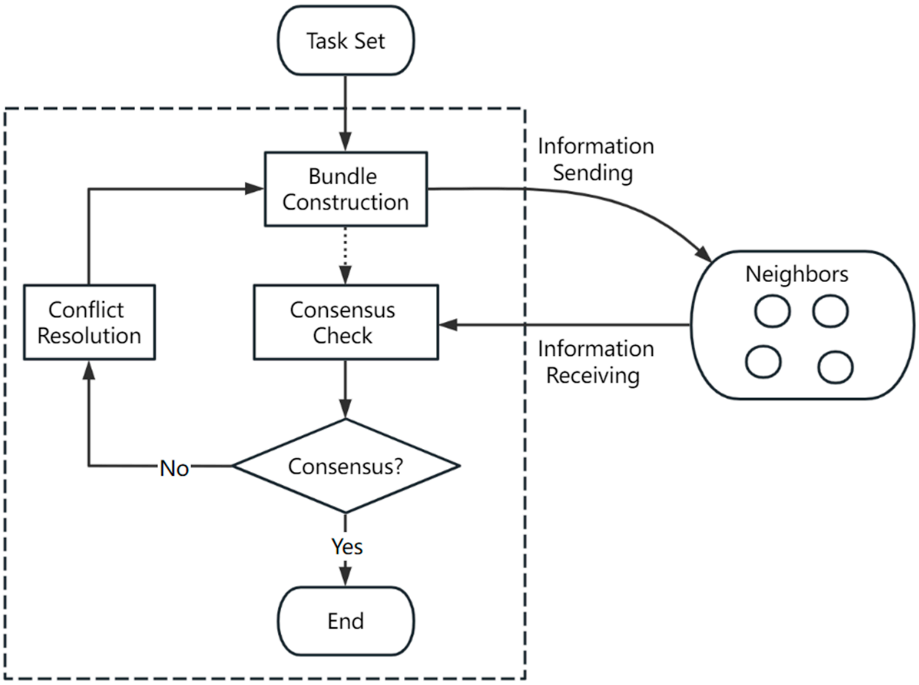


**Figure 4.** Flow chart of CBBA from a single satellite perspective.

### *3.2. Drawbacks of CBBA in Imaging Satellite Mission Planning*

From the introduction of the CBBA algorithm above, it can be seen that for task $t_j$, which currently has the highest global bid, satellite $s_i$ will prioritize adding to its own task bundle, and all satellites will obtain and update this information during the consensus check phase. Therefore, CBBA will prioritize determining the bidding result for the task with the highest global bid, and achieve consensus on this information globally. However, the selection of tasks affects each other. For example, the execution of one task may cause the O-VTW of other tasks to be unavailable (which can be regarded as a bid of 0), i.e., some tasks will be masked. Throughout the entire CBBA process, hidden tasks may be re-qualified for election due to conflict resolution during the consensus check phase.

Let us take Figure 5 as an example to explain the masking phenomenon. The figure shows the O-VTW information and bidding information of the two satellites for task A and task B, respectively. Both missions are visible to satellite $s_1$, but only one of them can be observed due to time window overlap conflicts. Table 4 shows the scheduling process information of the two stages of CBBA. In the first iteration, $s_1$, prioritizes task A with a higher bid, and due to time window conflicts, the bid for task B is 0. In the second iteration, task B regains the bidding qualification due to task A not being added to the bundle. With the increase in satellite scale and task scale, the frequency of such phenomena increases greatly, and even there are conflicting relationships among multiple tasks. At this point, the change in one task can affect the bidding of multiple tasks. This masking phenomenon leads to capricious bids for the same task. In the worst case, after a round of auction, only one task's scheduling result may be determined, that is, the task with the highest global bid. At this point, when the number of tasks is $|T|$, $|T|$ rounds of auctions are needed to determine the final scheduling result, which to some extent limits the convergence speed of the algorithm and indirectly increases the communication cost of the algorithm.

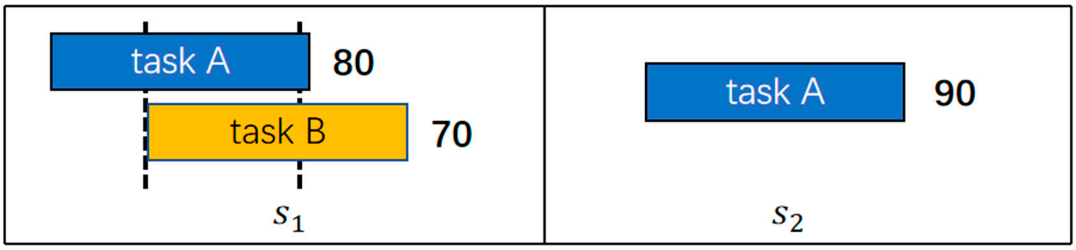


**Figure 5.** Example of masking phenomenon, which shows the visible time windows and their conflict relationship of two satellites for task A and B, as well as the bidding value for tasks.

**Table 4.** Solution process of CBBA for the masking issues.

| Iteration | Phase I | After Phase II |
|---|---|---|
| Iter1 | $b_1 = \{taskA\}$, $b_2 = \{taskA\}$ | $b_1 = \{\}$, $b_2 = \{taskA\}$ |
| Iter2 | $b_1 = \{taskB\}$, $b_2 = \{taskA\}$ | $b_1 = \{taskB\}$, $b_2 = \{taskA\}$ |

## 4. c-CBBA

Based on the above analysis, this section proposes an improved CBBA algorithm for communication problems, called c-CBBA, which mainly includes two strategies: the Task Preemption Strategy (TPS) and the Single-Chain Strategy (SCS). Figure 6 shows the process of c-CBBA. From the figure, it can be seen that TPS mainly acts on the consensus check stage of the basic CBBA, accelerating algorithm convergence by improving the auction rules. SCS is mainly used for inter-satellite communication. By analyzing the onboard communication topology, the communication links are pruned so as to reduce the communication cost of the algorithm in large-scale scenarios.

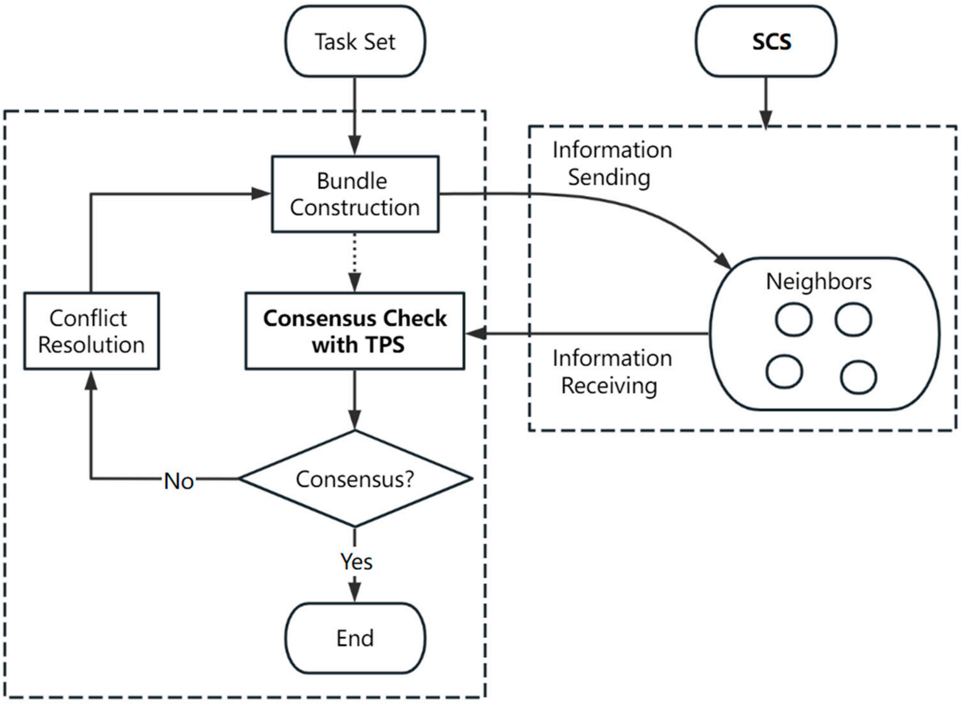


**Figure 6.** Flow chart of c-CBBA from a single satellite perspective.

### *4.1. Task Preemption Strategy*

Based on the analysis of the implicit drawbacks of CBBA algorithm, the Task Preemption Strategy (TPS) is introduced to accelerate the convergence of the algorithm. In TPS, single satellite can autonomously preempt tasks. If the satellite continues to believe that it has the highest bid after aggregating the auction messages sent by its neighbors for a certain number of communications, it can preempt the task and inform the other satellites of this message. Then, other satellites will no longer add the preempted task to their own task bundle. At the global level, TPS overcomes the drawbacks of the task masking phenomenon. It is not necessary to wait for a high-bid task to identify the winning satellite before a lower-bid task can be efficiently scheduled, thus accelerating the process of global consistency. During the distributed operation of the algorithm, there may be a situation where multiple satellites are preempting the same task. Therefore, it is necessary to record the timestamp of the satellite's preemption of the task, and stipulate that the satellite that first preempts the task acquires the task.

Each satellite requires additional maintenance of the following information:

- $u_i = \left\{u_{i1}, u_{i2}, \ldots, u_{ij}, \ldots, u_{i|T|}\right\}$, the list of task preemption markers under the cognition of satellite $s_i$, its length is the number of tasks $|T|$. $u_{ij}$ represents the preemptive marker of task $t_j$. If $t_j$ has already been preempted, then $u_{ij} = 1$; otherwise, $u_{ij}$ =0.
- $v_i = \left\{v_{i1}, v_{i2}, \ldots, v_{ij}, \ldots, v_{i|T|}\right\}$, the task preemption timestamp list under the cognition of satellite $s_i$, its length is $|S|$. $v_{ij}$ represents the timestamp of the earliest time when task $t_j$ was preempted under the cognition of satellite $s_i$.
- $q_i = \left\{q_{i1}, q_{i2}, \ldots, q_{ij}, \ldots, q_{i|T|}\right\}$, recording the number of communications made by the satellite that consistently considers itself to be the highest bidder. Its length is $|T|$.
- $\alpha$, the parameter of TPS. When $q_{ij}$ reaches $\alpha$ times, satellite $s_i$ preempts task $t_j$.

#### 4.1.1. Bundle Construction

This phase is essentially a mission scheduling phase performed independently by each satellite. The task bundle construction process is greedy. Satellite $s_i$ selects the highest bid task from its own unscheduled tasks to join the task bundle $b_i$ until there are no more tasks that meet the requirements. Note that the tasks in the task bundle need to be arranged in the order of addition. The selected task $t_j$ must meet three conditions simultaneously:

- The task bundle $b_i$ needs to be able to satisfy the constraints of the model after joining task $t_j$.
- Satellite $s_i$'s bid for task $t_j$ needs to be higher than the highest bid recorded for $t_j$ in the list $y_i$ it maintains.
- Task $t_j$ is currently not preempted, i.e., $u_{ij} = 0$.

As task $t_j$ is added to $b_i$, the corresponding task information in $y_i$ and $z_i$ should be updated at the same time.

Equation (9) defines the bidding function for the time window $w_{ijk}$.

$$g_{ijk} = \frac{r_j - \sum_{w_{ij'k'} \in Conf_{ijk}} r_{j'}/|S|}{m_j}, \tag{9}$$

where $Conf_{ijk}$ denotes the set of O-VTWs that have constraint conflicts (Equation (6)) with the time window $w_{ijk}$. The bidding function aims to select tasks with high profit, less storage usage, and low conflict with other tasks.

When tasks are executed on the same satellite, there are many coupling constraints, such as the constraints on the satellite attitude transition time represented in Equation (6). Therefore, during the bundle construction phase, constraint checking is required for every task added.

### 4.1.2. Consensus Check

In this phase, each agent updates its own knowledge about global information by interacting with neighboring agents, and then updates the task bundle accordingly. The consensus check phase under the TPS is as follows:

Step 1: when satellite $s_i$ receives the neighbor's message, it checks for consistency with its own message. If inconsistent, update the information. The basic CBBA needs to update the highest bid list $y_i$, highest bid satellite list $z_i$, and timestamp list $h_i$. The update rules are shown in Table 5.

There are three types of rules for updating information:

- update: $y_{ij} = y_{i'j}, z_{ij} = z_{i'j}$;
- reset: $y_{ij} = 0, z_{ij} = \text{none}$;
- leave: $y_{ij} = y_{ij}, z_{ij} = z_{ij}$.

The timestamp list $h_i$ is updated in such a way: when the satellite $s_i$ communicates with $s_{i'}$, the timestamp $h_{ii'}$ is updated to the moment $t$ of this communication, and the timestamp $h_{ii''}$, in which satellite $s_i$ obtains information about the other satellites $s_{i''}$, is updated to the larger of $h_{ii''}$ and $h_{i'i''}$. The timestamp update formula is shown in Equation (10) and (11)

$$h_{ii'} = t, \tag{10}$$

$$h_{ii''} = \max\{h_{ii''}, h_{i'i''}\}, \ \forall i'' \in T \backslash \{i, i'\}. \tag{11}$$

The c-CBBA with TPS also needs to update the list of task preemption markers $u_i$ and preemptive timestamp $v_i$. Table 6 lists the update rules for c-CBBA, which are modified based on Table 5.

The update rules for auction information are as follows:

- update: $y_{ij} = y_{i'j}, z_{ij} = z_{i'j}, u_{ij} = u_{i'j}, v_{ij} = v_{i'j}$;
- leave: $y_{ij} = y_{ij}, z_{ij} = z_{ij}, u_{ij} = u_{ij}, v_{ij} = v_{ij}$

Step 2: update $q_i$. After communicating with a neighbor, if task $t_j$ is not preempted and $z_{ij} = i$, then the continuous communication count $q_{ij}$ will be increased by 1; If $z_{ij} \neq i$, $q_{ij}$ is set to 0.

**Table 5.** CBBA's rules for updating information. It specifies the way in which a satellite updates its own information when it receives information transmitted by a neighboring satellite.

| $s_{i'}$ (Sender) Think $z_{i'j}$ Is | $s_i$ (Receiver) Think $z_{ij}$ Is | Receiver's Action (Default: Leave) |
|---|---|---|
| $i'$ | $i$ | if $y_{i'j} > y_{ij} \rightarrow$ update |
| | $i'$ | update |
| | $i'' \notin \{i,\ i'\}$ | *if* $h_{i'i''} > h_{ii''}$ *or* $y_{i'j} > y_{ij} \rightarrow$ update |
| | none | update |
| $i$ | $i$ | leave |
| | $i'$ | reset |
| | $i'' \notin \{i,\ i'\}$ | if $h_{i'i''} > h_{ii''} \rightarrow$ reset |
| | none | leave |
| $i'' \notin \{i,\ i'\}$ | $i$ | if $h_{i'i''} > h_{ii''}$ and $y_{i'j} > y_{ij} \rightarrow$ update |
| | $i'$ | if $h_{i'i''} > h_{ii''} \rightarrow$ update<br>else → reset |
| | $i''$ | $h_{i'i''} > h_{ii''} \rightarrow$ update |
| | $i^{''}{}' \notin \{i,\ i',\ i''\}$ | if $h_{i'i''} > h_{ii''}$ and $h_{i'i'''} > h_{ii'''} \rightarrow$ update<br>if $h_{i'i''} > h_{ii''}$ and $y_{i'j} > y_{ij} \rightarrow$ update<br>if $h_{i'i'''} > h_{ii'''}$ and $h_{ii''} > h_{i'i''} \rightarrow$ reset |
| none * | $i$ | leave |
| | $i'$ | update |
| | $i'' \notin \{i,\ i'\}$ | if $s_{i'i''} > s_{ii''} \rightarrow$ update |
| | none | leave |

* none indicates that the satellite has not obtained bidding information about the task or the information has been initialized.

**Table 6.** c-CBBA's rules for updating information.

| $s_{i'}$ (Sender) Think $z_{i'j}$ Is | $s_i$ (Receiver) Think $z_{ij}$ Is | Receiver's Action (Default: Leave) |
|---|---|---|
| 0 | 0 | Update $y_{ij}$ and $z_{ij}$ according to Table 5 |
| | 1 | leave |
| 1 | 0 | update |
| | 1 | if $v_{i'j} > v_{ij} \rightarrow$ leave<br>else → update |

Step 3: task preemption. If task $t_j$ is not preempted, determine whether the continuous communication count $q_{ij}$ has reached $\alpha$: if it has, the satellite $s_i$ will preempt task $t_j$. After preemption, update the preemption marker $u_{ij}$ to 1, update the preemption timestamp $v_{ij}$ to the current communication time, and advance task $t_j$ to the first position of $b_i$ (the purpose of the advance operation is to separate preempted tasks from non-preempted tasks, preventing preempted tasks from being discarded when the task bundle is updated at a later time).

Step 4: update the task bundle. For preempted tasks, when the preemption fails because the preemption time is later than other satellites, they are to be removed directly from $b_i$. For non preemptive tasks, adding each task during the construction of the task bundle will have an impact on subsequent task additions. Therefore, if the information is inconsistent after interacting with neighbors, a disintegration operation should be performed. Denote the position of the most forward task of $z_{ij} \neq i$ in $b_i$ as $\overline{n}$. Reset the $y_{ij}$ and $z_{ij}$ of the corresponding task after the $\overline{n}$ position in $b_i$. Decompose the task on position $\overline{n}$

and all tasks behind position $\overline{n}$ in $b_i$. Equations (12)–(14) provides specific update formulae.

$$y_{ib_{in}} = 0, \ \ \forall n > \overline{n}, \tag{12}$$

$$z_{ib_{in}} = \text{none}, \ \ \forall n > \overline{n}, \tag{13}$$

$$b_{in} = \emptyset, \ \ \forall n \geq \overline{n}, \tag{14}$$

### *4.2. Single-Chain Strategy*

According to whether the two satellites establishing the inter-satellite link belongs to the same orbit, the inter-satellite link is divided into two types: intra-plane ISLs and inter-plane ISLs. Satellites in the same orbital plane have stable inter-satellite links due to their stable relative positions, i.e., they have fixed "neighbors" that do not change over time. On the contrary, as relative motion occurs between satellites in different orbital planes, the connectivity of the inter-plane ISLs is time-varying and less stable.

As can be seen from Figure 3, when the constellation scale is large, each satellite has more than one neighbor, which is prone to information interaction redundancy during communication. For example, in Figure 7, satellites $s_2$ and $s_3$ are both neighbors of $s_1$, and $s_2$ and $s_3$ are neighbors of each other. If satellites $s_1$, $s_2$, and $s_3$ send information in sequence, the information transmitted in communication link $s_1 \rightarrow s_2 \rightarrow s_3$ already contains the information transmitted in $s_1 \rightarrow s_3$. Therefore, such multi-span communication links can be deleted. Based on the consideration of the stability of the communication link, the operation of removing the redundant link is only performed on the intra-plane ISLs. This operation is called the Single-Chain Strategy (SCS).

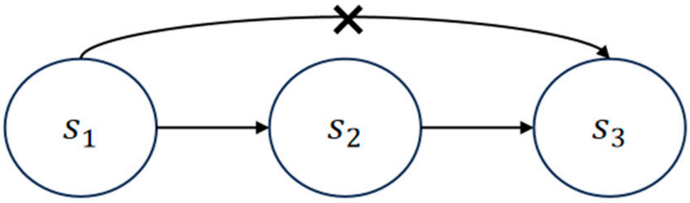


**Figure 7.** Redundant communication example, which shows the information transmission flow among three satellites. The information transmitted in link $s_1 \rightarrow s_2 \rightarrow s_3$ contains the information transmitted in $\mathrm{s}_1 \rightarrow \mathrm{s}_3$, so the link $\mathrm{s}_1 \rightarrow \mathrm{s}_3$ is redundant.

From a single satellite perspective, in the orbital plane, neighbors exist on both sides of the satellite. The single-chain strategy refers in particular to the fact that when there are multiple neighbors, the satellite selects only one neighbor for communication for each side. Assuming that the current satellite communication topology is shown in Figure 8a. Due to the fact that satellites are linked from an individual perspective, in order to ensure the connectivity of whole constellation, it can be specified that all satellites are linked according to the same rules, such as prioritizing the link with the nearest satellite. In this way, the communication topology is shown in Figure 8b. Considering the complexity of the communication environment, there may be situations where it is not possible to build a link due to special circumstances. In this case, the satellite can choose any neighbor to establish a link, as shown in Figure 8c.

The premise of CBBA convergence is that the communication topology is connected during the scheduling period. For other specific constellation configurations, we can design reasonable pruning strategies based on the characteristics of the constellation to improve information exchange efficiency while ensuring that the communication link topologies are connected graphs.

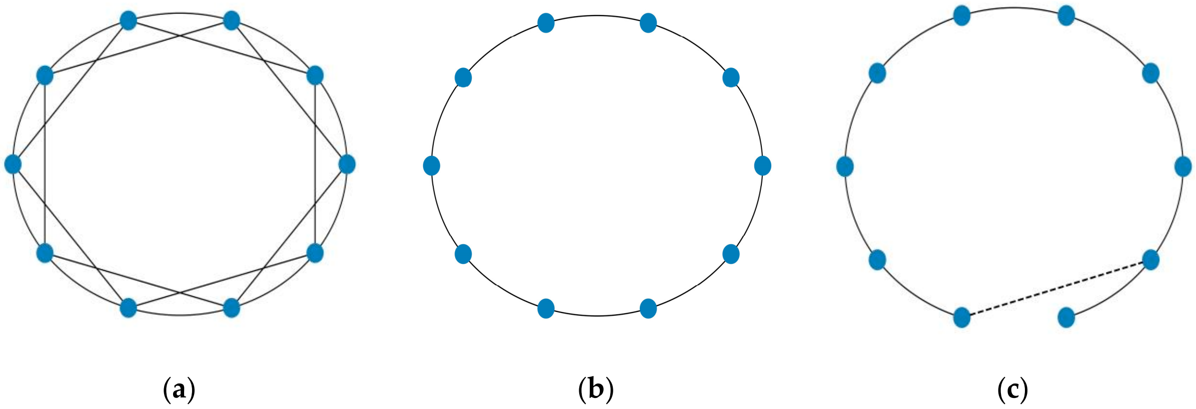


**Figure 8.** Constellation communication topology. (**a**) Original constellation communication topology. In addition to communicating with directly adjacent satellites, the satellite may be able to communicate with more distant satellites; (**b**) constellation communication topology under the single-chain strategy. Each satellite only communicates with directly adjacent neighboring satellites; (**c**) constellation communication topology under the single-chain strategy in case of partial communication link damage. When a neighboring satellite directly adjacent to the satellite fails, the satellite can communicate with a more distant neighboring satellite.

### *4.3. Response to Abnormal Situations*

The above description of the algorithm is based on the assumption that when the algorithm starts running, the satellite that is running normally remains normal throughout the entire algorithm running process. However, there may be a situation where a certain satellite malfunctions. This section provides a brief discussion based on this situation.

(1) Fault occurs after the algorithm operation

After the algorithm runs, each satellite can obtain the task allocation scheme for the entire constellation through its own maintained global highest bid satellite list $z_i$. If a satellite is unable to perform tasks due to a fault at this time, the task set assigned to the faulty satellite will be regarded as a new request and the c-CBBA algorithm will be rerun throughout the entire constellation.

(2) Fault occurs during the algorithm operation

Assuming that a satellite malfunctions during the operation of the algorithm. At this point, for any other satellite within the constellation, if the global highest bid satellite list maintained includes faulty satellites, initialize all task information corresponding to the faulty satellite, and then continue running the algorithm. Specifically, if $z_{ij}$ refers to the faulty satellite, the following variables need to be updated: $z_{ij}$ = none, $y_{ij} = 0$, $h_{ij} = 0$, $u_{ij} = 0$. Then, when the algorithm reaches global consistency, the faulty satellite will not be assigned to any tasks.

## 5. Results

### *5.1. Design of Tested Scenarios*

The algorithms are coded in C++, and the experiments are conducted using an Intel Core i7-6700 HQ 2.60 GHz CPU under a 64-bit Windows 10 system with 16 GB RAM.

The experimental scenarios are designed as follows: observation missions are generated according to a random distribution. They are divided into two categories according to the density of distribution: locally distributed tasks (3° N–53° N, 73° E–133° E) and globally distributed tasks (60° S–60° N). From each type, 360, 720 and 1080 tasks with O-VTWs within the planning cycle are randomly selected to form the set of examples, respectively. Set the estimated consumption storage $m_j$ and the priority $p_j$ of the task $t_j$ to a randomly sampled integer in [50, 100]. The time discount coefficient $\lambda$ for task profit is $10^{-5}$. Set the scheduling period to 1.5 h. The decay function of task benefits over time represented by

Equation (1) is shown in Figure 9, enabling the final profit to be approximately 95% of the original profit if the task is completed at the end of the planning cycle.

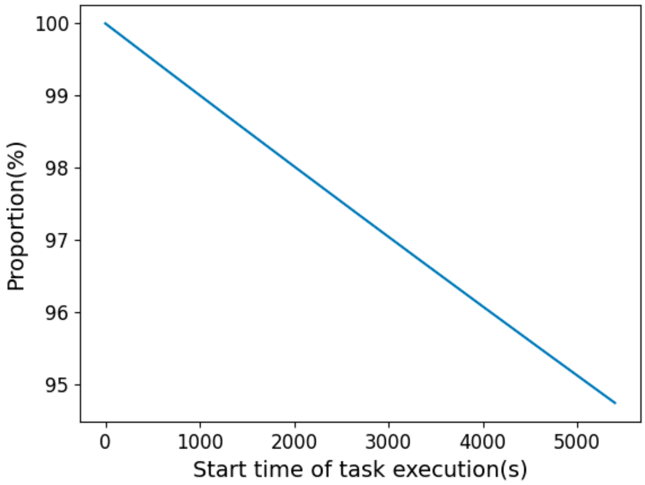


**Figure 9.** Task profit decay function over time $\left(\lambda = 10^{-5}\right)$.

The constellations are set to a scale of 30, 60 and 90 satellites. The constellation configuration parameters are Walker-δ (30/3/1, 600 km, 60°), Walker-δ (60/3/1, 600 km, 60°) and Walker-δ (90/3/1, 600 km, 60°), respectively. For global mission scenarios, set the maximum available storage of one satellite during the planning period to 750; for the regional mission scenarios, in order to highlight the time constraints, the storage constraints are appropriately relaxed by setting the maximum available storage for satellites during the planning cycle to 1125.

### 5.2. Experimental Result

This section mainly examines the performance of the algorithm on the objective function and the communication volume.

#### 5.2.1. Comparison of Algorithm Optimization Performance

The algorithm using Equation (9) as the bidding function is represented as c-CBBA (mix), while the algorithm using task profit as the bidding function is represented as c-CBBA (profit). The two are compared with the classical centralized-distributed algorithm Contract Network Protocol (CNP), which uses the task profit as the bidding function. In CNP, the master satellite bids for each task in the descending order of priority. Table 7 shows the comparison of various indicators of the three algorithms in a total of 18 scenarios with different task distributions, task scales, and satellite scales (the highest total profit is in bold). Table 7 compares the algorithms in terms of three aspects: the number of tasks successfully scheduled (TS), the total profit (TP), and the running time of the algorithms. For the convenience of expression, the following text uniformly refers to scenarios in the form of “task type—number of tasks—number of satellites”. For example, using “local-360-30” to refer to a scenario where the task is locally distributed, the number of tasks is 360, and the number of satellites is 30.

**Table 7.** Comparison of the results of various algorithms. TS refers to the number of tasks successfully scheduled; TP refers to the total profit.

| Task Type-Quantity | Sat | Scenario | c-CBBA (Mix) | | | c-CBBA (Profit) | | | CNP | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | TS | TP | CPU Time (s) | TS | TP | CPU Time (s) | TS | TP | CPU Time (s) |
| local | 30 | 1 | 237 | **18,002.2** | 0.11 | 228 | 17,805.9 | 0.103 | 231 | 17,985.1 | 0.001 |
| - | 60 | 2 | 335 | **24,458.2** | 0.50 | 331 | 24,156.6 | 0.501 | 331 | 24,155.9 | 0.001 |
| 360 | 90 | 3 | 350 | **26,217.9** | 1.16 | 348 | 26,096.5 | 1.206 | 348 | 26,096.5 | 0.001 |
| local | 30 | 4 | 277 | **22,321.3** | 0.19 | 255 | 21,674.9 | 0.183 | 253 | 21,522.9 | 0.002 |
| - | 60 | 5 | 496 | **38,083.1** | 1.02 | 475 | 37,488.8 | 1.037 | 471 | 37,233.3 | 0.002 |
| 720 | 90 | 6 | 628 | **47,275.2** | 2.93 | 612 | 46,462.6 | 2.765 | 611 | 46,374.1 | 0.003 |
| local | 30 | 7 | 298 | **24,807.6** | 0.29 | 263 | 23,484.9 | 0.273 | 264 | 23,550.7 | 0.005 |
| - | 60 | 8 | 552 | **44,125.3** | 1.49 | 522 | 43,552.6 | 1.481 | 521 | 43,511.4 | 0.006 |
| 1080 | 90 | 9 | 765 | **59,501.6** | 4.75 | 750 | 59,385.6 | 5.125 | 746 | 59,114.3 | 0.006 |
| global | 30 | 10 | 294 | 22,985.1 | 0.11 | 290 | 22,988.4 | 0.097 | 290 | **22,994.8** | 0.001 |
| - | 60 | 11 | 359 | **26,773.3** | 0.47 | 358 | 26,703.4 | 0.389 | 358 | 26,703.4 | 0.001 |
| 360 | 90 | 12 | 360 | **26,883.6** | 0.93 | 358 | 26,750 | 0.9 | 358 | 26,750 | 0.001 |
| global | 30 | 13 | 330 | **27,040.7** | 0.21 | 289 | 25,575.4 | 0.164 | 288 | 25,529.6 | 0.002 |
| - | 60 | 14 | 590 | **45,876.3** | 1.12 | 573 | 45,449.6 | 0.945 | 574 | 45,481.2 | 0.002 |
| 720 | 90 | 15 | 713 | **53,204** | 3.05 | 703 | 52,581.6 | 2.455 | 704 | 52,630 | 0.003 |
| global | 30 | 16 | 354 | **29,675.3** | 0.31 | 292 | 26,945.8 | 0.226 | 291 | 26,846.2 | 0.005 |
| - | 60 | 17 | 633 | **51,509.8** | 1.59 | 586 | 50,231.5 | 1.308 | 587 | 50,308.8 | 0.005 |
| 1080 | 90 | 18 | 892 | **69,741.8** | 4.61 | 869 | 68,907.9 | 4.247 | 866 | 68,767.1 | 0.006 |

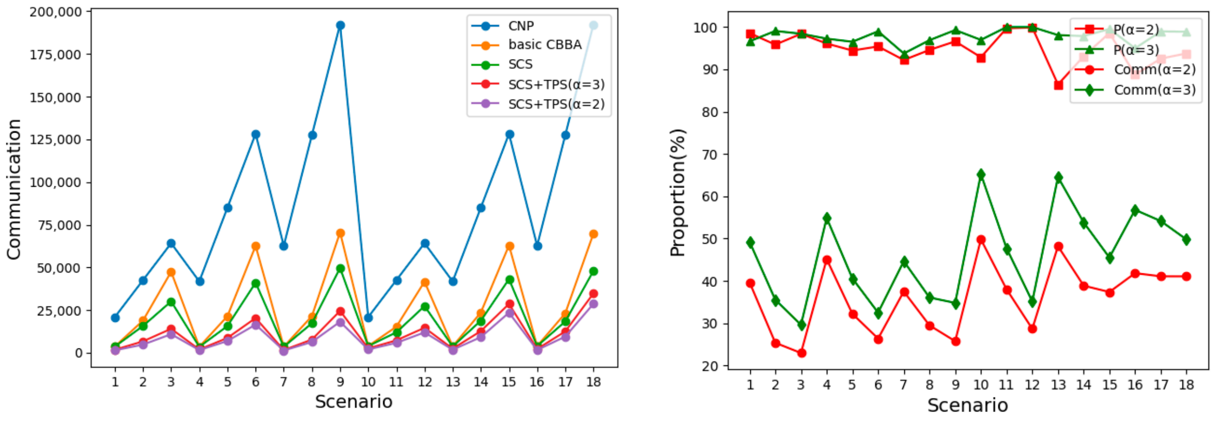
(1) As can be seen from Table 7, in most cases, c-CBBA (mix) significantly outperforms the other two algorithms in indicators such as the number of scheduled tasks and the total profit. This is because the bid value is more global when considering storage, conflicts, etc., which is conducive to achieving an overall better performance. The scheduling effect of c-CBBA (profit) is slightly better than that of CNP. In 18 scenarios, nine times of c-CBBA (profit) are better, seven times of CNP are better, and the other two have the same effect. Both have the same bidding function, but the reason for the difference in scheduling effectiveness is that CNP only bids for one task at a time, while CBBA bids for a group of tasks simultaneously. Relatively speaking, CBBA has a more global perspective. However, due to c-CBBA (profit) only considering benefits and not considering the mutual influence between tasks during actual scheduling, the results are poor.

(2) When communication time is not factored into the time cost, the CPU time required for both c-CBBA and CNP operation is not high. This indicates that the algorithm does not incur excessive computational burden and is therefore suitable for on-board operation.

The bid function of the basic CBBA and c-CBBA in the following experiment will be based on Equation (9).

### 5.2.2. Comparison of Communication Volume

The experiments tested the performance on communication volume and scheduling effect on three different configurations of c-CBBA, including separate SCS and two hybrid strategy of SCS and TPS: SCS+TPS ($\alpha = 2$) and SCS+TPS ($\alpha = 3$), simultaneously compared with CNP and the basic CBBA. Figure 10a shows the required communication volume for different algorithms in different scenarios. The calculation method is as follows: every time the satellite transmits information to a neighbor, the communication volume is increased by one. The horizontal axis represents the scenario, which corresponds to the scenario number in Table 7 one by one. The communication topology obtained by applying the SCS is based on Figure 8b.



(**a**) (**b**)

**Figure 10.** Comparison of communication volume. (**a**) Comparison of communication volume among five mechanisms. The vertical axis represents the total number of times the satellite sent information to its neighbors. (**b**) Comparison of communication volume (Comm) and scheduling effectiveness (P) of hybrid strategy. The vertical axis represents the percentage of the communication volume and profit of the hybrid strategy to the basic CBBA.

For ease of presentation, the strategy combining SCS and TPS is called the hybrid strategy. From Figure 10, we can draw the following conclusions:

(1) The communication volume of CBBA is much lower than that of the CNP. Assume that the number of satellites is $|S|$, the number of missions is $|T|$, and all satellites can communicate directly with each other. Each task in CNP involves three rounds of communication processes: master satellite bidding, child satellite tendering, and master satellite publishing the winning bid results. Considering that the master satellite combines the notification of the previous task bidding result and the sending of the next task bidding information into one communication, the required communication volume for the CNP is $(2|T| + 1) \times (|S| - 1)$. For CBBA, as each communication interaction involves information from the task bundle, the amount of information in each communication interaction is greatly increased, which can effectively reduce communication volume.

(2) Both SCS and TPS can effectively reduce communication volume. The SCS is used to remove redundant information transmission, so it does not affect the scheduling results, while TPS reduces communication volume at the expense of some scheduling effects. Figure 10b provides an in-depth analysis of the impact of the hybrid strategy on communication volume (Comm) and scheduling profit (P). The vertical axis of the figure represents the percentage of the communication volume and profit of the hybrid strategy to the basic CBBA. It can be seen that when $\alpha = 2$, the average scheduling profit reaches over 94.8% of the basic CBBA algorithm, while the average communication decreases to 36.1%. When $\alpha = 3$, the scheduling gain reaches more than 97.8% of the basic algorithm on average, while the communication drops to 46.2% on average. It can be seen that in similar scenarios in this experiment, when $\alpha = 3$, c-CBBA can significantly reduce communication volume while ensuring scheduling effectiveness.

(3) SCS and TPS strategies can effectively reduce the sensitivity of CBBA to mission scale and constellation scale. It is evident from Figure 10a that CNP and basic CBBA communication volume increases rapidly as the mission or satellite scale rises. When the scenario scale is small, the difference in communication volume between different algorithms is small. However, as the scenario scale increases, the increase in communication volume of c-CBBA is significantly weaker than that of the basic CBBA. This suggests that c-CBBA has better adaptability to large-scale scenarios.

## 6. Discussion

From the experiments in the previous section, it can be seen that c-CBBA can effectively reduce communication volume while ensuring scheduling quality. In order to better understand the algorithm, this section further analyzes the operational mechanism of c-CBBA and the effectiveness of the two strategies in different scenarios.

### *6.1. Parameter Testing of TPS*

The effect of TPS is influenced by parameter $\alpha$. This experiment tests two scenarios, local-1080-90 and global-1080-90, as examples. The algorithm results are shown in Table 8 (local) and Table 9 (global). As parameter $\alpha$ changes, the relationship between communication cost and global total benefit is shown in Figure 11.

**Table 8.** Parameter testing of local-1080-90 scenario. TS refers to the number of tasks successfully scheduled; TP refers to the total profit. Comm (%) represents the percentage of communication volume using TPS compared to the basic CBBA. P (%) represents the percentage of profit using TPS compared to the basic CBBA.

| $\alpha$ | Iterations | Comm Volume | Comm (%) | TS | TP | P (%) |
|---|---|---|---|---|---|---|
| 0 * | 55 | 70,533 | 1 | 765 | 59,501.6 | 1 |
| 1 | 14 | 15,671 | 22.22 | 593 | 47,296.6 | 79.49 |
| 2 | 22 | 27,120 | 38.45 | 732 | 56,987.2 | 95.77 |
| 3 | 30 | 37,602 | 53.31 | 773 | 59,762.6 | 100.44 |
| 4 | 35 | 44,228 | 62.71 | 767 | 59,311.4 | 99.68 |
| 5 | 41 | 52,124 | 73.90 | 773 | 59,814.5 | 100.53 |
| 6 | 42 | 53,425 | 75.74 | 770 | 59,718.6 | 100.36 |
| 7 | 42 | 53,409 | 75.72 | 773 | 59,962.9 | 100.78 |
| 8 | 49 | 62,637 | 88.81 | 772 | 59,799.1 | 100.50 |
| 9 | 49 | 62,637 | 88.81 | 774 | 59,895.3 | 100.66 |

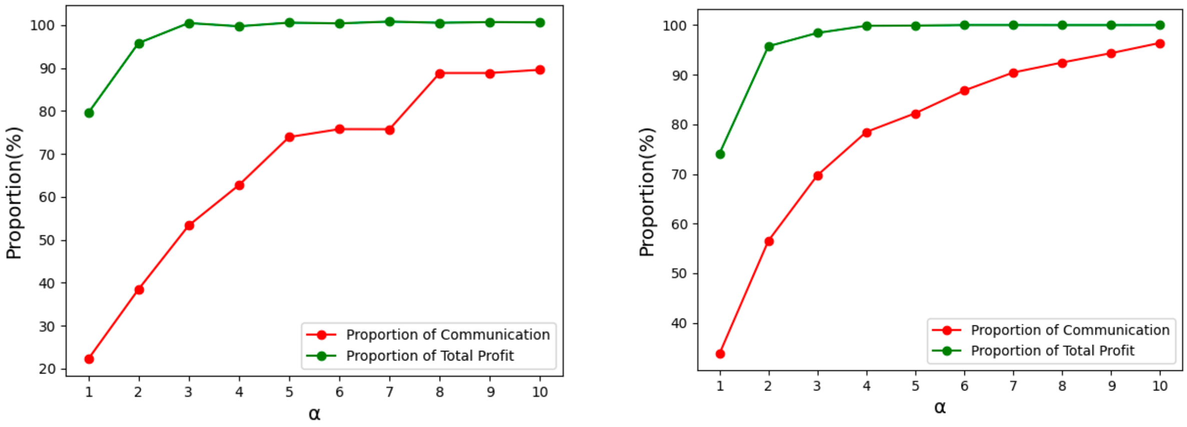
* The basic CBBA.

**Table 9.** Parameter testing of global-1080-90 scenario. TS refers to the number of tasks successfully scheduled; TP refers to the total profit. Comm (%) represents the percentage of communication volume using TPS compared to the basic CBBA. P (%) represents the percentage of profit using TPS compared to the basic CBBA.

| $\alpha$ | Iterations | Comm Volume | Comm (%) | TS | TP | P (%) |
|---|---|---|---|---|---|---|
| 0 * | 54 | 69,843 | 1 | 892 | 69,741.8 | 1 |
| 1 | 20 | 23,567 | 33.74 | 635 | 51,651.8 | 74.06 |
| 2 | 32 | 39,481 | 56.53 | 846 | 66,763.1 | 95.73 |
| 3 | 38 | 48,698 | 69.72 | 874 | 68,634.8 | 98.41 |
| 4 | 43 | 54,802 | 78.46 | 891 | 69,642.6 | 99.86 |
| 5 | 45 | 57,434 | 82.23 | 891 | 69,673.8 | 99.90 |
| 6 | 47 | 60,631 | 86.81 | 892 | 69,741.8 | 100.00 |
| 7 | 49 | 63,168 | 90.44 | 892 | 69,741.8 | 100.00 |
| 8 | 50 | 64,579 | 92.46 | 892 | 69,732.2 | 99.99 |
| 9 | 51 | 65,889 | 94.34 | 892 | 69,732.2 | 99.99 |

* The basic CBBA.

The basic CBBA algorithm determines the winning satellite for one task after global communication. For each task's bidding process, it belongs to the global optimal strategy. TPS is essentially a local optimal choice to preempt a task when the satellite still believes it has the highest bid after communicating with neighbors $\alpha$ times in a row. $\alpha$ represents the "local" range, and the larger it is, the more times the satellite communicates with its neighbors, the more information it obtains, and the closer its scheduling results are to CBBA. The purpose of TPS is to significantly reduce communication volume at the expense of a small amount of profit. From the experiment, it can be seen that the local-1080-90 and global-1080-90 scenarios exhibit similar trends. When $\alpha = 1$, the communication volume significantly decreased, accounting for 22.22% and 33.74% of the basic CBBA, respectively. However, there was also a certain degree of decrease in revenue, accounting for 79.49% and 74.06% of the basic CBBA, respectively. In addition, when $\alpha = 2$, the revenue both reach over 95% of the basic CBBA, and the communication volume is 38.45% and 56.53% of the basic CBBA, respectively. As α continues to increase, the scheduling profit is almost equal to that of the basic CBBA, while the communication volume is significantly lower. Therefore, the TPS policy can significantly reduce the communication volume while effectively guaranteeing the scheduling quality.



(**a**) (**b**)

**Figure 11.** Parameter α testing of TPS. The vertical axis represents the percentage of the communication volume and profit of c-CBBA (using TPS policy with different parameter α) to the basic CBBA. (**a**) Example of local distribution; (**b**) example of global distribution.

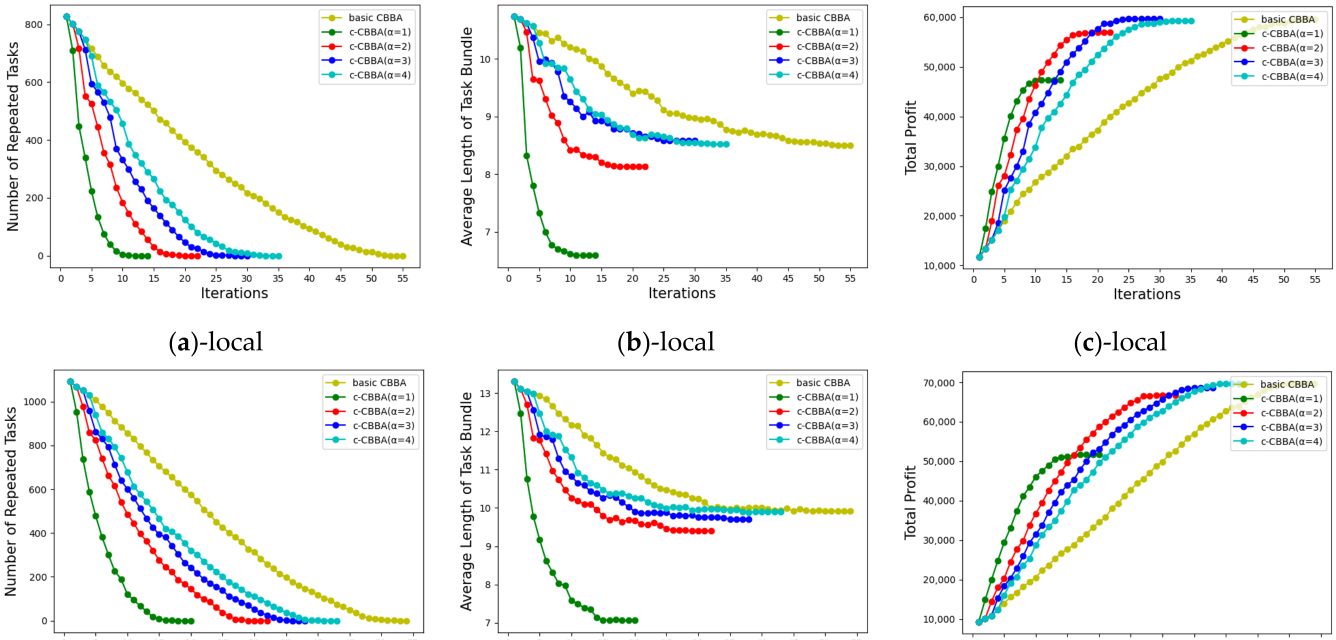
### *6.2. Tests for Algorithm Convergence*

As the algorithm continues to iterate, auction information spreads globally, and satellites continuously update their local cognition, gradually achieving global consistency, and then c-CBBA tends to converge. This experiment is used to analyze the changes in various indicators during the convergence process of c-CBBA, where c-CBBA only uses the TPS mechanism.

This experiment takes two scenarios, local-1080-90 and global-1080-90, as examples to test the convergence from three perspectives: (a) number of repeated tasks in all task bundles, (b) the average length of task bundles, and (c) the total profit. The results are shown in Figure 12.

From Figure 12, it can be seen that the trend of the above indicators with iterations is roughly the same in global and local distribution scenarios. The global repeated tasks in all task bundles are the most intuitive indicator for testing convergence. The statistical method of this indicator is: if the task bundles of satellite $s_1$, $s_2$, and $s_3$ all contain task $t_j$, the number of global repeated tasks will be increased by 2. Figure 12a shows that this indicator continues to decrease during the iteration process, dropping to 0 at convergence. When using TPS, the convergence speed is significantly accelerated; the smaller the setting of α, the faster the convergence speed.

Indicator (b) also shows a downward trend. This is because satellites have less understanding of global auction information in the early stages of algorithm operation, and greedily add tasks to their own task bundles, resulting in longer average task bundle. With the iteration, the satellites' understanding of the global bidding information deepens, and there are more and more tasks that cannot be added to the bundle set due to their own bids being lower than other satellites, coupled with the constraints of the algorithms, which gradually reduces the number of tasks in the task bundles. This indicator also reflects the planning effect to some extent. The longer the average length at convergence, the better the planning effect. When α is small, the indicator is poor, and as α increases, the indicator gradually approaches the basic CBBA.



(**a**)-local (**b**)-local (**c**)-local

(**a**)-global (**b**)-global (**c**)-global

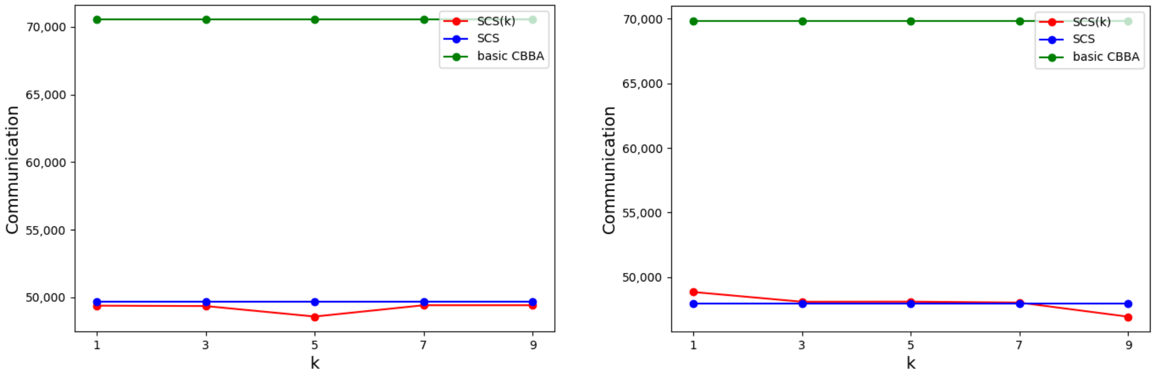
**Figure 12.** Changes in various indicators during algorithm convergence. (**a**) Indicator of the number of repeated tasks; (**b**) indicator of the average task bundle length; (**c**) indicator of the total profit. The suffix local and global refers to the task distribution in the example being local or global.

Indicator (c) considers the repeated execution of tasks as a non-profit observation, and only calculates the profit once for repeated tasks. In the early stage of algorithm, due to a large number of tasks being repeatedly added to the task bundles, the global total profit is lower with limited resources, so this indicator shows the opposite trend compared to the first two indicators.

### *6.3. Robustness Testing of SCS*

The single-chain strategy means that when more than one neighbor exists on one side of the satellite in constellation, one of them is selected to build the link. The above experiments all use the closest distance as the selection rule. This experiment tests the effect of the single-chain strategy on the convergence and the communication volume of the algorithm when the communication link is partially damaged, requiring the construction of the path as shown in Figure 8c. Additionally, it provides a comparison with the c-CBBA (SCS) algorithm utilizing the communication topology shown in Figure 8b, as well as the basic CBBA. The algorithm in a partially damaged communication environment is represented by SCS (k), where k represents the number of damaged communication links. Construct this environment by randomly deleting adjacent communication links. Figure 13a,b show the test results in local-1080-90 and global-1080-90 scenarios, respectively.

As can be seen from Figure 13, the SCS policy in Figure 8c can still converge quickly and will not have a significant impact on communication volume. That is, the SCS can effectively reduce the communication volume when using different communication topology. This indicates that this strategy does not reduce system robustness due to pruning communication links, and it still has high adaptability to the partially damaged communication environment.



(**a**) (**b**)

**Figure 13.** Robustness test results of SCS policy. SCS (k) denotes the communication volume generated by the algorithm in a partially compromised communication environment, where k represents the number of damaged communication links. (**a**) local-1080-90 scenario; (**b**) global-1080-90 scenario.

## 7. Conclusions

In this paper, we study the multi-satellite autonomous imaging mission planning problem and propose the fully distributed algorithm c-CBBA. The algorithm is improved on the basis of the basic CBBA, mainly addressing its drawbacks of high communication volume. The Task Preemption Strategy (TPS) and Single-Chain Strategy (SCS) are proposed to improve the operation efficiency of the algorithm during the actual on-board scheduling. By analyzing the operation process of CBBA under the background of satellite imaging mission planning, TPS, an acceleration convergence mechanism, is designed to address the hidden drawbacks in its convergence process. The SCS policy is used to prune communication links that have redundant information interactions during satellite communication. The experimental results show that c-CBBA can achieve an average communication volume reduction of 46.2% of the original algorithm while ensuring that the average profit is not less than 97.8%, effectively improving the actual operational efficiency of the algorithm. In further work, the application of c-CBBA to large-scale scenarios will be explored. Task allocation modes such as dividing satellite clusters and multi-stage allocation will be investigated, as well as designing the operation mechanism of c-CBBA under the corresponding modes.

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

**Funding:** This research was funded by National Natural Science Foundation of China, grant number 72001212; and Young Elite Scientists Sponsorship Program by CAST, grant number 2022QNRC001.

**Data Availability Statement:** not applicable.

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

## References

1. Xie, P.; Du, Y.; Yao, F.; Tan, Y. Literature Review for Autonomous Scheduling Technology of Agile Earth Observation Satellites. *J. Astronaut.* **2019**, *40*, 127–138.
2. Fang, F.; Wu, M. Research on the Development of Global Low Earth Orbit Satellite Constellations. *Aerodyn. Missile J.* **2020**, *50*, 88–92.
3. Ke, Z.; Huang, S.; Li, Y.; Qiao, K.; Wang, X. Research on the Development Status and Key Technologies of Large LEO Remote Sensing Constellations. *Spacecr. Recovery Remote Sens.* **2023**, *44*, 93–101.

4. Wang, P. Research on Branch-and-Price Based Multi-satellite Multi-station Integrated Scheduling Method. Doctoral Dissertation, National University of Defense Technology, Changsha, China, 2011.
5. Bai, B.; He, R.; Li, J.; Chen, Y. Satellite orbit task merging problem and its dynamic programming algorithm. *Syst. Eng. Electron.* **2009**, *31*, 1738–1742.
6. Zhao, P.; Chen, Z. An adapted genetic algorithm applied to satellite autonomous task scheduling. *Chin. Space Sci. Technol.* **2016**, 47–54. [CrossRef]
7. Liu, X.; Laporte, G.; Chen, Y.; He, R. An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time. Comput. *Oper. Res.* **2017**, *86*, 41–53.
8. He, L.; Liu, X.; Laporte, G.; Chen, Y.; Chen, Y. An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling. *Comput. Oper. Res.* **2018**, *100*, 12–25.
9. Lemaıtre, M.; Verfaillie, G.; Jouhaud, F.; Lachiver, J.-M.; Bataille, N. Selecting and scheduling observations of agile satellites. *Aerosp. Sci. Technol.* **2002**, *6*, 367–381.
10. Cichy, B.; Chien, S.; Rabideau, G.; Tran, D. Validating the autonomous EO-1 science agent. In Proceedings of the International Workshop on Planning & Scheduling for Space, Darmstadt, Germany, 22–25 June 2004; p. 39.
11. Truszkowski, W.; Hallock, H.; Rouff, C.; Karlin, J.; Rash, J.; Hinchey, M.; Sterritt, R. *Autonomous and Autonomic Systems: With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems*; Springer: Berlin/Heidelberg, Germany, 2014; p. 2139.
12. Wang, C.; Tang, J.; Cheng, X.; Liu, Y.; Wang, C. Distributed cooperative task planning algorithm for multiple satellites in delayed communication environment. *J. Syst. Eng. Electron.* **2016**, *27*, 619–633.
13. Zhang, K.; Sun, Y.; Xia, L.; Zhu, Z.; Wang, J. A method of network satellite on-orbit distributed collaborative mission scheduling. *J. Harbin Eng. Univ.* **2019**, *40*, 393–399.
14. Smith, R.G. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE T. Comput.* **1980**, *C-29*, 1104–1113.
15. Zhang, Z. Research on Mission Planning and Control Problem for Distributed Imaging Satellite System Based on MAS. Doctoral Dissertation, National University of Defense Technology, Changsha, China, 2006.
16. Long, J.; Qian, Z.; Xie, F.; Ding, Z.; Liu, L. An Improved Multi-Satellite Cooperative Task Planning Method Based on Distributed Multi-agent System. In Proceedings of the 2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Beihai, China, 16–17 January 2021.
17. Zheng, Z.; Guo, J.; Gill, E. Onboard mission allocation for multi-satellite system in limited communication environment. *Aerosp. Sci. Technol.* **2018**, *79*, 174–186.
18. Gao, T.; Hu, X.; Xia, W. Constellation autonomous mission planning algorithm based on distributed co-evolution. *Syst. Eng. Electron.* **2022**, *44*, 1600–1608.
19. Zheng, Z.; Guo, J.; Gill, E. Distributed onboard mission planning for multi-satellite systems. *Aerosp. Sci. Technol.* **2019**, *89*, 111–122.
20. Choi, H.; Brunet, L.; How, J.P. Consensus-Based Decentralized Auctions for Robust Task Allocation. *IEEE Trans. Robot.* **2009**, *25*, 912–926.
21. Song, B.; Chen, Y.; Yang, Q.; Zuo, Y.; Xu, S.; Chen, Y. On-Board Decentralized Observation Planning for LEO Satellite Constellations. *Algorithms* **2023**, *16*, 114.
22. Matricciani, E. Geocentric Spherical Surfaces Emulating the Geostationary Orbit at Any Latitude with Zenith Links. *Future Internet* **2020**, *12*, 16.
23. Zhao, Q.; Hu, Z.; Chen, C.; Gong, Z.; Song, G.; Li, M. Opportunities and Challenges of Large-scale LEO Constellation. *Space Debris Res.* **2020**, *20*, 1–9.

**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.