Article

Multi-Granularity Mission Negotiation for a Decentralized Remote Sensing Satellite Cluster

Xuelel Deng , Yunfeng Dong * and Shucong Xie

School of Astronautics, Beihang University, Beijing 100191, China; dengxuelel@buaa.edu.cn (X.D.); xieshucong@buaa.edu.cn (S.X.)

* Correspondence: sinosat@buaa.edu.cn; Tel.: +86-1330-102-0700

Received: 2 October 2020; Accepted: 29 October 2020; Published: 2 November 2020

Abstract: Satellite remote sensing is developing towards the micro-satellite cluster, which brings new challenges to mission assignment and planning for the cluster. A multi-agent system (MAS) is used, but the time delay caused by communication and computation is rarely considered. To solve the problem, a neural-network-based multi-granularity negotiation method under decentralized architecture is proposed. Firstly, we divided negotiation into three levels of granularity, and they work in different modes. Secondly, a neural network was trained to help the satellite select the best level in real-time. Through experiments, we compared the satellites working in three different levels of granularity, in which a multi-granularity decision was used. As a result of our experiments, a lower cost-effectiveness ratio was obtained, which proved that the multi-granularity negotiation method proposed in this paper is practical.

Keywords: satellite cluster; mission assignment and planning; decentralized; multi-granularity negotiation

1. Introduction

Satellite remote sensing aims to obtain information from the Earth's surface. It has been widely used in geography, Earth science, meteorology, military, etc. [1]. In traditional imaging missions, a single low Earth orbit satellite is often used to take images of multiple targets multiple times. However, the increasing number of missions and the higher time resolution requirements call for more satellite members [2]. Besides, different missions require different kinds of payload, but multiple payloads can hardly be carried by one satellite. Furthermore, multiple payloads can hardly work together at the same time in the narrow imaging time window. Therefore, current remote sensing missions rely more and more on large satellite clusters, which brings new challenges to mission assignment and planning [3].

A multi-agent system (MAS) is the main solution to the distributed autonomy problem for multiple satellites. Agent technology was first used in DeepSpace-1 (DS-1) [4,5]. Although MAS applied in DS-1 is a single satellite system, with its success the MAS started to be widely used in satellite cluster task planning. Campbell [6–8] summarized the work of predecessors, and defined four different working modes of satellite cluster, including traditional mode, top-down mode, centralized mode, and distribute mode. The traditional mode and the top-down mode were used in the early satellite clusters. For instance, He [9] proposed an edge computing framework including a central node and several edge nodes using a constructive heuristic algorithm based on the density of residual tasks. Although these algorithms have achieved good results, the central node (ground station or central master satellite) works very hard.

Current research mainly focuses on the latter two modes: the centralized mode and the distributed mode. Cheng [10] developed a satellite cluster negotiation and mission assignment method by using contract net in the centralized mode. He proposed three negotiation models: acquaintance's trust-based

announcing bidding, adaptive bidding with swarm intelligence, and a multi-attribute decision-based fuzzy evaluation bidding method. Wang [11] discussed the cooperative multicenter problem on satellite imaging scheduling and proposed a cooperative co-evolutionary algorithm, and then proposed a novel fixed-length binary encoding mechanism for missions assignment. Iacopino [12] designed a self-organizing multi-agent structure belong to the distribution mode. The system can adapt to the change of the problems and can synchronize the satellites' plans in order to avoid duplication. In order to improve cluster consistency, Zhao [13] designed a distributed time consensus protocol for multi-agent systems with general linear dynamics on a directed graph. Buzzi [14] proposed an agent-based framework to study the self-organization of the satellite cluster, which can establish a temporary alliance to change the modularity of the system.

Since a remote sensing mission is usually divided into multiple stages (discovery, identification, confirmation, tracking, and monitoring) and one stage can further be divided into multiple steps (collection, transmission, process, etc.), a micro-satellite is not able to finish a mission all by itself. Current research has begun to focus on the cooperation of satellite clusters toward finishing one complex mission instead of merely using one satellite. Globus [15] used multiple variants of the genetic algorithm, hill climbing, simulated annealing, squeaky wheel optimization, and iteration to realize the coordination of multiple satellites and multiple observations of the same target. Araguz [16] designed autonomous operational schemes for Earth-observing swarms of nano-satellites, in which all satellites cooperate to observe multiple targets. Gallud [17] presented an agent-based simulation framework. In his study, satellites could work together to perform a set of observational tasks. His framework supported allocation targets for satellites and collective sensing of a target for multi-satellites. He [18] considered three important sections of planning (store sequence of tasks, action sequence, and the satellite status) to deal with scientific event discovery, satellite faults, cloud obscuration, and emergencies.

The main problem of the remote sensing mission lies in the conflict between resource consumption and mission accomplishment. The researchers considered many factors. Nag Sreeja [19] proposed a modular framework that combines orbital mechanics, attitude control, and scheduling optimization working on Cubesats. Wang [20] discussed the impact of clouds in the case of joint probabilities, and established a sample average approximation (SAA) model accordingly. Wang [21] also proposed a branch and cut algorithm based on lazy constraint generation. Although these kinds of limited resources onboard have been considered, the influence of time delay on negotiation is less related to that. The time delay is mainly caused by limited communication ability and computational resources. When a satellite keeps tracking a target, it has to act out a continuous attitude maneuver which affects the pointing accuracy of its antenna, resulting in a decrease of the communication quality. Besides, tracking a target produces a large number of images, the process of which is heavy work for a single micro-satellite. A common way is to transmit those images to other micro-satellites. However, image transmission occupies a large amount of communication bandwidth, so that less communication bandwidth can be left for negotiation. Lastly, the image process itself is computationally expensive, so that a micro-satellite can hardly negotiate with others when processing images. In conclusion, a large time delay may cause the delay of negotiation and may finally lead to negotiation failure.

In this paper, we propose a neural-network-based multi-granularity negotiation method under decentralized architecture belonging to distributed mode in [6–8]. Firstly, we propose a multi-granularity negotiation model. The negotiation process is divided into three levels of granularity: low-level granularity, middle-level granularity, and high-level granularity. In each granularity, satellites transmit different kinds of information and use different negotiation processes and assignment algorithms. Secondly, we establish a neural network to intelligently select the appropriate level of granularity. The neural network could output the success probabilities of negotiation in each level of granularity. Then a simple decision mode based on the probabilities is used to select the level in real-time.

The remainder of the paper is organized as follows. In Section 2, the modeling process of the multi-granularity negotiation method is introduced considering three aspects: mission preprocesses for all levels of granularity, multi-granularity partitioning of the multi-agent negotiation model, and intelligent real-time level selection of negotiation granularity. In Section 3, experiments to verify this method in different conditions are described. The proposed method is discussed in Section 4. Finally, the conclusions of the study are given in Section 5.

2. Proposed Methods

To finish negotiation with limited communication resources, a multi-granularity negotiation method was developed. Firstly, the traditional multi-agent negotiation model is divided into three levels of granularity: low-level granularity, middle-level granularity, and high-level granularity. Mission selection algorithms were designed for each level of granularity. A fixed consistency algorithm is used to realize the decentralization of the satellite cluster. After that, we designed different negotiation processes and consistency mechanisms to simplify the negotiation process of different granularity. Secondly, different samples on different target numbers, bandwidths and transmit hop numbers were collected to train a neural network. Finally, the best neural network was established to select the best granularity onboard. The flowchart of the multi-granularity negotiation method proposed in this paper is illustrated in Figure 1.

2.1. Mission Preprocess for All Levels of Granularity

2.1.1. Time Window Calculation

Suppose that there are n targets, the set of all targets can be defined as T :

$$T = \{t_1, \dots, t_i, \dots, t_n\}, i \in N^+, 1 \leq i \leq n, \quad (1)$$

where t_i expresses the information of the i th target, including latitude, longitude, height, and motion characteristics. Motion characteristics contain course, angular velocity, speed, and acceleration. All these motion characteristics can be unknown.

The calculation of the time window is to find mission start time T_i^S and mission end time T_i^E to make sure that for $\forall t \in [T_i^S, T_i^E]$, two constraints are satisfied: the optical visibility constraint and the geometric visibility constraint. The optical visibility constraint for the i th target $C_{o_i}(t)$ can be expressed as:

$$C_{o_i}(t) = \begin{cases} 1 & \mathbf{R}_i \cdot \mathbf{R}_{Sun} > 0 \\ 0 & \mathbf{R}_i \cdot \mathbf{R}_{Sun} \leq 0 \end{cases}, \quad (2)$$

where $\mathbf{R}_i$ is the position vector of t_i in the inertial coordinate, which can be calculated by integrating with time. $\mathbf{R}_{Sun}$ is the position vector of the sun in inertial coordinate. The geometric visibility constraint for i th target $C_{g_i}(t)$ can be expressed as:

$$C_{g_i}(t) = \begin{cases} 1 & \alpha \leq \alpha_{max} \text{ and } \theta \leq \theta_{max} \\ 0 & \text{else} \end{cases}, \quad (3)$$

where α is the angle between the target's position vector $\mathbf{R}_i$ and its projection vector $\hat{\mathbf{R}}_i$ in the orbital plane of the satellite, which can be expressed as:

$$\alpha = \arccos\left(\frac{\mathbf{R}_i \cdot \hat{\mathbf{R}}_i}{|\mathbf{R}_i| \cdot |\hat{\mathbf{R}}_i|}\right). \quad (4)$$

α_{max} is the maximum value of α , expressed as:

$$\alpha_{max} = \arccos\left(\frac{|\mathbf{R}_i|}{|\mathbf{R}|}\right), \quad (5)$$

where $\mathbf{R}$ is satellite's position vector in inertial coordinate. θ_{max} in Equation (3) is the maximum attitude maneuver angle of the satellite. The desired attitude maneuver angle θ is expressed as:

$$\theta = \arctan\left(\frac{|\mathbf{R}_i| \cdot \sin(\alpha)}{|\mathbf{R}| - |\mathbf{R}_i| \cdot \cos(\alpha)}\right). \quad (6)$$

Figure 1. The flowchart of the multi-granularity negotiation method.

2.1.2. Profit Calculation

One remote sensing mission can be divided into five stages: discovery, identification, confirmation, tracking, and monitoring. Different initial profits p_j^{ini} are defined as constants for each stage of the mission in this paper, as is shown in Table 1.

Table 1. Initial profits p_j^{ini} for each stage of one mission.

Stages	Discovery	Identification	Confirmation	Tracking	Monitoring
profit	0.25	0.1	0.15	0.25	0.25

Mission profit $P_{i,j}$ can be expressed as:

$$P_{i,j} = A_{t_i} \times A_{p_i} \times p_j^{ini}, \quad (7)$$

where i is the index of the target. j is the index of the stage, and $j \in N^+, 1 \leq j \leq 5$. A_{t_i} is a constant coefficient varying with the type of the target. A_{p_i} is also a constant coefficient varying with the position of the target. In this model, it is easy to find out that the profit is only related to the targets.

2.1.3. Cost Calculation

Mission cost consists of three parts: time cost, memory cost, and power cost. The time cost c_{t_i} is defined as:

$$c_t = \begin{cases} 10 \cdot \log_{10} \frac{T_i^S}{100} & T_i^S > 100 \\ 0 & T_i^S \leq 100 \end{cases}, \quad (8)$$

where T_i^S is the start time of time window of target i .

The mission duration $t_{d_{i,j}}$ corresponds to the i th target and j th stage of the mission, expressed as:

$$t_{d_{i,j}} = \begin{cases} 2 & j \leq 2 \\ T_i^E - T_i^S & j > 2 \end{cases}. \quad (9)$$

Memory cost per second is considered as a constant corresponding to the camera, written as MEM. The memory cost c_m changing with mission duration can be expressed as

$$c_m = MEM \times t_{d_{i,j}}. \quad (10)$$

Similar to memory cost, average satellite power is also considered as a constant: POW. The power cost c_p also changes with mission duration, expressed as

$$c_p = POW \times t_{d_{i,j}}. \quad (11)$$

The total cost is the sum of these three costs, i.e., the cost for i th target during its j th stage, is $c_{i,j} = c_t + c_m + c_p$.

2.2. Multi-Granularity Partition of the Multi-Agent Negotiation Model

The framework in this paper is decentralization. In the blockchain technology, there are several consistency algorithms, including RAFT [22], PBFT [23], DPOS [24], RIPPLE [24], POW [25], POS [26], etc. We applied RAFT, PBFT, and RIPPLE in this paper. In RAFT, there are three roles of leader, candidate, and follower. At the start, all nodes in the net are followers. Then some nodes become candidates and initiate a vote. After voting, only one leader is elected, who manages the net by synchronizing logs and connects all followers by heartbeat. Followers only replicate the log. Differently from RAFT, PBFT introduces a result verification mechanism. It can avoid a byzantine fault by

using the processes of REQUEST, PRE-PREPARE, PREPARE, COMMIT, and REPLY. In RIPPLE, more leaders are elected. The consistent process is divided into two parts: consistency in leader group and global consistency.

There are three roles of satellites set in this system: leader node (*Leader*), temporary leader node (*TLeader*), and follower node (*Follower*). The *Leader* is elected to organize the negotiation. The *Leader* periodically processes the mission requests from *Follower* and releases negotiation results to *Followers*. The *Leader* establishes a leader group consisting of itself and *TLeaders*. They firstly reach an agreement in the group by group vote. After being verified by *TLeaders* the negotiation result can be sent to *Followers*. *Followers* originate missions themselves and negotiate with each other under the *Leader's* control.

2.2.1. Low-Level Granularity

In low-level granularity, the satellites are allowed to negotiate only once to save communication resources. The negotiation process can be described as the following steps (the *TLeaders's* assistance process is ignored in this granularity and is introduced in high-Level granularity) shown in Figure 2:

- **Step 1:** Preprocessing the missions. The *Follower* first preprocesses the missions by calculating the time window, profit, and cost.
- **Step 2:** Select a set of desired mission stages. The desired mission stages are selected by decision algorithm in *Follower*.
- **Step 3:** Transmit the mission set. The set of desired mission stages MS_{desire} and their cost is broadcast, and the *Leader* records the mission set.
- **Step 4:** Leader process. The *Leader* finishes the mission assignment using a greedy algorithm. According to the profit calculation model, the profits of each mission stage are the same. That means the mission stages are assigned to the satellite with less cost.
- **Step 5:** Announce temporary result. The *Leader* announces a temporary result to all *Followers*.
- **Step 6:** Result check and national vote. The *Follower* logs and checks the result. If the assigned missions MS_{actual} are all in the desired mission set submitted, the *Follower* votes in favor, and if not votes against. The national vote result can be expressed as:

$$V = \begin{cases} 1 & MS_{actual} \subseteq MS_{desire} \\ 0 & else \end{cases} \quad (12)$$

- **Step 7:** Result release. The *Leader* records the selection from the *Follower* who votes in favor. Then the *Leader* releases the final negotiation result.
- **Step 8:** Execute. The *Follower* checks the result again, then picks the intersection of the set of assigned missions MS_{actual} and the set of desired mission stages MS_{desire} , and then executes the mission stages in the intersection.

Under the process, each *Follower* needs to decide the desired mission set by itself. It becomes a group knapsack problem with time constraint, which can be described as:

$$\begin{aligned} & \max \sum_{i=1}^n \sum_{j=1}^5 P_{i,j} s_{i,j} \\ & S.T. \sum_{j=1}^5 s_{i,j} < 2 \\ & \quad \forall i' \text{ and } i'', T_{i'}^S > T_{i''}^E \text{ OR } T_{i''}^S > T_{i'}^E, \end{aligned} \quad (13)$$

where $s_{i,j}$ is mission selection result, $s_{i,j} = 1$ if the j th stage of i th target is selected; else $s_{i,j} = 0$. Both of i' and i'' are the target ID in the selected mission set. Dynamic programming is chosen to solve the problem.

Figure 2. Low-level granularity negotiation process.

In the low-level granularity, the communication resource is most significant. Due to the RAFT algorithm's low consumption, it was selected to be the consistency algorithm work at this level. The algorithm's parameters are adjusted to space usage. The modifications include:

Heartbeat interval is longer. Due to the time-varying characteristics of satellite networks, the connection may be unstable. Frequent broadcast behavior for heartbeat may cause wastage of bandwidth. Additionally, the network delay may cause misjudgment of the connection between the *Leader* and *Followers*. The longer interval brings a bigger allowance.

The election interval is longer. Based on the principle of RAFT, the election happens when the *Leader* is unconnected. That means the election interval is related to the heartbeat interval. Therefore, the election interval is longer too.

The negotiation result is not released in a heartbeat message. Based on the principle of RAFT, a longer heartbeat interval causes a delay of mission response. Thus the release negotiation result package is designed to be independent. It usually transmits periodically; besides, it also supports emergency transmission to improve real-time performance for emergency missions.

2.2.2. Middle-Level Granularity Model

Middle-level granularity corresponding to a normal communication environment is designed to balance bandwidth usage and the negotiation effect.

Similar to low-level granularity, the satellites are divided into one *Leader* and several *Followers*. The process can be divided into the following steps shown in Figure 3:

- **Step 1:** Preprocessing the missions. Same as the **Step 1** in low-level granularity.
- **Step 2:** Select a set of mission stages. Differently from the **Step 2** in low-level granularity, the set of mission stages contains both desired and candidate mission stages.
- **Step 3:** Transmit the mission set. Differently from the **Step 3** in low-level granularity, these mission stages broadcast include their execution costs and time windows, and the *Leader* records the set.
- **Step 4:** Leader process. Same as **Step 4** in low-level granularity.

- **Step 5:** Announce temporary result. Same as **Step 5** in low-level granularity
- **Step 6:** Result check and vote. The *Followers* working in the middle-level granularity are responsible for their own mission set. The assigned missions MS_{actual} must all be in the submitted mission set. The total cost of the mission set is less than satellite's capability c_{max} . Besides, the assigned mission set must contain enough missions. The *Follower* votes in favor if the above two constraints are satisfied. If not, the *Follower* votes against. The vote result V is expressed as:

$$V = \begin{cases} 1 & MS_{actual} \subseteq MS_{desire} \text{ and } c_{max}/2 \leq \sum c_{i,j} \leq c_{max} \\ 0 & \text{else} \end{cases} \quad (14)$$

- **Step 7:** Vote process. If all *Followers* vote in favor or number of negotiation failures is too high, the *Leader* stops negotiation and goes to **Step 8**. If not, the *Leader* adjusts the set of mission stages while still using a greedy algorithm, and then goes to **Step 5**.
- **Step 8:** Result release. The *Leader* releases the final negotiation result.
- **Step 9:** Execute. Same as the **Step 8** in low-level granularity.

Figure 3. Middle-level granularity negotiation process.

In this process, the *Follower* chooses some spare mission stages. Thus, the time constraint no longer fits for this process. It becomes a group knapsack problem, which can be described as:

$$\begin{aligned} \max \quad & \sum_{i=1}^n \sum_{j=1}^5 P_{i,j} s_{i,j} \\ \text{S.T.} \quad & \sum_{j=1}^5 s_{i,j} < 2. \end{aligned} \quad (15)$$

Dynamic programming is still chosen to solve the problem.

In the middle-level granularity, the RAFT algorithm is still valid. The usage is the same as the consistency algorithm in low-level granularity.

2.2.3. High-Level Granularity Model

High-level granularity is designed to obtain a better negotiation effect in a good communication environment. Differently from the other two levels of granularity, the satellites are divided into a *Leader* group and several *Followers*, and the *Leader* group consists of a *Leader* and *TLeaders*. The process can be divided into the following steps shown in Figure 4:

- **Step 1:** Apply to join the *Leader* group. There exists one *Leader* in the system as the *Leader* for all levels of granularity. In order to increase the fault tolerance rate, the *Follower* with a direct connection (only one hop to the *Leader*) applies to join the *Leader* group. After being authorized by the *Leader*, it becomes a *TLeader*.
- **Step 2:** Preprocessing the missions. Same as before.
- **Step 3:** Select a set of mission stages. The set of mission stages, which contains all the supported mission stages, is selected by decision algorithm in *Followers* and *TLeaders*.
- **Step 4:** Transmit the mission set. These mission stages are broadcast, including their execution cost and time window, and both the *Leader* and *TLeaders* record the set.
- **Step 5:** Leader Process. The *Leader* and *TLeaders* assign missions using a greedy algorithm, considering an extra constraint of the maximum cost of each *Follower*.
- **Step 6:** Announce temporary result. The *Leader* firstly announces a temporary result in the *Leader* group and votes for the result. Then the temporary result is announced if confirmed by most *TLeaders*. If not they go to **Step 5** to process again.
- **Step 7:** Result check and vote. Same as before.
- **Step 8:** Vote process. If all *Followers* vote in favor or negotiation fails times too much, the *Leader* stops negotiations and goes to **Step 9**. If not, the *Leader* adjusts the set of mission stages still using a greedy algorithm and then goes to **Step 6**.
- **Step 9:** Result release. Same as **Step 6**, the *Leader* releases the final negotiation result first in the leader group. After being confirmed, the negotiation results are released to all nodes.
- **Step 10:** Execute. Same as before.

In this process, the nodes chose as many mission stages as possible. All mission stages satisfied cost effectiveness ratio are chosen, described as:

$$s_{i,j} = \begin{cases} 1 & c_{i,j}/P_{i,j} \leq CER^{thd} \\ 0 & c_{i,j}/P_{i,j} > CER^{thd} \end{cases} \quad (16)$$

In the high-level granularity, RIPPLE and PBFT algorithms are combined to improve system stability and negotiation accuracy. The modifications include:

Leader group member selection method. In RIPPLE algorithm, the *Leader* group member is selected from all *Followers*. Considering the communication environment, all *Followers* satisfying a condition should make applications for the group.

The PBFT verification mechanism is introduced into the leader group. The PBFT verification mechanism could effectively prevent node failure or disturbance. This is well adapted to satellites because of their limited computation resources. Verification in the leader group can reduce the possibility of error information release. In other words, the leader group sacrifices its communication bandwidth and computational resources for other nodes working in lower granularity.

Figure 4. High-level granularity negotiation process.

2.3. Real-Time Level Selection of Negotiation Granularity

The satellite needs to select an appropriate level of granularity. Selecting a higher level brings large data transmission and complex negotiation processes which may cause negotiation failure. Selecting a lower level can improve the success rate of negotiation but may cause bad negotiations.

Level selection is also difficult because the selections of satellites interact and unknown to each other. We collected lots of samples about selection and negotiation effectiveness. A neural network was trained to describe the relationship between the communication environment and success probabilities of negotiation in each level of granularity. Thus, the satellite can make decisions of level selection based on the probabilities in real-time onboard.

2.3.1. Neural Network Description

The negotiation information transmission is related to network status and size of information which is selected as input parameters of the neural network. In this paper, the network status is described by two parameters: remaining communication bandwidth and hop number to the *Leader*. The size of information is described by the number of targets n . The outputs of the neural network are the success probabilities of negotiation in each level of granularity recorded as P_L , P_{Mm} and P_H .

Suppose there are N_L layers. All layers are fully-connected. In the k th layer, there are N_n^k nodes. They are all connected to $(k - 1)$ th layer's nodes with parameters w_k and b_k . The activation function is optimized during training. The mean squared error MSE was chosen to evaluate network performance:

$$MSE = \frac{1}{M} \sum_{i=1}^M \|y_i - \hat{y}_i\|_2^2, \quad (17)$$

where y_i is the vector of real value form sample; $\hat{y}_i$ is the vector of predicted value output by the neural network.

2.3.2. Neural Network Training

The neural network training is divided into two steps: finding the best activation function and determining the neural network structure. In the first step, the two-layer model is set to be the initial neural network structure. The output layer's activation function is set to be the *purelin* model. There are 14 activation functions and different node numbers of hide-layers are traversed to find out the best three activation functions. In the second step, the three-layers structure and also the different node numbers in each layer are selected and traversed, including 4, 6, 8, 16, 32, 64, 128, and 256. By comparing the MSE, the best structure of the neural network is found.

3. Experiments and Analyses

3.1. Experimental Data

3.1.1. Targets and Mission Status

We randomly generated a set of targets and missions. The target's initial position was between 2° N and 2° S. The target's motion was also random, including course angle ψ_i between -180° and 180° and velocity v_i between 0 and 30 knots. The status of the mission defined as the numbers shown in Table 2 was random, between 0 and 5.

Table 2. Definition of the mission status.

Mission Status ID	Mission Status
0	not discovered
1	discovered
2	identified
3	confirmed
4	tracking
5	monitoring

Finally, we got the set of targets and mission information shown in Table 3; the rest of them are listed in Appendix A.

Table 3. Definition of the mission status.

Target ID	Position	Course Angle	Velocity	Mission Status
1	46.78° E, 0.16° N	13°	3.2	3
2	135.27° E, 1.05° S	78°	16.3	1
3	148.65° E, 1.62° N	−51°	8.6	0
...	...	...	...	...

3.1.2. Imaging Satellite

Satellite parameters are shown in Table 4. They are ten isomorphic satellites in the scene, with an interval phase of -5° .

Table 4. Parameters for satellite 1.

Category	Parameter	Value
Orbit Parameter	Semimajor Axis (m)	6,878,000
	Eccentricity	0
	Inclination (°)	0
	RAAN (°)	0
	Argument of Perigee (°)	0
	True Anomaly (°)	230
Attitude Parameter	Agile	Y
	Max. Maneuver Angle	45
	Angular Velocity	1
	Angular Acceleration	0.1
	Maneuver Mode	Trapezoid Method
Payload Parameter	Imaging Width (km)	4
	Image Size per Frame (Mb)	6.4
Communication Parameter	Frequency Band	S
	Antenna Number	4
	Bandwidth (M)	10
	Modulation Mode (M)	PSK
	Spread Spectrum Mode (M)	CDMA

At the beginning of simulation, mission 1 was assigned to satellite 4. Therefore, satellite 4 executed a tracking mission stage which affects the negotiation.

3.2. Neural Network Training

3.2.1. Sample Collection

We traversed 3 levels of granularity for satellite 6 to satellite 10; different bandwidth usage is shown in Table 5; and target numbers are shown in Table 6.

Table 5. Simulation value of bandwidth usage.

Group ID	1	2	3	4	5	6	7
bandwidth usage	60%	65%	70%	75%	80%	85%	90%

Table 6. Simulation value of target number.

Group ID	1	2	3	4	5	6	7	8
target number	30	40	50	60	70	80	90	100

We collected $3^5 \times 7 \times 8 = 13,608$ samples. Negotiation results can be obtained from the *Leader* node and converted into connectivity probabilities of each level of granularity as the output of the neural network. Remaining communication bandwidth Bw and the number of targets n are also the inputs of simulation. The hop number to the *Leader* node of each satellite H_n can be read from the simulation result.

3.2.2. Training Result

We chose 14 different activation functions (elliotstsig, hardlim, hardlims, logsig, netinv, poslin, purelin, radbas, radbasn, satlin, satlins, softmax, tansig, and tribas) using the first layer. Additionally, the activation function of the second layer was purelin. After training, the best ten neural networks' distributions of error for the first step in Section 2.3.2 were gathered to be shown in Figure 5, and their parameters are shown in Table 7.

The best three activation functions (radbas, softmax, satlin) were found out. The convergence process of the best neural network is shown in Figure 6.

Then we used these three activation functions and the three-layer structure to find out the best neural network. After training, the best ten neural networks' distributions of error for the second step were recorded for Figure 7, and their parameters are shown in Table 8.

The training convergence process of the best neural network is shown in Figure 8.

Figure 5. The error distribution of the best ten neural networks under the two-layer structure.

Table 7. Parameters of the best ten neural networks under the two-layer structure.

Network ID	First Layer		Second Layer	
	Activation Function	Nodes Number	Activation Function	Nodes Number
1	radbas	32	purelin	3
2	softmax	32	purelin	3
3	satlin	32	purelin	3
4	softmax	256	purelin	3
5	satlin	64	purelin	3
6	radbas	32	purelin	3
7	poslin	32	purelin	3
8	tansig	32	purelin	3
9	logsig	32	purelin	3
10	radbasn	32	purelin	3

Figure 6. Training convergence process of the best neural network under the two-layer structure.

Figure 7. The error distributions of the best ten neural networks under the two-layer structure.

Table 8. Parameters of the best ten neural networks under the two-layer structure.

Network ID	First Layer		Second Layer		Third Layer	
	Activation Function	Nodes Number	Activation Function	Nodes Number	Activation Function	Nodes Number
1	satlin	8	softmax	16	purelin	3
2	satlin	128	radbas	16	purelin	3
3	satlin	16	radbas	64	purelin	3
4	satlin	16	softmax	64	purelin	3
5	satlin	64	softmax	256	purelin	3
6	softmax	32	radbas	64	purelin	3
7	softmax	32	satlin	16	purelin	3
8	satlin	32	satlin	32	purelin	3
9	satlin	64	softmax	128	purelin	3
10	softmax	64	softmax	256	purelin	3

Figure 8. Training convergence process of the best neural network under the two-layer structure.

3.3. Experimental Results

All parameters in Tables 5 and 6 were chosen to verify the neural-network-based multi-granularity negotiation method. We collected and analyzed the profits and costs based on negotiation results. The profit shown in Figure 9 and the cost-effectiveness ratio shown in Figure 10 are most relevant.

Figure 9. Profit of each mode.

Figure 10. Cost-effectiveness ratio of each mode.

4. Discussion

Resources onboard are limited, which means the resources for satellite clusters, such as communication resources and computation resources, may have conflicting demands from the mission and negotiations. This causes failure of negotiations easily. Therefore we need a method to balance the gains from negotiation and negotiation delay. We proposed a method in which the negotiations are divided into three levels of granularity with different working modes. As shown the result, these three levels of granularity perform differently, which means this division is effective and can be used in different situations. Moreover, the satellite selects the best level onboard in real-time, and achieves better performance than each of the three levels. Additionally, the satellite can select the level onboard without many resources. The cost-effectiveness ratio of the satellite cluster is increased. It is worth noting that the three levels of granularity were designed by the author based on the author's task planning algorithm. That means the number and mode of each granularity may be different from those in other task planning algorithms.

We used machine learning to train a neural network. As shown in the training result, the MSE of the neural network reaches 0.002, which is not very good for the training itself, and this may be caused by a small sample set of 112 kinds of the sample. Those kinds of samples are the statistical result from 13,608 samples. However, this error is satisfactory in application, because possibility does not require high accuracy. Nonlinear fitting and small sample learning may obtain better results, but may not change the selection of granularity.

5. Conclusions

Limited communication and computational resources onboard are significant factors for negotiation, which is not getting enough attention. There usually exists cooperation in the micro-satellite cluster, which means the effect of time delay caused by communication and computational cannot be negligible.

In this paper, a neural-network-based multi-granularity negotiation method under decentralized architecture is proposed, which is proven to be effective. Advantages of the proposed method are summarized as follows:

1. The three levels of granularity we allocated work for different situations. We combine the advantages of the levels and use the best level of granularity according to the situation, which brings about better profits and a higher cost-effectiveness ratio.
2. Complex situation analysis for granularity selection also brings about time delay; therefore, a neural network is trained for granularity selection in real-time.
3. The framework of satellites is decentralization, which means it is suitable for a large satellite cluster containing failure nodes and malicious nodes.

Author Contributions: Conceptualization, Y.D. and X.D.; methodology, X.D. and Y.D.; software, X.D. and S.X.; validation, X.D., S.X. and Y.D.; formal analysis, X.D. and S.X.; investigation, X.D.; resources, Y.D.; data curation, X.D.; writing, X.D.; visualization, X.D.; supervision, Y.D.; project administration, X.D.; funding acquisition, Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (number 2016YFB0501102).

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

Appendix A

Table A1. Definition of the mission status.

Target ID	Position	Course Angle	Velocity	Mission Status
1	46.78° E, 0.16° N	13°	3.2	3
2	135.27° E, 1.05° S	78°	16.3	1
3	148.65° E, 1.62° N	−51°	8.6	0
4	105.82° E, 1.14° N	−133.97°	24.12	2
5	22.83° W, 1.58° S	−100.56°	29.78	0
6	170.52° W, 1.71° N	10.64°	2.11	1
7	117.21° W, 1.63° N	39.89°	11.31	3
8	155.38° W, 0.74° N	−71.28°	7.25	3
9	97.38° E, 0.82° S	−145.84°	6.68	0
10	126.06° E, 1.98° S	2.46°	9.55	0
11	105.34° W, 0.4° N	−25.63°	13.69	4
12	77.51° E, 0.86° S	168.43°	26.08	1
13	135.86° E, 0.35° S	106.17°	25.44	3
14	2.22° E, 1.64° N	−67.71°	22.44	4
15	70.67° W, 0.56° N	−16.66°	22.43	1
16	17.6° E, 0.68° N	−76.47°	27.61	3
17	126.7° E, 0.43° S	90.43°	11.53	2
18	112.02° E, 0.3° S	−23.47°	3.07	1
19	41.83° E, 0.19° N	−173.1°	14.92	3
20	95.34° E, 0.67° S	153.94°	12.87	2
21	79.91° E, 1.81° S	−33.18°	14.69	1
22	93.55° W, 0.25° S	151.94°	8.31	3
23	67.62° E, 1.14° N	−23.34°	0.76	3
24	137.7° W, 1.35° N	151.97°	6.76	4
25	2.81° E, 0.24° N	−32.75°	13.52	3
26	129° E, 0.07° N	116.76°	21.59	2
27	43.86° E, 0.48° S	−85.93°	21.44	4
28	36.48° W, 1.31° N	41.81°	6.89	4
29	97.03° E, 1.09° S	29.79°	8.49	2
30	113.56° W, 0.49° N	−169.45°	17.23	0

Table A1. Cont.

Target ID	Position	Course Angle	Velocity	Mission Status
31	133.4° E, 1.63° S	10.05°	22.34	3
32	19.59° W, 1.05° S	53.3°	20.34	3
33	8.16° W, 0.71° S	93.46°	1.85	4
34	1.83° W, 0.59° N	−162.03°	27.42	3
35	12.1° E, 1.21° S	−75.38°	6.97	5
36	38.86° W, 1.61° N	−144.91°	16.06	1
37	162.48° E, 0.09° S	−10.36°	1.36	3
38	19.94° W, 0.49° N	−66.05°	1.89	3
39	135.58° E, 0.59° S	5.57°	4.11	3
40	148.4° E, 0.92° S	−152.6°	3.02	0
41	66.06° W, 1.62° S	−0.55°	17.06	3
42	27.49° W, 1.47° N	178.75°	23.1	3
43	152.13° E, 1.88° S	82.93°	14.51	1
44	116.76° E, 0.43° N	−160.38°	29.13	4
45	35.72° E, 0.95° S	−47.87°	21.9	1
46	130.59° W, 0.77° N	−72.46°	3.6	4
47	153.51° E, 0.68° S	−179.4°	17.78	4
48	112.79° E, 1.31° N	−151.89°	14.86	5
49	163.11° E, 1.76° S	84.84°	12.45	4
50	126.49° E, 0.94° N	87.96°	18.17	3
51	60.67° W, 1.37° S	135.86°	18.85	2
52	22.05° E, 1.45° N	−88.89°	14.6	1
53	70.15° W, 1.28° N	10.91°	13.85	2
54	102.21° E, 0.24° N	62.43°	24.97	1
55	22.49° W, 0.06° S	92.65°	14.14	3
56	73.21° E, 1.43° N	112.86°	3.76	2
57	72.99° E, 1.56° S	−150.73°	7.81	0
58	97.73° E, 0.07° N	−160.3°	24.27	2
59	120.5° E, 1.01° N	−101.32°	18.42	3
60	83.49° E, 0.89° N	−143.76°	22.46	1
61	42.29° E, 0.54° S	−165.52°	22.5	4
62	119.69° E, 1.59° N	−49.59°	13.76	3
63	137.64° E, 0.16° N	109.7°	20.11	0
64	44.04° E, 0.9° N	145.48°	17.54	3
65	34.77° E, 1.94° N	93.66°	10.2	4
66	16.9° E, 1.71° S	−172.48°	9.42	1
67	166.71° E, 1.4° N	54.01°	26.03	1
68	8.43° W, 1.62° S	66.33°	16.09	2
69	9.69° E, 1.17° S	−136.29°	6.59	0
70	25.3° E, 1.55° N	133.45°	21.39	4
71	162.65° W, 0.72° S	45.93°	22.26	5
72	71.92° E, 0.75° S	−48.13°	18.14	2
73	126.34° W, 1.3° S	−59.5°	11.53	4
74	129.3° E, 1.21° S	96.87°	12.91	1
75	148.87° E, 0.6° N	−75.96°	28.33	4
76	175.76° E, 1.89° S	−7.35°	21.8	2
77	51.35° W, 1.51° N	−96.73°	5.56	0
78	82.54° W, 1.26° S	149.41°	19.72	2
79	88.86° W, 0.78° S	−19.87°	19.76	2
80	143.25° E, 1.43° S	−21.42°	15.94	1
81	34.11° E, 0.25° S	111.23°	22.74	4
82	66.3° E, 1.43° N	125.25°	14.13	1
83	88.45° W, 1.37° N	−42.87°	18.76	3
84	166.84° W, 0.1° S	17.65°	1.04	3
85	114.67° W, 1.96° N	29.31°	20.07	5
86	137.9° W, 1.33° S	−119.95°	24.72	2
87	135.84° W, 0.78° S	−91.32°	2.48	4

Table A1. Cont.

Target ID	Position	Course Angle	Velocity	Mission Status
88	130.55° E, 1.93° S	154.1°	2.58	5
89	65.21° W, 0.01° N	21.85°	13.07	1
90	155.26° W, 1.65° N	−112.8°	29.73	2
91	129.67° W, 1.78° N	−36.53°	7.34	4
92	99.27° W, 0.41° S	52.87°	15.02	3
93	33.13° E, 1.37° N	−139.99°	6.01	1
94	168.98° W, 0.99° N	170.18°	7.95	2
95	74.4° W, 1.4° N	19.14°	9.08	2
96	137.41° E, 0.36° N	165.85°	20.56	4
97	62.26° E, 1.99° S	83.51°	16.07	2
98	177.51° W, 1.49° N	−125.73°	9.44	2
99	34.16° W, 0.31° S	−102.79°	7.26	5
100	173.14° W, 1.55° N	167.35°	22.78	5

References

1. Van der Werf, G.R.; Randerson, J.T.; Giglio, L.; Collatz, G.J.; Mu, M.; Kasibhatla, P.S.; Morton, D.C.; DeFries, R.S.; Jin, Y.; van Leeuwen, T.T. Global fire emissions and the contribution of deforestation, savanna, forest, agricultural, and peat fires. *Atmos. Chem. Phys.* **2010**, *10*, 11707–11735. [\[CrossRef\]](#)
2. Opgenoorth, H.J.; Lockwood, M.; Alcayd, D.; Donovan, E.; Engebretson, M.J.; Van Eyken, A.P.; Kauristie, K.; Lester, M.; Moen, J.; Waterman, J.; et al. Coordinated ground-based, low altitude satellite and Cluster observations on global and local scales during a transient post-noon sector excursion of the magnetospheric cusp. *Ann. Geophys.* **2001**, *19*, 1367–1398. [\[CrossRef\]](#)
3. Lockwood, M.; Opgenoorth, H.; Van Eyken, A.P.; Fazakerley, A.; Bosqued, J.M.; Denig, W.; Wild, J.A.; Cully, C.; Greenwald, R.; Lu, G.; et al. Coordinated Cluster, ground-based instrumentation and low-altitude satellite observations of transient poleward-moving events in the ionosphere and in the tail lobe. *Ann. Geophys.* **2001**, *19*, 1589–1612. [\[CrossRef\]](#)
4. Rabideau, G.; Knight, R.; Chien, S.; Fukunaga, A.; Govindjee, A. Iterative Repair Planning for Spacecraft Operations in the ASPEN System. *ISAIRAS* **1999**, *440*, 99.
5. Rayman, M.D.; Varghese, P.; Lehman, D.H.; Livesay, L.L. Results from the Deep Space 1 Technology Validation Mission. *ACTA Astronaut.* **2000**, *47*, 475–487. [\[CrossRef\]](#)
6. Campbell, M.; Schetter, T. Comparison of multiple agent-based organizations for satellite constellations. *J. Spacecr. Rockets* **2002**, *39*, 274–283. [\[CrossRef\]](#)
7. Schetter, T.; Campbell, M.; Surka, D. Agent Systems, Mobile Agents, and Applications. In *Multiple Agent-Based Autonomy for Satellite Constellations*; Elsevier Science Publishers Ltd.: Cambridge, UK, 2003.
8. Schetter, T.; Campbell, M.; Surka, D. Multiple agent-based autonomy for satellite constellations. *Artif. Intell.* **2003**, *145*, 147–180. [\[CrossRef\]](#)
9. He, Y.; Chen, Y.; Lu, J.; Chen, C.; Wu, G. Scheduling multiple agile earth observation satellites with an edge computing framework and a constructive heuristic algorithm. *J. Syst. Archit.* **2019**, *95*, 205–208. [\[CrossRef\]](#)
10. Si-wei, C.; Jing, C.; Lin-Cheng, S. A MAS-based negotiation model and its application to multiple observation collaborative satellite planning. In Proceedings of the 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), Singapore, 26–28 February 2010; pp. 205–208.
11. Chong, W.; Ning, J.; Jun, L.; Jun, W.; Hao, C. Cooperative Co-evolutionary Algorithm in Satellite Imaging Scheduling of Cooperative Multiple Centers. In Proceedings of the IEEE Congress on Evolutionary Computation, Barcelona, Spain, 18–23 July 2010.
12. Iacopino, C.; Palmer, P.; Brewer, A.; Policella, N.; Donati, A. EO Constellation MPS based on ant colony optimization algorithms. In Proceedings of the International Conference on Recent Advances in Space Technologies, Istanbul, Turkey, 12–14 June 2013; pp. 159–164.
13. Zhao, Y.; Liu, Y.; Wen, G.; Ren, W.; Chen, G. Designing Distributed Specified-Time Consensus Protocols for Linear Multiagent Systems Over Directed Graphs. *IEEE Trans. Autom. Control* **2019**, *64*, 2945–2952. [\[CrossRef\]](#)

14. Buzzi, P.G.; Selva, D.; Hitomi, N.; Blackwell, W.J. Assessment of constellation designs for earth observation: Application to the TROPICS mission. *Acta Astronaut.* **2019**, *161*, 166–182. [[CrossRef](#)]
15. Globus, A.; Crawford, J.; Lohn, J.; Pryor, A. A comparison of techniques for scheduling earth observing satellites. In Proceedings of the Conference on Nineteenth National Conference on Artificial Intelligence, San Jose, CA, USA, 25–29 July 2004.
16. Araguz, C.; Closa, M.; Bou-Balust, E.; Alarcon, E. A Design-Oriented Characterization Framework for Decentralized, Distributed, Autonomous Systems: The Nano-Satellite Swarm Case. In Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 26–29 May 2019.
17. Gallud, X.; Selva, D. Agent-based simulation framework and consensus algorithm for observing systems with adaptive modularity. *Syst. Eng.* **2018**, *21*, 432–454. [[CrossRef](#)]
18. He, L.; Li, G.; Xin, L.; Chen, Y. An autonomous multi-sensor satellite system based on multi-agent blackboard model. *Eksplot. Niezawodn. Maint. Reliab.* **2017**, *19*, 447–458. [[CrossRef](#)]
19. Nag, S.; Li, A.S.; Merrick, J.H. Scheduling algorithms for rapid imaging using agile Cubesat constellations. *Adv. Space Res.* **2018**, *61*, 891–913. [[CrossRef](#)]
20. Wang, J.; Demeulemeester, E.; Qiu, D. A pure proactive scheduling algorithm for multiple earth observation satellites under uncertainties of clouds. *Comput. Oper. Res.* **2016**, *74*, 1–13. [[CrossRef](#)]
21. Wang, J.; Demeulemeester, E.; Hu, X.; Wu, G. Expectation and SAA Models and Algorithms for Scheduling of Multiple Earth Observation Satellites Under the Impact of Clouds. *IEEE Syst. J.* **2020**, 1–12. [[CrossRef](#)]
22. Ongaro, D.; Ousterhout, J. In search of an understandable consensus algorithm. *USENIX Annu. Tech. Conf.* **2014**, *14*, 305–319.
23. Castro, M.; Liskov, B. Practical Byzantine fault tolerance. *OSDI* **1999**, *99*, 173–186.
24. Dhillon, V.; Metcalf, D.; Hooper, M. Recent Developments in Blockchain. *Blockchain Enabled Appl.* **2017**, *11*, 151–181.
25. Jakobsson, M.; Juels, A. Proofs of Work and Bread Pudding Protocols. In *Secure Information Networks*; Springer: Boston, MA, USA, 1999; pp. 258–272.
26. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Environ. Sci. Technol.* **2003**, *37*, 1–9.

© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).