Journal of Systems Engineering and Electronics
Vol. 35, No. 4, August 2024, pp.842 – 854

Dynamic access task scheduling of LEO constellation based on space-based distributed computing

LIU Wei[1,2,*], JIN Yifeng[1,2], ZHANG Lei[1,2], GAO Zihe[1,2], and TAO Ying[1,2]

1. Institute of Telecommunication and Navigation Satellites, China Academy of Space Technology, Beijing 100094, China;
2. Innovation Center of Satellite Communication System, China National Space Administration, Beijing 100094, China

Abstract: A dynamic multi-beam resource allocation algorithm for large low Earth orbit (LEO) constellation based on on-board distributed computing is proposed in this paper. The allocation is a combinatorial optimization process under a series of complex constraints, which is important for enhancing the matching between resources and requirements. A complex algorithm is not available because that the LEO on-board resources is limited. The proposed genetic algorithm (GA) based on two-dimensional individual model and uncorrelated single paternal inheritance method is designed to support distributed computation to enhance the feasibility of on-board application. A distributed system composed of eight embedded devices is built to verify the algorithm. A typical scenario is built in the system to evaluate the resource allocation process, algorithm mathematical model, trigger strategy, and distributed computation architecture. According to the simulation and measurement results, the proposed algorithm can provide an allocation result for more than 1 500 tasks in 14 s and the success rate is more than 91% in a typical scene. The response time is decreased by 40% compared with the conditional GA.

Keywords: beam resource allocation, distributed computing, low Earth obbit (LEO) constellation, spacecraft access, task scheduling.

DOI: 10.23919/JSEE.2024.000071

1. Introduction

The spacecraft quantity grows rapidly with the fast-developing space applications such as communication, navigation, remote telemetry, remote sensing, manned spaceflight, and deep space exploration. Meanwhile, the pressure of real-time data relay is also increasing [1−3]. However, the traditional geosynchronous orbit (GEO) data relay satellite (DRS) system has some disadvantages such as less satellite amount, expensive device cost, high transfer delay, and limited beam source to provide access and data relay service for the large scale of spacecraft. The booming low Earth orbit (LEO) communication constellation [4−7] provides more chance for access and data relay of spacecraft based on the application of phased-array antenna and digital beam forming (DBF) technique [8−10]. The constellation consists of large amount LEO satellites and provides a low delay and high-capacity mesh network to connect spacecraft and ground stations. The data created by spacecraft can be transferred to an arbitrary destination after the spacecraft accessing in the network through a relay beam. However, the difficulty of beam source allocation is enhanced by the large size of LEO satellite constellation and large number of spacecrafts [11−13].

The autonomous scheduling system should provide a response as soon as possible when the spacecraft submit a beam request dynamically. The constrains factors such as visible windows between LEO satellites and spacecraft, beginning and ending time of tasks, and the relay link building time should be considered [14,15] in a beam resource allocation. In addition, the service experience is affected by the response time. It is a difficulty to provide a quasi-real-time response of beam resource task scheduling under such constrains. There are many researches proposed to improve the on-board scheduling performance of traditional GEO DRS [16−19], the research in LEO satellite field is scarce because that the LEO technique is at a rising stage.

The process of similar on-board task scheduling consists of three main parts: requirement submission and beam resource state collection, trigger condition checking, and tasks scheduling. The combinatorial optimization problem is more attractive. Therefore, the most research focus on the performance improvement of the third step. Many algorithms and strategies are proposed to maximize a multi-object function because that the satel-

Manuscript received April 06, 2023.
*Corresponding author.
This work was supported by the National Key Research and Development Program of China (2021YFB2900603) and the National Natural Science Foundation of China (61831008).

lite beam task scheduling is a similar optimization problem under source constraints [20–23]. The task scheduling problem of data relay satellite is a typical non-deterministic polynomical (NP)-hard problem [24]. The solving process has been more complex because of more LEO constellation satellite and more request spacecraft. In recent years, constrains satisfaction model is widely used in task scheduling of industrial production especially under dynamic conditions [25].

In addition, there are many methods mentioned in earlier literatures to solve the optimization problem [26–28]. However, the research objects are always single or multi-GEO relay satellite or LEO sensor satellite, and the constraints are always assumptive, static and ideal, the achievements may be not sufficient enough for an LEO constellation beam task scheduling. In [29,30], the ability of heuristic optimization algorithm to obtain an optimal solution was proposed. The widely used genetic algorithm (GA) and its innovation were used in DRS task scheduling. However, the algorithms were investigated in laboratory, and the practical constraints of spacecraft beam request were not fully considered.

In addition, the capacity of on-board computation in fore-mentioned research is not considered which affects the solution finding source. According to the provider's information, the performance of on-board computer (OBC) is not excellent to support complex computation. In [31], distributed on-board mission planning in multi-satellites system was proposed. The main idea is to enhance the robustness of system to decrease the risk of system faults. However, the cooperation among multi-satellites inspired us in the research.

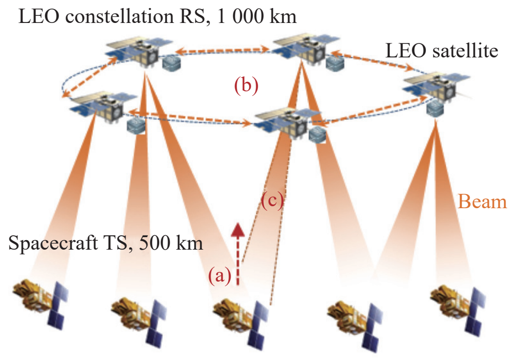
In this paper, we focus on all-process of dynamic task scheduling based on improved GA and on-board distributed computation. Both necessary and accessional constraints of the beam request and resource are proposed to support the direct application in an engineering case. The mixed trigger strategy can meet the dynamic requirement and reduce the total scheduling response frequency. The two-dimensional description of an individual can enhance the efficiency of GA. The scheduling task can be decomposed based on the uncorrelated single paternal inheritance method. Distributed computation architecture is provided to reduce response time and enhance the robustness of system. The proposed algorithm is verified by simulation and measurement. The results demonstrate the better performance in success rate of task and response time of scheduling compared with traditional GA.

This paper is organized as follows. The research object and task scheduling model are introduced in Section 2. The improved task scheduling algorithm is proposed in Section 3. The simulation and measurement results and compression and discussing are provided in Section 4 and conclusions are drawn in Section 5.

2. LEO constellation task scheduling model

2.1 Task scheduling scene and process

A multi-satellite and multi-spacecraft scene is given in Fig. 1. The LEO constellation is used to provide a quasi-real-time response for the various beam requests from spacecraft. The height of LEO constellation orbit is 1 000 km, and the number of satellites is RS. Each LEO satellite has 1−8 movable spot beams to meet the requirements from spacecraft. In addition, the beam can support only one spacecraft at one time. The orbit height and number of spacecrafts are 500 km and TS, respectively.



(a): Beam request; (b): Task scheduling; (c): Result response.

Fig. 1 Multi-satellite and multi-spacecraft model

All the spacecraft can submit their beam requests to the LEO constellation randomly and dynamically. The constellation should give a beam resource task scheduling response based on the considering of requirement and beam state as soon as possible. The large amount of LEO satellites and spacecraft increase the challenge in computation difficulty of task scheduling.

The detailed steps of the task scheduling process include three parts: (i) collection information of spacecraft requests and beam resource states; (ii) checking of trigger condition; (iii) task scheduling and obtaining the optimized result.

Step 1 Information collection

The requests and beam states should be collected to a satellite which plays a center role in the LEO constellation. It means that the center satellite knows all the necessary information for a task scheduling. And then, the beam source can be assigned to different spacecraft according to the beam requirements when the trigger condition is met.

The center satellite is denoted as core node and other satellites are named as normal node as shown in Fig. 2. All nodes can collect beam requests from spacecraft and transmit the requests information to the core node. The information collection is realized by a distributed resource monitor application.

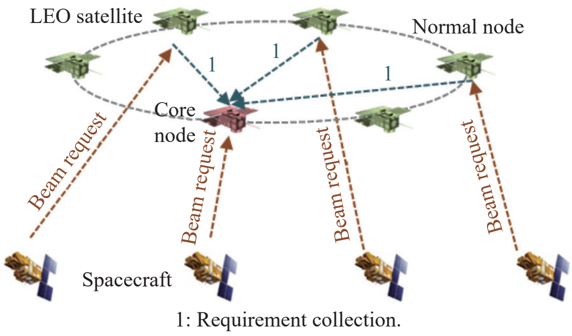


Fig. 2 Different notes in the LEO constellation

Step 2 Trigger condition checking

The trigger conditions are used to determine when to trig a new task scheduling of the observation window. The proposed trigger condition consists of two parts: cycle trigger condition and dynamic trigger condition.

The period of periodic triggering is set as 3 min to avoid not being triggered by tasks for a long time. And the observation window can keep moving forward under such a trigger. The task scheduling can run in a flow.

If a dynamic request from spacecraft is urgent, and the next cycle trigger is too far to meet the scheduling, the task would not be satisfied. A dynamic trigger is set to avoid such a case. If the beginning time of a task in task queue is less than ts from current time, the scheduling should be triggered. In this paper, ts is defined as 40 s.

The task scheduling trigger frequency cannot be too frequent because the scheduling process need time. A timer is used to record the time when to trig a new task scheduling, the timer setting is updated when a task scheduling is trigged by whether cycle trigger or dynamic trigger. The minimum interval between two closed trigger is set as 20 s in this work. The detailed sequence chart is presented in Fig. 3.

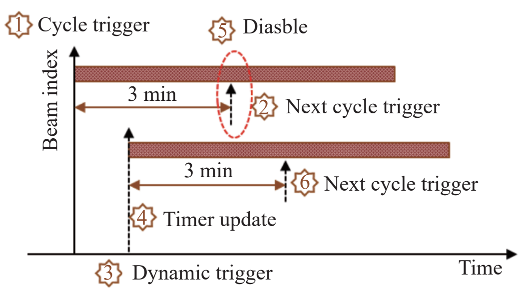


Fig. 3 Proposed trigger sequence chart

Step 3 Task scheduling based on distributed computing

As fore-mentioned in Section 1, the task scheduling of satellite beam source is a typical optimization problem with lots of complex constraints. In this paper, an improved GA is used to optimize the task scheduling result.

The computation cost created by the optimization iteration should be considered because the computation power of a LEO satellite is limited. Inspired by the distributed computing which is widely used in ground cloud computing and edge computing, The proposed GA is designed to support the multi-satellite distributed computing cluster. It means that the computation task of scheduling can be processed by multi-satellites at the same time to avoid the single power constraint and enhance the response speed. The distributed cluster is built based on resource virtualization. The computing on-board is provided by a form of resource pool to avoid the restricted power of a single LEO satellite. It is shown as Fig. 4, and the detailed technology introduction of distributed computing is given in Section 3 and Section 4.

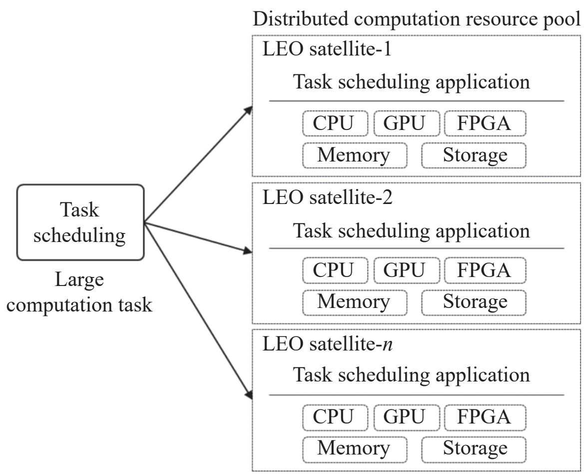


Fig. 4 Distributed computation resource pool

2.2 Task request and beam source model

According to the GA design, in the optimization process of the task scheduling, the constraints come from two main factors: task request and beam source.

The beam requests from user spacecraft are regarded as a task request model which can be presented by a set with 10 elements denoted as follows:

$$\mathrm{BR} = [\mathrm{TID}, \mathrm{RTS}, \mathrm{TSID}, \mathrm{TSN}, \mathrm{TP}, \mathrm{TF}, \mathrm{TS}, \mathrm{TE}, \mathrm{TD}, \mathrm{TED}]$$

where TID is the exclusive task number, RTS is the task state, TSID is the identity document (ID) number of spacecraft who submits this task, TSN is a supplementary parameter, TP indicates the priority of the task, TF indicates that whether the task can be decomposed into different parts or not, TS and TE are the beginning time and ending time of the task available window, TD is the length of the time window which can be calculated by TE-TS, TED is the actual task time like the time consumed by data transmission.

TID: The detailed information can be indexed in the system and database by this ID at any time, therefore, the TID number should be one and only.

RTS: Different value indicates different states of the task. The six states include to be submitted, submitted, scheduled, executing, executed, and failure.

TSID, TSN: The visible window between LEO constellation and request spacecraft is calculated according to the orbit and antenna information which can be indexed by TSID. And TSN can accelerate the indexing process.

TP: The priority affects the beam resource allocation result. The task will get beam resource earlier with the increase of TP.

TF: The available time window between LEO satellite and spacecraft may be not long enough because of the high-velocity motion. As for a long time task, the task may need to be decomposed into two parts to meet the available time window. It means that two beam support one spacecraft's request by their cooperation.

TS, TE: The beginning time and ending time of a time window in which the task can be executed.

TD: The length of the time window and TD can be calculated as TE-TS.

TED: The actual time of a task. If a sensing image's size is 5 GB and the link capacity of one beam is 1 Gbps, TED could be predicted as 40 s. For example, TS is 8:00 am and TE is 8:30 am, TED is 40 s, it means that the task scheduling system should provide a beam source which is longer than 40 s to meet this task during 8:00 and 8:30 am. The time window TD is 30 min, but the actual task time is only 40 s.

The necessary information of a task is concluded in this model, and it is enough to be scheduled. The spacecraft can submit such a beam request to the LEO constellation at anywhere and anytime, the constellation will schedule it and provide a response as soon as possible.

The information of beam resource is given as a mathematical model as following:

$$\mathrm{BRS} = [\mathrm{BID}, \mathrm{BT}, \mathrm{BD}]$$

where BID is the beam number which is one and only, BT is the type of the beam source like microwave and laser. BD is a supplementary parameter which is used to supplement BT like mechanical scanning and phase scanning.

In addition, the constraints are very important for the link between LEO satellite and spacecraft. The model is presented as follows:

$$\mathrm{CP} = [\mathrm{OED}, \mathrm{RED}, \mathrm{VED}, \mathrm{BRT}, \mathrm{LBT}]$$

where OED is the observation time window, it is time span in which the task can be scheduled at once a time. RED is the visible time window, VED is the available visible time window and BRT is the state of the beam resource.

RED: RED includes two elements RS and RE which are the beginning time and ending time of the window. The visible time window between LEO satellite and spacecraft can be calculated by the orbit and antenna information. It implies the fundamental and important constraints for the establishment of a communication link between the two spacecrafts. The space visible relationship is shown in Fig. 5.

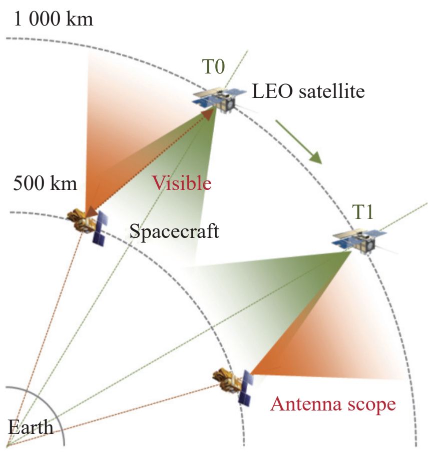


Fig. 5 Visible relationship between moving satellite and spacecraft

VED: VED includes two elements VS and VE which are the beginning time and ending time of the available visible window. The available visible time window is the intersection between RED and [TS,TE]. The inter-satellite link is effective when the two constraints are satisfied at the same time. The different time windows are given in Fig. 6.

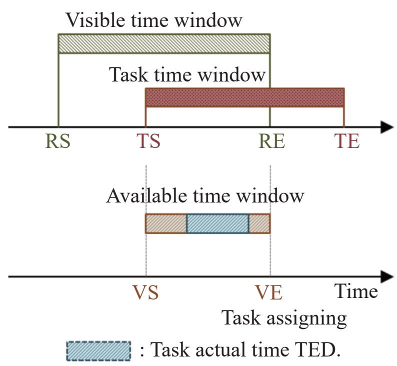


Fig. 6 Different time windows

BRT: Only the unoccupied beam resource can be assigned to the task. BRT is used to record the state of the beam source.

LBT: The time cost of a communication link building between LEO satellite and spacecraft. The typical value is 100 s.

All the parameters introduced above are used to describe the scene and the input of the proposed algorithm. Different scenes can be set with different parameter setting. The proposed algorithm has wider applicability.

3. Improved GA

In this section, the improved GA is described in detail. The main innovation points are two-dimensional individual model and uncorrelated single paternal inheritance method, respectively. The gene coding and decoding in conventional GA is unnecessary in the proposed algorithm based on the new individual description model. Therefore, the efficiency of the algorithm is enhanced. Meanwhile, the genetic process is only associated with a single paternal individual to support distributed computing. The benefit value of final optimized task scheduling result is enhanced and the response time of is reduced based on the two main points compared with the conventional centralized GA.

3.1 Flow of proprosed GA

There are three main steps of the proposed GA as shown in Fig. 7. Firstly, the initial population which has CN individuals is generated randomly. The individuals are described by a two-dimensional matrix and each individual is an available but not optimal beam resource task scheduling result. Secondly, a paternal individual can generate a son individual by gene operation. The new paternal population which has CN individuals is selected from the old paternal population and son population based on the objective function. Finally, the second step is repeated until the result converges or the end condition is met, and the optimized result is obtained. It is important that the gene coding and decoding processes are not operated in the flowchart to increase algorithm efficiency.

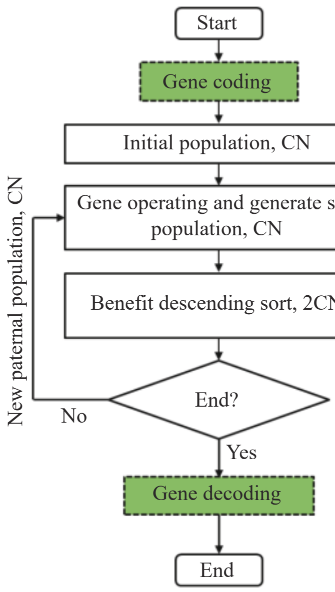


Fig. 7 Flowchart of the proposed GA

3.2 Two-dimensional individual model

The information of beam resource state, task time, spacecraft, and LEO satellite can be obtained directly from the Gantt chart. Therefore, the final optimized beam resource task result is usually described as the form of Gantt chart. Inspired by the Gantt chart, the individual method is designed as the same two-dimensional matrix form as shown in Fig. 8. In Fig. 8, the horizontal axis task scheduling window (30 min) is divided into 90 same length spans (20 s). The vertical axis is the beam resource number. The colored rectangle indicates the task scheduling result of a spacecraft request.

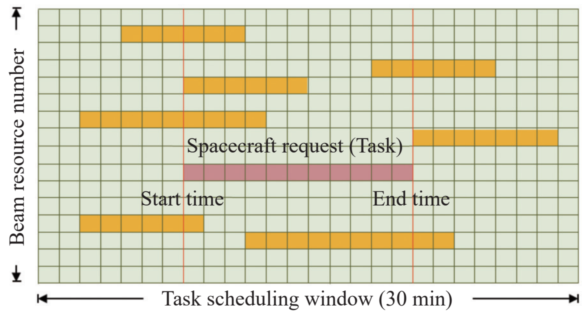


Fig. 8 Two-dimensional form of individual

The Gantt chart can be described by a matrix given integrally as

$$\mathbf{FMP} = \begin{bmatrix} \mathrm{fmp}_{11} & \mathrm{fmp}_{12} & \cdots & \mathrm{fmp}_{1j} \\ \mathrm{fmp}_{21} & \mathrm{fmp}_{22} & \cdots & \mathrm{fmp}_{2j} \\ \vdots & \vdots & & \vdots \\ \mathrm{fmp}_{i1} & \mathrm{fmp}_{i2} & \cdots & \mathrm{fmp}_{ij} \end{bmatrix} \tag{1}$$

where i is the index of beam resource and j is the index of time slice. The value of the matrix element can be 0 or the task ID. The beam resource is available when the element value is 0. The beam resource is occupied when the element value is a task ID. The total occupied time can be calculated by the continuous task ID which is regarded as a gene. And the start time and end time information can be obtained by the subscript j indirectly. The optimized **FMP** is the final task scheduling result. The individual model in the proposed algorithm uses such a matrix form to gene operating and avoid conventional gene coding and decoding. The efficiency is enhanced. This matrix **FMP** can fully represent the results and at it can be used as the individuals in optimization process. The elements in **FMP** are both spacecraft tasks and individual genes. The whole resource allocation result can be obtained by changing the spacecraft's task locations in the matrix which is like the Genetic operation in genetic algorithms. The detailed simulation, measurement, comparison, and analysis are given in Section 4.

3.3 Distributed computing

The proposed GA is designed to apply in LEO satellite. The limitation of on-board computing power should be considered. A large computation task should be broke into small sub-tasks based on the proposed algorithm to support distributed computing. The gene operating and inheritance method is improved in this paper.

The gene operating shown in Fig. 7 has two main steps.

Step 1 Some tasks scheduled, if any, are fetched out from the matrix **FMP** which means an old paternal individual to release the occupied beam resource.

Step 2 All the tasks unscheduled are put in the matrix with a strategy like random, priority and order to get a new matrix which means a son individual. In addition, under this case, the generation of son individual is only related to one paternal individual. The process is important for the distributed computing.

The flowchart of distributed computing is given as Fig. 9. In the initial step, the size of population is set as a large value CN×N, which is better for the optimized result. To avoid the intolerable computation cost, the population is broke into N sub-population whose size is CN. It means that the sub-task size is only 1/N of the initial task. The optimized process can be operated in different satellites.

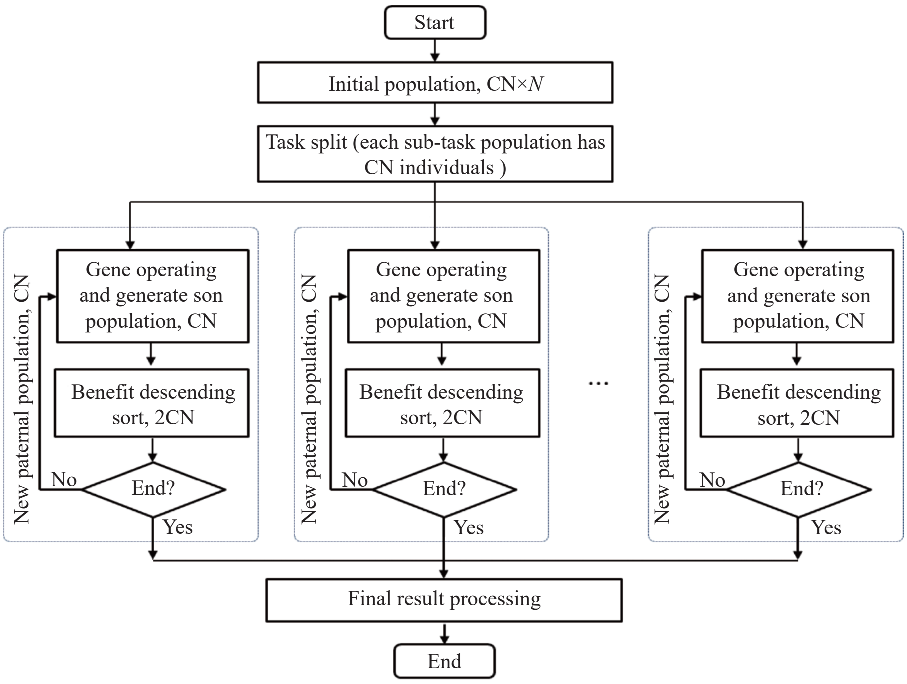


Fig. 9 Distributed computing of proposed GA

3.4 Objective function and ending condition

The objective function is designed to evaluate the quality of an individual. The input of the function is an individual status matrix obtained by **FMP** and the output is the benefit value of the individual. The individual is better when the benefit value is larger.

$$\mathrm{OF}=\alpha\cdot\sum_{i=1}^{\mathrm{TN}}\mathrm{PF}_i+\beta\cdot\sum_{j=1}^{\mathrm{TN}}I_j-\lambda\cdot\sum_{k=1}^{\mathrm{TN}}\mathrm{SW}_k \tag{2}$$

where TN is the amount of task planned. PF is the priority of task. SW is the beam switching number. The result is good for spacecraft when the first two terms are lager. It means more tasks are successfully planned. And the result is good for satellite when the third term is small. It means the system overhead is small.

The ending condition consists of convergence condition and maximum iteration condition. If the escalating rate of OF becomes small or the iteration reaches the preset value, the optimization would be stopped automatically to save computation time.

4. Simulation results and discussions

In this section, three kinds of numerical simulation results are presented to illustrate the availability and performance of the proposed algorithms. Firstly, the characteristic curve results of the improvement GA are given to indicate the validity. And then, the comparison results between different methods are provided. Finally, the proposed algorithm is implemented in Docker and Kubernetes (K8s) to illustrate the availability in distributed computing.

4.1 Validation of the proposed algorithm

To verify the proposed algorithm, a multi-satellite and multi-spacecraft model is created and simulated. The detailed parameters of simulation setup are given as in Table 1. Some parameters that are not listed are due to the use of typical values as mentioned above.

Table 1 Parameter of the example

Index	Parameter	Value
1	Number of LEO satellite (RS)	25
2	Beam number of per LEO satellite (BN)	4
3	Number of spacecraft (TS)	100
4	Observation time window (OED)/min	30
5	Time of link build (LBT)/s	100
6	Population size of i-GA (CN)	20
7	Iteration limitation of i-GA (LX)	20

The Gantt chart of spacecraft mission is given in Fig. 10, where the horizontal axis is time and the vertical axis is spacecraft, respectively. There are 675 missions during the OED are created, and the TS, TE information of missions can be indicated indirectly by the positions of the colored rectangles.

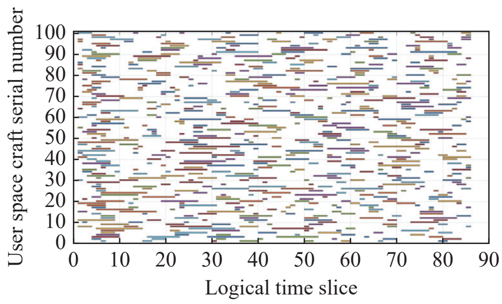


Fig. 10 Gantt chart result of the mission from spacecraft

This optimization is operated in a centralized computing model to verify the algorithm. The size of initial population is set as 20 and the maximum iteration is set as 20. The optimized result is given in Fig. 11. The time of communication link building between the LEO satellite and spacecraft is set as 100 s. It means that each task adds a header of 100 s. The task length looks like much longer in Fig. 11.

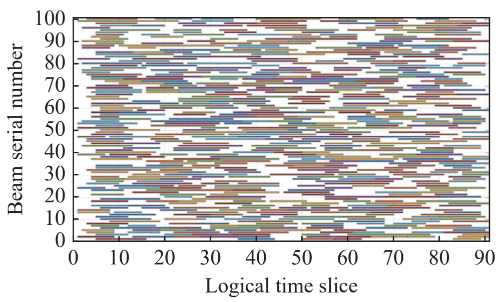
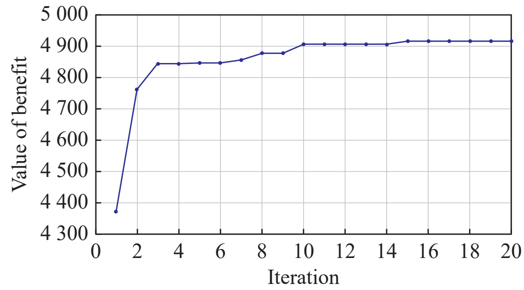
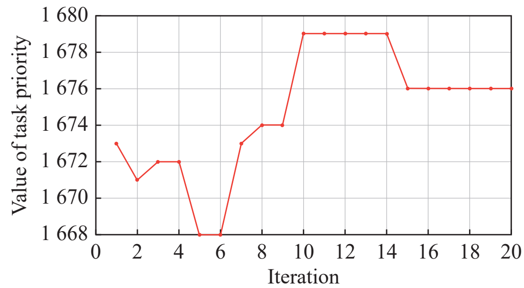


Fig. 11 Gantt chart result of task scheduling (time cost of link building is considered)

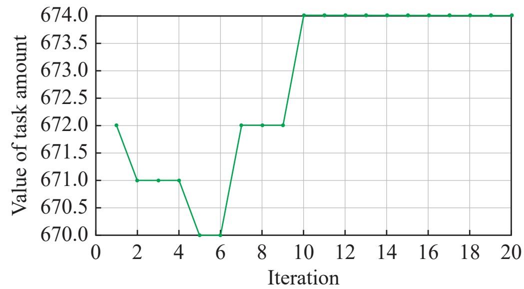
The simulation time is 9.23 s, and 674 tasks are successfully scheduled. The success rate is 99.85%. The characteristic curves are given in Fig. 12.



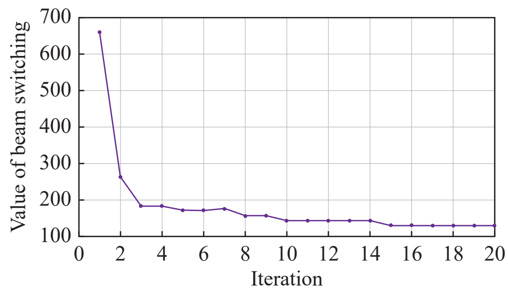
(a) Benefit value



(b) Task priority



(c) Task amount



(d) Beam switching

Fig. 12 Characteristic curves calculated by objective function

The proposed algorithm can get a convergence optimized result rapidly, and the result is good both for spacecraft and LEO satellite. The validity of the algorithm is verified.

4.2 Comparison analysis

There are three examples are given to illustrate the improvement of proposed algorithm compared with the conventional one.

Example 1 The performance analysis under the scarce beam source case

In this example, the value of TS is [50:4:86], the value of RS is [10:1:19], and BN is 4. The sample amount is 100. The characteristic curve, success rate and computation time results are given in Fig. 13–Fig. 15.

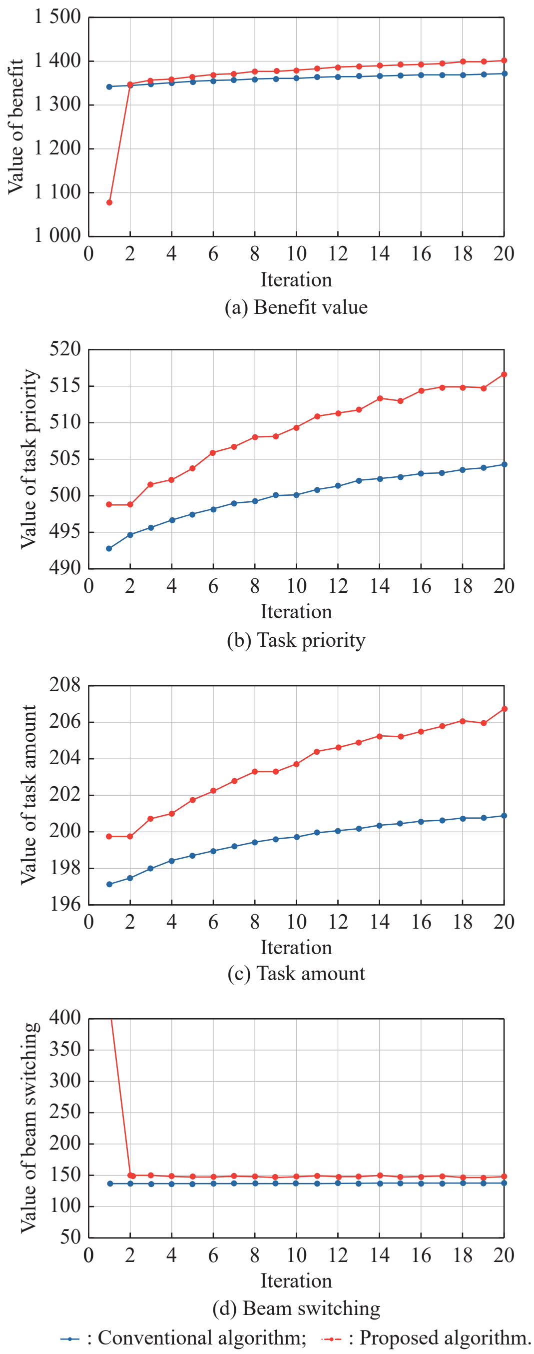


Fig. 13 Characteristic curve comparison results when the beam resource is insufficient

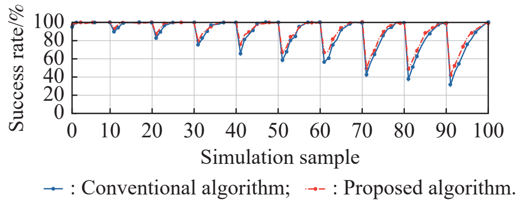


Fig. 14 Comparison results of success rate when the beam resource is insufficient

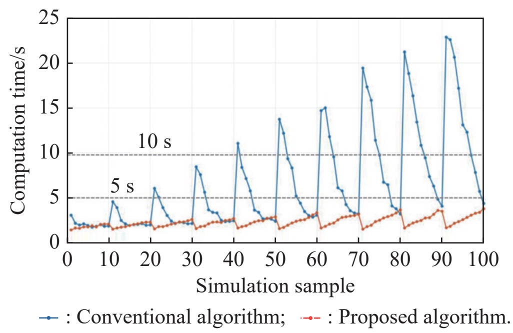


Fig. 15 Comparison results of computation time when the beam resource is insufficient

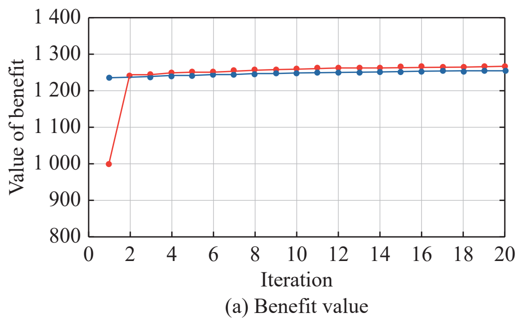
The results are calculated by the average value of 100 samples in 20 iterations. It is clear that the proposed algorithm (red) has a better performance than the conventional one (blue).

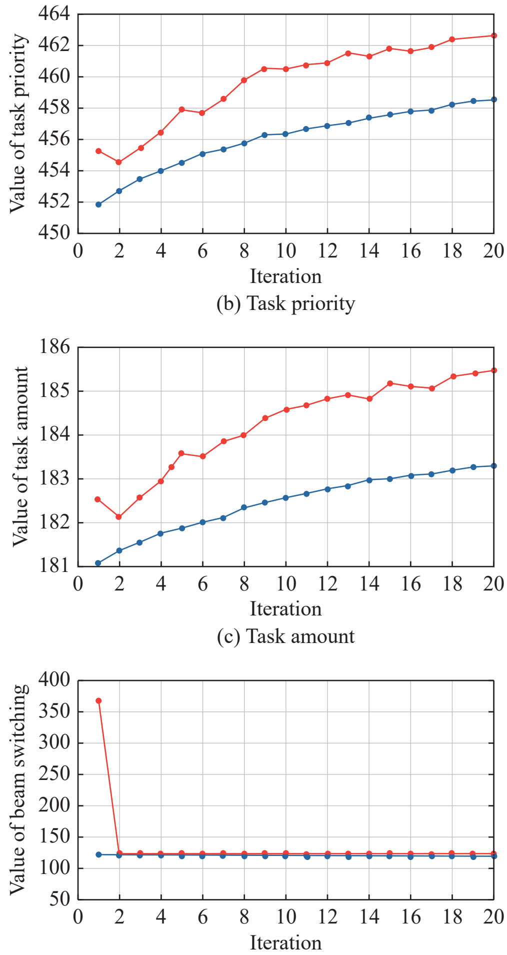
Due to the lack of beam resource, some tasks may not be scheduled successfully. However, the proposed algorithm (red) still has a higher success rate than conventional one (blue).

It is clear that the computation time of conventional algorithm (blue) is much longer than the proposed one (red), especially when the beam source is obvious fewer.

Example 2 The performance analysis under the enough beam source case

In this example, the value of TS is [40:4:76], the value of RS is [10:1:19], and BN is 4. The sample amount is 100. The characteristic curve, success rate and computation time results are given in Fig. 16–Fig. 18.





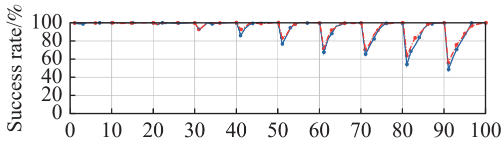
(b) Task priority

(c) Task amount

(d) Beam switching

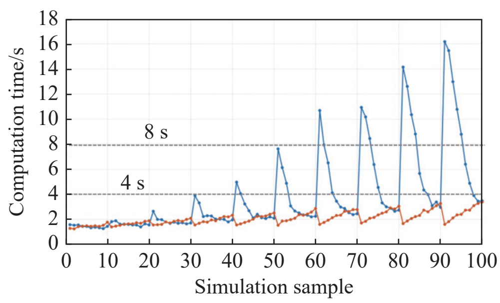
: Conventional algorithm; : Proposed algorithm.

Fig. 16 Characteristic curve comparison results when the beam resource is enough



: Conventional algorithm; : Proposed algorithm.

Fig. 17 Comparison results of success rate when the beam resource is enough



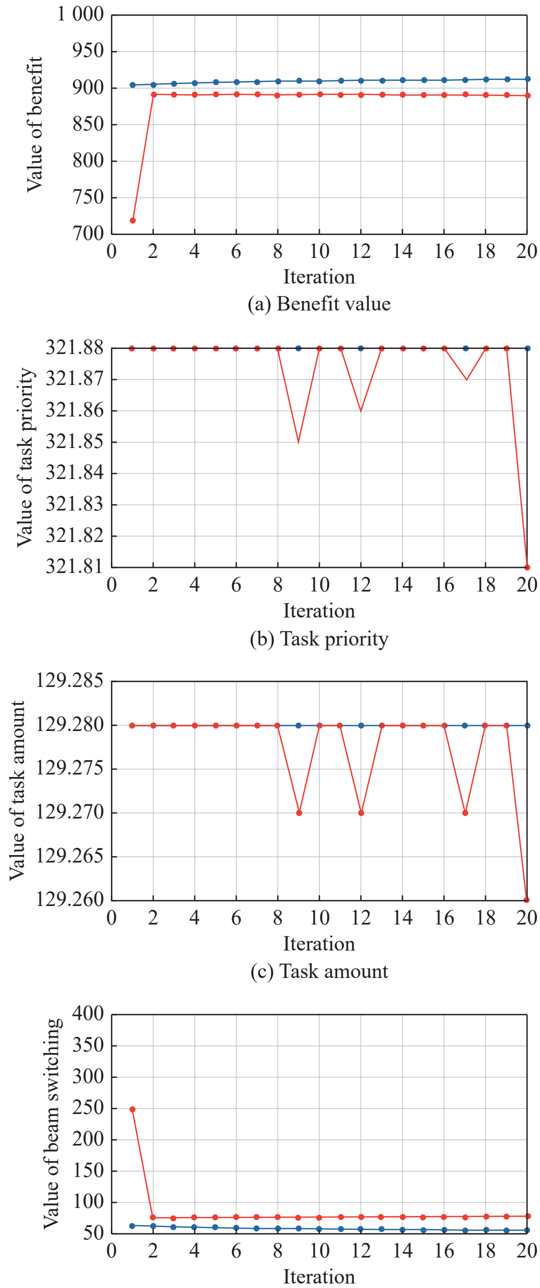
: Conventional algorithm; : Proposed algorithm.

Fig. 18 Comparison results of computation time when the beam resource is enough

The conclusions of these three results are similar to the Example 1. But the advantage is getting smaller.

Example 3 The performance analysis under the rich beam source case

In this example, the value of TS is [20:4:56], the value of RS is [10:1:19], BN is 4. The sample amount is 100. The characteristic curve, success rate and computation time results are given in Fig. 19–Fig. 21.



(a) Benefit value

(b) Task priority

(c) Task amount

(d) Beam switching

: Conventional algorithm; : Proposed algorithm.

Fig. 19 Characteristic curve comparison results when the beam resource is sufficient

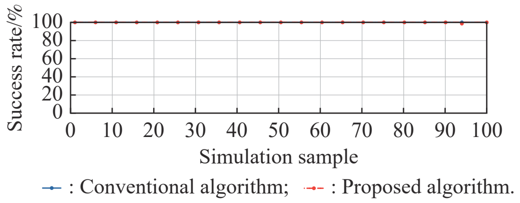


Fig. 20 Comparison results of success rate when the beam resource is sufficient

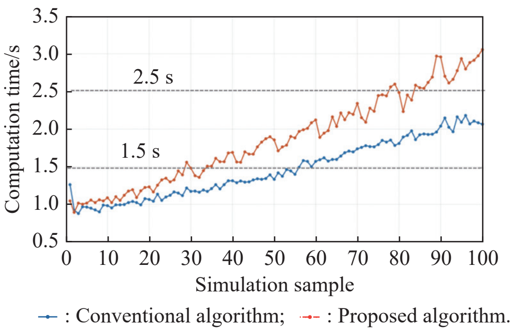


Fig. 21 Comparison results of computation time when the beam resource is sufficient

The results are calculated by the average value of 100 samples in 20 iterations. With the increase of beam source, the conventional algorithm (blue) has a narrow advantage compared with the proposed one (red).

Both the two algorithm has the same success rate when the beam resource is rich. All the tasks from spacecraft are scheduled successfully.

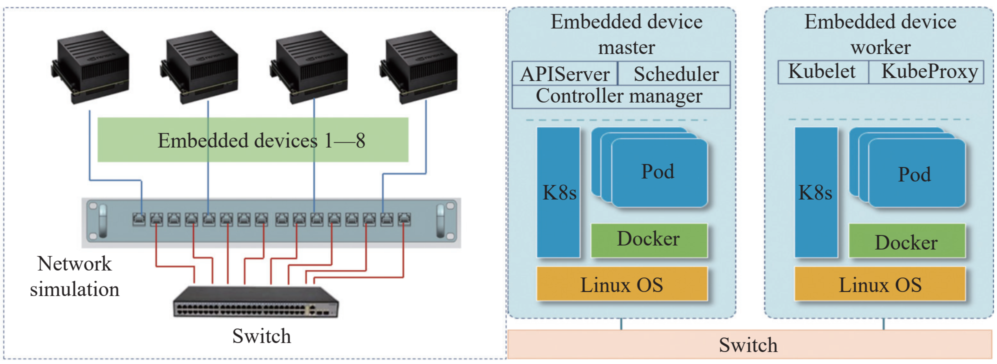
According to the results in Fig. 21, the proposed algorithm (red) has a bigger time cost because all the available beam resource in FMP is checked when a task is scheduled. The advantage created by the checking process decreases when the available beam resource is richer.

The three examples are used to illustrate the performance of proposed GA under different cases. The proposed algorithm has obvious advantage when the beam source is scarce which is acquainted with a real scene.

4.3 Analysis of distributed computing model

The computation of dynamic task scheduling became larger and larger with the increase of LEO satellite scale. Therefore, restricted by the on-board computing power, the large-scale computing task cannot be processed in a real-time by only one LEO satellite. Inspired by the cloud computing technology, the distributed computing model among multi-satellite can be used in a LEO constellation to solve the beam task scheduling by rich computation resource of the cloud cluster. In other word, a large computing task can be processed by multi-satellites at the same time to reduce processing time delay.

A principal proof distributed system as shown in Fig. 22 based on Docker and K8s is built in laboratory. There are eight embedded devices (ground commodities) used as simulator of satellite computing payload. The dynamic network is simulated based on the Linux tool TC and NETEM. The container engine Docker and K8S cluster components are installed on each device. The computing task can be scheduled into the device automatically by the distributed architecture to realize the multi-satellite cooperation.



(a) Device connection diagram (b) Distributed computing components

Fig. 22 Principal proof distributed system

The process of task scheduling of beam resource is given in Fig. 23 and described as follows:

Step 1 The core node of the beam resource task scheduling application creates a computation task according to the requests from spacecraft and the beam resource state of LEO satellites.

Step 2 And then, the computation task is divided into five subtasks with same scale and submitted separately to the distributed computing cluster.

Step 3 The virtual agent component Kube-Proxy puts the subtasks into different cluster node based on load balance strategy. The subtasks are processed independently and the task scheduling results are returned to the core node as mentioned in Step 1.

Step 4 The results of subtasks are compared and analyzed to obtain the optimized result. In other word, the final result is selected among the subtasks' results.

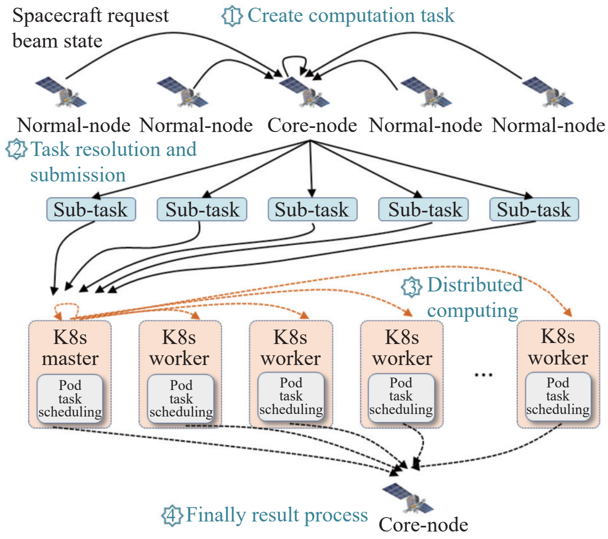


Fig. 23 Flowchart of distributed task scheduling

In Fig. 23, the large computation task is divided into many small parts as uncorrelated subtasks. As mentioned above, the distributed computing is supported by the improved GA based on the single paternal inheritance. Under this case, although the maximum benefit of the task scheduling result is reduced slightly. The computation time decreases significantly due to the reduction of computation task scale.

A numerical result of different simulation samples are given in Fig. 24 and Fig. 25 to illustrate the difference between traditional centralized computing and proposed distributed computing.

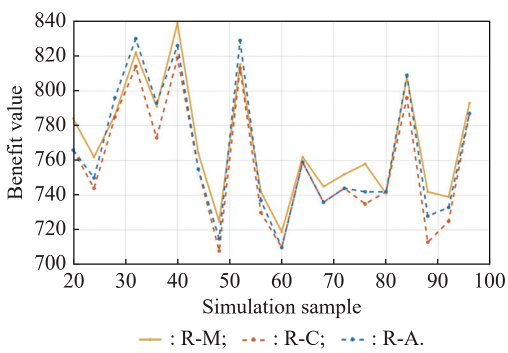


Fig. 24 Benefit value under three different cases

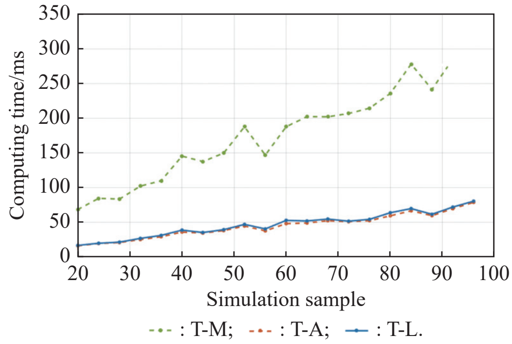


Fig. 25 Computing time under three different cases

In Fig. 24, R-M means the benefit of centralized computing result. R-A and R-C mean the average and worst benefit of distributed computing result. As shown in the results, the benefit value of centralized computing is better than the distributed computing result, but the difference is not obvious. In Fig. 25, T-M means the computing time of centralized computing result. T-A and T-L mean the average and longest computing time of distributed computing. As shown in the results. The distributed computing time is obviously less than the centralized one.

The conclusion can be drawn as that the distributed computing can save a lot of computing time at the expense of benefit reduction. If the typical time delay (17−30 ms) between different satellites is considered, the conclusion is still correct when the computation scale is large.

The conventional algorithm does not support the distributed computing model, only the proposed algorithm analysis is presented. In addition, the disadvantage like resource consumption due to additional software installations and network resource consumption due to data interaction within a distributed system should be alerted.

5. Conclusions

In this paper, an improvement GA which can be operated in on-board distributed computing system is proposed to solve the dynamic beam resource task scheduling in LEO constellation. Compared with the traditional algorithm, the benefit and computation time are both improved based on the two-dimensional individual model and uncorrelated single paternal inheritance method. The simulation results are given to illustrate the validation of the algorithm. In addition, a principal proof system of distribution computing based on Docker and K8s is built to present the advantage in computation time compared with the centralized one. It should be noted that the proposed algorithm consumes more time with increase of spacecraft. The boundary conditions that the algorithm applies to require continuous exploration.

Although, the distributed computing cluster can solve the limited computation power of single satellite, the influence on the on-board distributed cluster created by time delay and dynamic network should be considered in the further work.

References

[1] JING F, LI W, ZHAO W H. Research on access capability of laser link GEO data relay satellite system based on Erlang relay theory. Proc. of the International Conference on Robots & Intelligent System, 2020: 335−337.

[2] DENG C L, GUO W, HU W S, et al. Algorithm for the lightpath reservation provisioning of data relay services in a GEO network. Journal of Optical Communications and Networking, 2017, 9(8): 658–668.

[3] WU B B, FANG F, FU S. Improving the system performance in terrestrial-satellite relay networks by configuring aerial relay. IEEE Trans. on Vehicular Technology, 2021, 70(12): 13139–13148.

[4] TASNEEM D, GUNES K K, HALIM Y, et al. Location management in Internet protocol-based future LEO satellite networks: a review. IEEE Open Journal of the Communications Society, 2022, 3(1): 1035–1062.

[5] LI B, FEI Z S, ZHOU C Q, et al. Physical-layer security in space information networks: a survey. IEEE Internet of Things Journal, 2020, 7(1): 33–52.

[6] HAN W, WANG B S, FENG Z Q, et al. Grimm: a locator/identifier split-based mobility management architecture for LEO satellite network. Proc. of the 6th International Conference on Instrumentation & Measurement, Computer, Communication and Control, 2016: 605−608.

[7] CHEN Q, MA Z X, LAN B, et al. Multi-satellite tracking for the LEO satellite communication network. Proc. of the IEEE International Conference on Communications, 2022: 3082−3087.

[8] CHEN Q, XU Y Y, SONG C Y, et al. Adaptive tracking for beam alignment between ship-borne digital phased-array antenna and LEO satellite. Journal of Communications and Information Networks, 2019, 4(3): 60–70.

[9] SEONG-MO M, SOHYEUN Y, IN-BOK Y, et al. Phased array shaped-beam satellite antenna with boosted-beam control. IEEE Trans. on Antennas and Propagation, 2019, 67(12): 7633–7636.

[10] BAPTISTE P, NELSON J, MAXIME R, et al. Multi-beam antennas for very high throughput satellites in europe: technologies and trends. Proc. of the 11th European Conference on Antennas and Propagation, 2017: 2413−2417.

[11] LIU X J, XU K M, WU F Y, et al. A beam-dominating frequency resource allocation and scheduling scheme for multibeam satellite system. Proc. of the IEEE International Conference on Power Electronics, Computer Applications, 2021: 532−535.

[12] LIN Z Y, NI Z Y, KUANG L L, et al. Dynamic beam pattern and bandwidth allocation based on multi-agent deep reinforcement learning for beam hopping satellite systems. IEEE Trans. on Vehicular Technology, 2022, 71(4): 3917–3930.

[13] DENG B Y, JIANG C X, WANG J C, et al. Beam scheduling with various mission demands in data relay satellite systems. Journal of Communications and Information Networks, 2021, 6(4): 396–410.

[14] WANG L, JIANG C X, KUANG L L, et al. High-efficient resource allocation in data relay satellite systems with users behavior coordination. IEEE Trans. on Vehicular Technology, 2018, 67(12): 12072–12085.

[15] DENG B Y, JIANG C X, KUANG L L, et al. Two-phase task scheduling in data relay satellite systems. IEEE Trans. on Vehicular Technology, 2018, 67(2): 1782–1793.

[16] SHI D Y, LIU F, ZHANG T. Resource allocation in beam hopping communication satellite system. Proc. of the International Wireless Communications and Mobile Computing, 2020: 280−284.

[17] ZHANG T, ZHANG L X, SHI D Y. Resource allocation in beam hopping communication system. Proc. of the IEEE/AIAA 37th Digital Avionics Systems Conference, 2018. DOI: 10.1109/DASC.2018.8569327.

[18] LIU H Y, WANG Y, YU PENG, et al. Satellite relay task scheduling based on dynamic antenna setup time and splittable task. Proc. of the IEEE Global Communications Conference, 2022: 3917−3922.

[19] FANG Y S, CHEN Y W. Constraint programming model of TDRSS single access link scheduling problem. Proc. of the International Conference on Machine Learning and Cybernetics, 2006: 948−951.

[20] CHU X G, CHEN Y N, TAN Y J. An anytime branch and bound algorithm for agile earth observation satellite onboard scheduling. Advances in Space Research, 2017, 60(9): 2077–2090.

[21] TIZIANA S, GIOVANNI V, STEFANO S, et al. Planning and scheduling algorithms for the Cosmo-skymed constellation. Aerospace Science and Technology, 2008, 12(7): 535–544.

[22] ZHENG Z X, GUO J, EBERHARD G. Distributed onboard mission planning for multi-satellite systems. Aerospace Science and Technology, 2019, 89(1): 111–122.

[23] XU Y J, LIU X L, HE R J, et al. Multi-satellite scheduling framework and algorithm for very large area observation. Acta Astronautica, 2020, 167(1): 93–107.

[24] ZHUANG S F, YIN Z D, WU Z L, et al. The relay satellite scheduling based on artificial bee colony algorithm. Proc. of the International Symposium on Wireless Personal Multimedia Communications, 2014: 635−640.

[25] SAID N, MUHAMMAD I, JOSE M. DRALBA: dynamic and resource aware load balanced scheduling approach for cloud computing. IEEE Access, 2021, 9: 61283–61297.

[26] ZHIBO E, SHI R H, GAN L, et al. Multi-satellites imaging scheduling using individual reconfiguration based integer coding genetic algorithm. Acta Astronautica, 2021, 178: 645–657.

[27] LI P Y, LI J D, LI H Y, et al. Graph based task scheduling algorithm for earth observation satellites. Proc. of the IEEE Global Communications Conference, 2018. DOI:10.1109/GLOCOM.2018.8647444.

[28] SUNIL S, GOUTAM S, An optimal scheduling policy for satellite constellation deployment. Proc. of the IEEE International Conference on Industrial Engineering and Engineering Management, 2017: 100−104.

[29] WEI Z, LONG T, SHI R H, et al. Scheduling optimization of multiple hybrid-propulsive spacecraft for geostationary space debris removal missions. IEEE Trans. on Aerospace and Electronic Systems, 2022, 58(3): 2304–2326.

[30] CUI R W, HAN W, SU X C, et al. A multi-objective hyper heuristic framework for integrated optimization of carrier-based aircraft flight deck operations scheduling and resource configuration. Aerospace Science and Technology, 2020, 107(1): 106346.

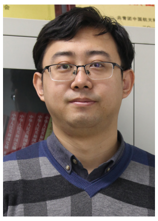
[31] WEN J, LIU X L, HE L. Real-time online rescheduling for multiple agile satellites with emergent tasks. Journal of Systems Engineering and Electronics, 2021, 32(6): 1407–1420.

Biographies

LIU Wei was born in 1993. He received his B.S. degree from Harbin Engineering University, Harbin, China, in 2015, and Ph.D. degree from Beihang University, Beijing, China, in 2020. He is currently an engineer of Institute of Telecommunication and Navigation Satellites with China Academy of Space Technology. His research focuses on the design of satellite communication system, distributed computing, cloud computing, edge computing, and artifical intelligence.
E-mail: emcf431@163.com

JIN Yifeng was born in 1995. He received his Ph.D. degree in communication and information systems from Peking University, Beijing, China in 2022, and now works in the Institute of Telecommunication and Navigation Satellite, China Academy of Space Technology as an engineer. His research interests are satellite communication, resource allocation, and space-terrestrial networks.
E-mail: jyfcast@outlook.com

ZHANG Lei was born in 1987. He received his B.E. degree in communication engineering and M.E. degree in electromagnetic theory and microwave engineering from Beijing Institute of Technology, Beijing, China, in 2008 and 2010. He was a visiting fellow with the French Institute of Aeronautics and Astronautics, Toulouse, France, in 2018, and now works in the Institute of Telecommunication and Navigation Satellite, China Academy of Space Technology as an engineer. His research interest includes design of satellite communication system.
E-mail: leizhang200@163.com



GAO Zihe was born in 1983. He received his B.S degree in electronics and information engineering, M.E. degree in signal and information processing, and Ph.D degree in information and communication engineering from Harbin Institute of Technology, Harbin, China, in 2005, 2007, and 2011, respectively. He has been engaged in the development planning, system demonstration, and key technology research of satellite communication
E-mail: biblejiayou@163.com

TAO Ying was born in 1974. She received her B.E. degree in wireless communication from Beijing Jiaotong University, Beijing, China, in 1997, and Ph.D. degree in communication and information system from Beijing Jiaotong University, Beijing, China, in 2004. Her research interest is design of satellite communication system.
E-mail: tao.ying@126.com