Swarm of Satellites: Implementation and Experimental Study of Multi-Agent Solution for Adaptive Scheduling of Large-Scale Space Observation Systems

Petr Skobelev[1(✉)], Elena Simonova[2], Vladimir Galuzin[3], Anastasiya Galitskaya[4], and Vitaly Travin[4]

[1] Samara Federal Research Scientific Center RAS, Institute for the Control of Complex Systems RAS, Sadovaya Street, 61, 443020 Samara, Russia
[2] Samara National Research University, Moskovskoye Shosse, 34, Samara 443086, Russia
[3] Samara State Technical University, Molodogvardeyskaya Street, 244, Samara 443100, Russia
[4] SEC «Smart Solutions», Moskovskoye Shosse, Office 1201, 17, Samara 443013, Russia
{galitskaya,travin}@smartsolutions-123.ru

Abstract. The paper dwells on solution to the problem of planning large-scale space observation systems, which can include from several dozens to hundreds of small spacecrafts. These systems are created in response to massive increase in the load on currently operating systems. New space systems, in comparison with traditional single spacecrafts, impose much more tough requirements on methods of planning, and only a few of the existing solutions can at least partially correspond to them. Thus, there is a need for new planning approaches that take into account domain semantics more deeply. The paper presents expanded application of multi-agent technology. Its essence lies in negotiations between agents of imaging through mutual compromises and concessions. The desired efficiency is achieved by searching for the near-to-global optimum for each application and using this information in a targeted search for a solution for the entire system. Experiments have demonstrated that approach helps promptly draw up a schedule for dozens of spacecrafts and thousands of observation objects.

Keywords: Small spacecraft · Space system · Ground stations · Observation objects · Multi-agent technologies · Adaptive planning

1 Introduction

Satellite systems based on small spacecrafts for Earth remote sensing (ERS) belong to a new generation of space observation systems (SpOS) designed to obtain images of the earth's surface in various spectral ranges. The obtained data is more and more in demand in various areas of human activity, such as military, forestry and agriculture, cartography, climate research, disaster recovery operations, etc. [1]. Thus, there is a need to build up the ERS orbital group and put into operation a large-scale constellation of small spacecrafts

F. Dignum et al. (Eds.): PAAMS 2021, LNAI 12946, pp. 267–278, 2021.
https://doi.org/10.1007/978-3-030-85739-4_22

along with traditional large-mass ones. Depending on the task of SpOS, they can vary from several dozens [2] to hundreds [3, 4] of satellites. Consequently, the number of ground stations for receiving and transmitting information (GS) also increases. They are part of the ground complex for servicing the orbital group.

To ensure targeted functioning of the space observation system, it is required to solve the problem of planning execution of incoming imaging applications, which is an NP-complete task [5]. At the same time, large-scale orbital constellations impose much more rigid requirements on methods and means of planning in comparison with traditional single ERS satellites. Among the main requirements are the following [6]:

- scalability: planning thousands of applications on a significant horizon without losses in processing speed with a growing number of applications and resources;
- adaptability: changing plans according to incoming events in a mode close to real time without stopping and completely recalculating the schedule;
- flexibility: taking into account individual characteristics of applications and resources to build the most optimal solution for multi-criteria optimization;
- efficiency: the time for placing new applications should be measured in minutes;
- reliability and survivability: in case of failure of some of the SpOS resources.

Most of existing developments such as SaVoir [7], STM [8], STK Scheduler [9], etc. are centralized, monolithic, hierarchical and sequential solutions. They only partially satisfy requirements, which makes them poorly applicable for large-scale SpOS. Thus, there is a need either for serious revision of existing software and algorithmic solutions, taking into account the emerging requirements, or for development of new approaches to planning orbital groups, taking into account domain semantics more deeply.

One of such approaches to resource management in complex systems is the use of virtual market methodology based on multi-agent technology (MAT) [10, 11], allowing flexible adaptation of schedules by events in real time. MAT also takes into account individual characteristics of orders and resources. The model of demand-resource network (DR-network) helps create high-performance, distributed, fault-tolerant solutions for resource management of SpOS in comparison with traditional methods.

The purpose of this work is to present the implementation and experimental study of a two-stage hybrid method for planning large-scale SpOS: 1st stage - building an initial plan using a greedy optimization algorithm (conflict-free planning); 2nd stage - multi-agent adaptation and optimization (proactive planning).

This approach develops the initial solution [12, 13] by improving architecture of multi-agent system (MAS) and introducing additional heuristics, significantly reducing complexity of combinatorial search for a solution for virtual market and DR-networks.

The second section of this paper proposes a SpOS model and formulation of the planning problem. Section 3 provides an overview of the current state of solutions. The fourth section describes the developed adaptive scheduling method. Section 5 examines a prototype SpOS planning system and presents experiment results. Section 6 provides a conclusion on development and application of the described solution.

2 The Problem of Planning Space Observation Systems

2.1 Model

The SpOS model consists of a set of small spacecrafts $Sat = \{sat_i\}$, $i = \overline{1, L}$ and a set of ground stations $GS = \{gs_j\}$, $r = \overline{1, G}$. Each sat_i spacecraft has its own orbit O_i (the orbits of satellites can be located both in one plane, and in different planes; in the first case they have a similar trajectory), limiting roll angle $maxRollAngle_i$ and pitch angle $maxPitchAngle_i$, as well as parameters of imaging equipment (f - focal length, $matx$ - matrix dimensions, minimum angle of sun elevation $minSunAngle_i$, $memVol_i$ memory capacity). And each gs_j is characterized by geographic location $coord_j$ and parameters of antenna (opening angle and data reception rate). The composition of spacecrafts and GS may change over time. Each satellite may have restrictions for data transfer to a certain GS. Besides, time intervals of inaccessibility can be indicated.

The targeted functioning of SpOS consists in execution of a set of applications $R = \{r_p\}$, $p = \overline{1, P}$. The r_p application can have its priority pr_p and restrictions (execution period $t_p = [t_p^{start}; t_p^{end}]$, admissible image linear resolution $minR_p$ and $maxR_p$ and admissible sun angle $minSunAngle_p$ and $maxSunAngle_p$). Besides, R can also change.

In the described model, two operations are performed:

- imaging of the observation object (OO), characterized by execution interval $t_p^{imag} = [t_p^{imagStart}; t_p^{imagEnd}]$, roll and pitch angles $rollAngle_p$ and $pitchAngle_p$.
- transfer of the images $drop_p$, characterized by execution interval $t_p^{drop} = [t_p^{dropStart}; t_p^{dropEnd}]$ and data transmission speed $baudRate_p$.

2.2 Problem Statement

It is necessary to provide adaptive scheduling of incoming applications, redistributing them between spacecrafts in order to increase the SpOS productivity, obtain images of the highest quality, minimize the lead time for individual orders and ensure fulfillment of other criteria. The system's objective function (OF) has the following form:

$$OF = \frac{1}{S} \sum_{k=1}^{N} OF_k \rightarrow max, \tag{1}$$

$$OF_k = \sum_{m=1}^{M} c_m F_m^k \rightarrow max, \tag{2}$$

where

OF is the system's objective function,
OF_k – is the OF of the k-th application,
S is the total number of applications,
N is he number of placed applications,
M is the number of optimization criteria,

c_m is the weighting factor of the m-th optimization criterion, such that $0 \leq c_m \leq 1$, $\sum_{m=1}^{M} c_m = 1$,
F_m^k is evaluation of the m-th optimization criterion for the k-th application.

Minimization of the imaging time F_1^k (3) and maximization of image quality F_2^k (4) are chosen as optimization criteria.

$$F_1^k = \frac{t_k^{end} - t_k^{imagEnd}}{t_k^{end} - t_k^{start}}, \tag{3}$$

$$F_2^k = \frac{minR_k - r\left(f, \text{matx}, \text{rollAngle}_k, \text{pitchAngle}_\text{p}\right)}{minR_k - maxR_k}, \tag{4}$$

where r is the function for linear resolution of the image for the k-th application [14].

3 State of the Art Review

Various classical and metaheuristic optimization algorithms are proposed for solving the problem of planning space imagery. Application of machine learning (ML) methods is also studied. One of the most famous metaheuristic algorithms is the ant colony one [15, 16]. Other equally popular algorithms are the local search method [17, 18] and the genetic algorithm [19, 20]. Heuristic and metaheuristic algorithms show higher performance in comparison with traditional optimization methods, however, heuristics requires strict specifications for problem conditions. Meanwhile, operation time and quality of obtained solutions can strongly depend on the initial data. Attempts of using ML methods are described in [21–23]. ML has great potential because it allows for training on data but does not require users to hard-code parts of the algorithm. However, ML algorithms currently have limited interpretability (e.g., there is no way to explicitly specify constraints) and require quite a large amount of data for training.

Recently, approaches to planning ERS data using agents have begun to develop [24–26]. The planning process proposed in [24] consists in interaction between agents of the survey strip and agents of the spacecraft. It is based on heuristics of programming in constraints together with virtual market approach. Results of its comparison with the currently used greedy algorithm show advantages of the proposed approach. However, performance of this solution is still insufficient for the task proposed in this paper. [25] describes the mechanisms of market auctions for distribution of orders for OO imaging between spacecrafts. They are operated by their own mission centers, coordinating their schedules using auction protocols, bidding on vacant orders based on the influence on the onboard plan and forecasted profits. [26] discusses the idea of fully autonomous planning on board a spacecraft. Its main advantages lie in using the current actual data on the state of the spacecraft and its resources to respond to emerging events in real time. However, in order to create a full-fledged MAS using a spacecraft in orbit, it is necessary to overcome limitations of the computing capabilities of onboard equipment. It is also important to build a stable communication system with several spacecrafts.

Complexity and dynamics of the market of ERS services leads to the fact that traditional, centralized, hierarchical and sequential methods based on heuristic algorithms do not effectively solve the problem of adaptive resource management for large-scale SpOS with acceptable quality and within the required time. A promising area is the use of methods and algorithms based on artificial intelligence and an agent-based virtual market, taking into account the domain semantics, conflict analysis, non-deterministic behavior, self-organization and adaptation in real time. However, currently these methods are at the stage of initial development, therefore, integral solutions, suitable for practical digital implementation, have not yet been designed and implemented [26, 27].

4 Adaptive Planning Method

Figure 1 shows a diagram of the adaptive planning method, which consists of preparing initial data (visibility intervals and placement options) and hybrid scheduling, combining a greedy optimization algorithm (conflict-free scheduling) with multi-agent adaptation and optimization (proactive scheduling).

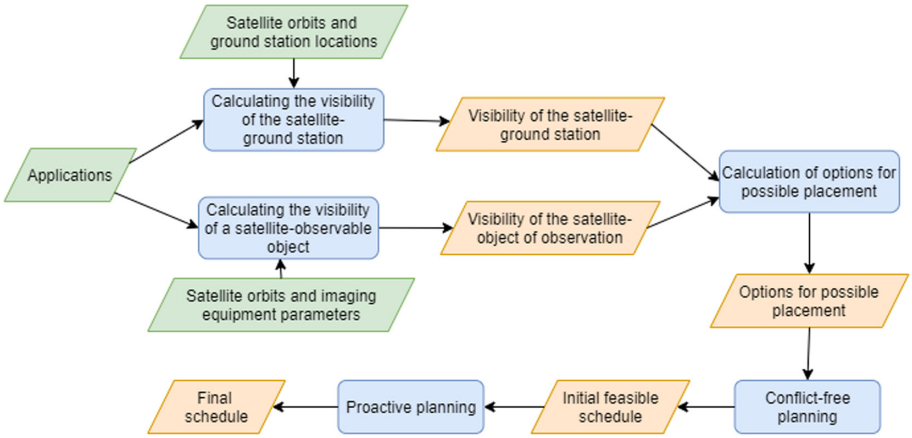

Fig. 1. Diagram of the adaptive planning method

Before planning is initiated, calculation of the satellite-GS and satellite-OO visibility intervals is performed on the specified planning horizon $[t_{min}^{start}, t_{max}^{end}]$. Next, possible placement options po_k for each application are calculated. The possible placement option is a combination of both visibility intervals within which imaging and dropping operations can be performed. The exact operation time is determined during planning.

Calculation of possible placement options is implemented based on the method of successive concessions between optimization criteria. A sequence of possible placement options is formed, sorted in descending order of objective functions of applications (2). The first place is for the placement option at the global optimum point.

4.1 Conflict-Free Planning

At this stage, an initial feasible schedule is constructed using a greedy optimization algorithm. The quality of the schedule does not really matter. The purpose is to form an initial state for the next stage of proactive planning. Applications occupy the first vacant placement option, without trying to displace those that are already allocated.

Algorithm 1 presents the pseudocode for conflict-free planning method. The list of applications is organized and grouped according to the value of the pr_p priority (lines 1–2). Then, for each group of applications, an attempt of planning is made (line 4–11). Options for placing each application are sequentially sorted out (line 7–11). For the next option, a search is performed for specific intervals of imaging and transmitting operations within the specified visibility intervals (line 8), for which there are no conflicts with other previously placed applications. If such intervals are found, the imaging job is formed (line 10–11). Otherwise, the algorithm proceeds to the next placement option.

Algorithm 1: Conflict-free planning algorithm

```
Input: applications, ImgJobOpts is set of possible placement
options for the problem
Output: ImgJobs is set of planned jobs
1:  groupedApplications = group(applications, 'priority')
2:  sort(groupedApplications, 'priority', 'desc')
3:  ImgJobs = []
4:  for applicationGroup in groupedApplications
5:    parallel for application_j in applicationGroup
6:       ImgJobOpts_j = ImgJobOpts[application_j]
7:       for imgJobOpt_k in ImgJobOpts_j
8:         imgAndDropInters = findImgAndDropInters(imgJobOpt_k)
9:         if imgAndDropInters not empty
10:           imgJob= createImgJob(ijo_k, task_j, imgAndDropInters)
11:           ImgJobs.push(imgJob)
12: return ImgJobs
```

4.2 Proactive Planning

At this stage, the resulting schedule is optimized using a multi-agent approach, which consists in competition and cooperation of agents with certain resources or demands [12]. Agents interact via negotiations on the virtual market through mutual compromises and concessions and arrive at a locally optimal solution.

Two types of agents are introduced: an application agent with the goal of occupying the most advantageous placement, and a scene agent, designed to control the activity of application agents and interact with external systems. The application agent is responsible for making changes to the schedule: it can move another agent from a more advantageous position or change its own position upon the request of another agent. To assess the current position of an agent, its satisfaction function $SF_k(po_k)$ (5) is used, which is the difference between the value of the task's OF (2) at the global optimum point $OF_k(po_k)$ and the OF value for the current accommodation option $OF_k(po_k)$:

$$SF_k(po_k) = 1 - (OF_k(p\dot{o}_k) - OF_k(po_k)). \tag{5}$$

During proactive planning, the scene agent grants the proactive right to application agents, starting with those with the least advantageous position ($SF_k(po_k) = min$).

Figure 2 shows the negotiation protocol of agents during proactive planning.

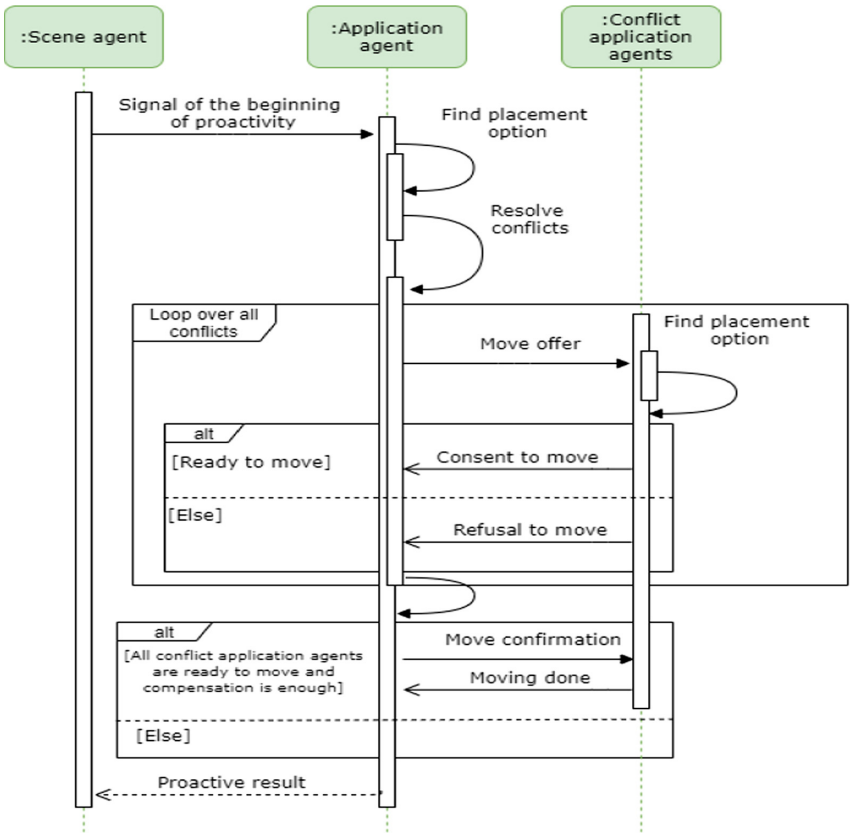

Fig. 2. Negotiation protocol at the stage of proactive planning

An agent that is launched for proactivity acts according to the following algorithm:

1. Sequentially search through placement options that are better than the current one.
2. For the next placement option, determine the list of conflicting applications and send a message to their agents with a proposal to find another placement, using compensation equal to the increment of the agent's OF $\Delta F = F_j\left(\widetilde{po}_j\right) - F_j(po_j)$. Upon receipt of this message, the agent of the conflicting application R_c recursively searches for another placement option using a similar algorithm (embedded proactivity) and sends its solution in a response message. It indicates the agent's willingness to change its position in the schedule and the ΔF_c losses in case of agreement.

3. If all agents of conflicting applications agree to move, the total losses $\sum \Delta F_c$ are evaluated, and if the agent of a proactive application can compensate for them due to the increase of its OF, i.e. $\Delta F > \sum \Delta F_c$, the resulting permutation is applied.
4. Otherwise, the application agent proceeds to the next placement option (point 1).

The schedule is synthesized until, during the next planning iteration, none of the application agents can occupy a more advantageous position. This means achievement of the local optimum of the system's OF. After that, the constructed schedule is saved.

It is important to note that the state of the system is not static. The data may be subject to changes due to arrival of new events. In this case, part of the constructed schedule may become irrelevant and adaptively adjusted by conducting new negotiations between the application agents without stopping and restarting the system.

5 Software Implementation and Experimental Studies

To test the applicability of the proposed approach to solving the problem of planning large-scale SpOS in real time, its software implementation has been created in the form of a prototype of a multi-agent system for planning the targeted use of SpOS (Fig. 3). This prototype has been then used for a number of experimental studies.

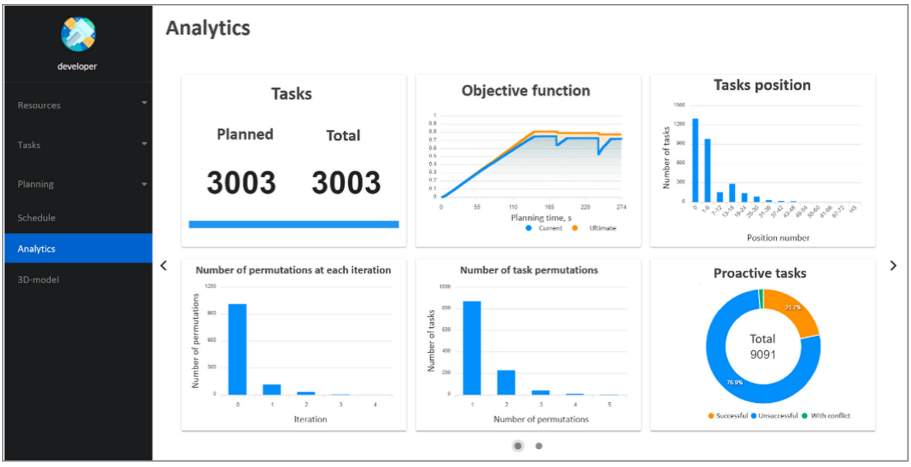

Fig. 3. System user interface

The prototype has a client–server network architecture. The server side of the system is written in Java using the Spring framework. The user interface is a one-page web application through which the initial data is loaded and modified (spacecrafts, GS, applications, calendars, resource availability restrictions, etc.). The interface also helps manage the planning progress, monitor resources, view reports and planning results.

5.1 Investigation of the Method's Performance and Ability to Adapt the Schedule

This study evaluated performance of the presented method and its ability to adapt the schedule damaged by failure of one of the spacecrafts. During the experiments, the scheduling of applications for OO imaging has been carried out first, the number of applications varied from 100 to 20,000 for a different number of small spacecrafts (15, 25, or 35). The system's OF value (1) (Fig. 4a) and the planning time (Fig. 4b) were measured. Then, one of the spacecrafts was excluded from the system and the time spent on restoring the solution was also measured (Fig. 5). The experiments were carried out on a PC with a 4-core CPU Intel Core i7-3770 and 8 GB RAM. The number of GS in all experiments was the same - 20. The planning horizon was 21 days.

Experimental results have shown that the developed method meets performance requirements when working with large volumes of applications. In this case, the quality of the obtained solution weakly depends on the number of small spacecrafts and applications in the system. The time spent on restoring a schedule damaged by failure of one of the spacecrafts is a much smaller fraction of the total planning time and increases proportionally with the growing number of applications and spacecrafts. Comparison of the obtained results with those presented in [12, 17, 24] shows that the developed algorithm demonstrates higher performance and scalability, allowing for a similar time interval to process a much larger number of incoming applications (by 5–10 times).

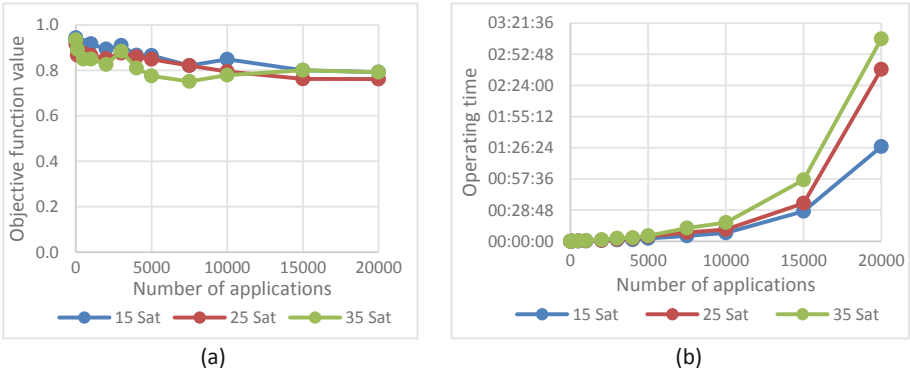

Fig. 4. Graphs of dependence of the OF value (a) and the planning time (b) on the number of applications and spacecrafts in the system

5.2 Comparison with Centralized Scheduling Algorithms

In this study, the effectiveness of the developed method has been analyzed in comparison with centralized scheduling algorithms based on traditional optimization methods, such as the simulated annealing algorithm and the Tabu Search algorithm, implemented in the Optaplaner open-source Java scheduling framework. They have been compared based on the quality of the resulting schedule and the time required for its compilation.

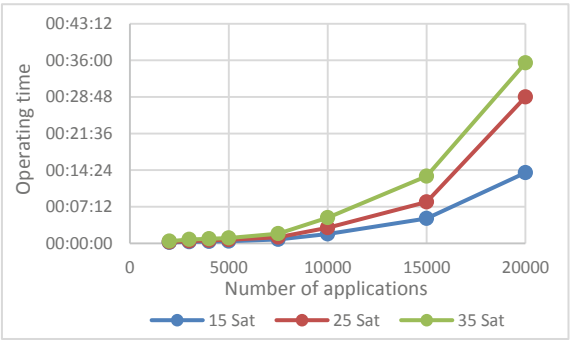

Fig. 5. Graphs of dependence of the schedule recovery time on the number of applications and spacecrafts in the system

In the course of the study, a series of experiments have been carried out, in which the number of applications for OO imaging varied from 100 to 5000. At the same time, the time spent on drawing up the schedule and the system OF value have been measured (1). The PC configuration and the number of ground stations are similar to the previous experiment, the number of spacecrafts is 25. Based on the results of the experiments, graphs of dependence of the system OF value (Fig. 6a) and the planning time (Fig. 6b) on the number of applications for various planning algorithms have been built.

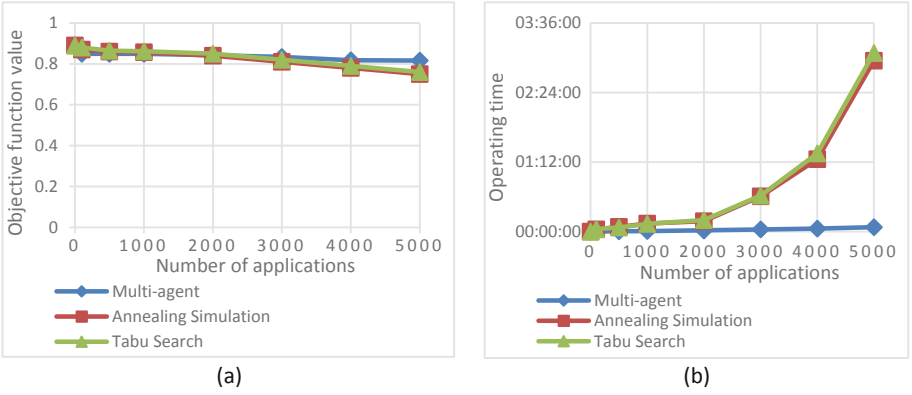

Fig. 6. Graphs of dependence of the OF value (a) and planning time (b) on the number of applications for various planning algorithms

The obtained experimental results show that the proposed method is comparable with centralized scheduling methods in terms of the scheduling quality. While with an increase in the number of planned applications, it demonstrates a more linear growth in the processing time without any loss of quality.

6 Conclusion

The authors of the paper proposed a method for adaptive planning of large-scale space observation systems based on multi-agent technologies. Results of experimental studies on the developed prototype demonstrate compliance of the presented approach with requirements for methods and tools of planning large-scale SpOS in terms of scalability, adaptability, flexibility, efficiency, reliability and survivability. As the next step, it is proposed to introduce the concepts of more advanced virtual market and ontology of space observation systems into the multi-agent system to provide the possibility of more flexible and adaptive planning settings. All these actions will ultimately create an actual platform for planning space observation systems.

Acknowledgements. The paper has been prepared based on materials of scientific research within the subsidized state theme of the Samara Federal Research Scientific Center RAS, Institute for Control of Complex Systems RAS for research and development on the topic: № AAAA-A19-119030190053-2 “Research and development of methods and means of analytical design, computer-based knowledge representation, computational algorithms and multi-agent technology in problems of optimizing management processes in complex systems”.

References

1. Shimoda, H.: Remote sensing data applications. In: Handbook of Satellite Applications, pp. 1–70 (2016)
2. Henely, S., Baldwin-Pulcini, B., Smith, K.: Turning off the lights: aAutomating SkySat mission operations. In: Small Satellite Conference (2019)
3. Irisov, V., Nguyen, V., Duly, T., et al.: Recent Ionosphere collection results from Spire’s 3U CubeSat GNSS-RO constellation, American Geophysical Union Fall Meeting (2018)
4. Kopacz, J., Herschitz, R., Roney, J.: Small satellites an overview and assessment. Acta Astronautica **170**, 93–105 (2020)
5. Wang, M., Dai, G., Vasile, M.: Heuristic Scheduling Algorithm Oriented Dynamic Tasks for Imaging Satellites. Hindawi Publishing Corporation (2014)
6. Galuzin, V., Matyushin, M., Kutomanov, A., Skobelev, P.: A review of modern methods for planning and scheduling of the operations in advanced space systems. Mekhatronika, Avtomatizatsiya, Upravlenie **21**(11), 639–650 (2020)
7. SaVoir. https://www.taitussoftware.com/products/applications/savoir. Accessed 30 Mar 2021
8. STM. https://www.stm.com.tr/en/our-solutions/satellite-and-aerospace. Accessed 30 Mar 2021
9. AGI: STK Scheduler. https://www.agi.com/products/stk-specialized-modules/stk-scheduler. Accessed 30 Mar 2021
10. Rzevski, G., Skobelev, P.: Managing Complexity. WIT Press, Boston (2014)
11. Gorodetsky, V., Skobelev, P.: System engineering view on multi-agent technology for industrial applications: barriers and prospects. Cybernet. Phys. **9**(1), 13–30 (2020)
12. Belokonov, I., Skobelev, P., Simonova, E., et al.: Multiagent planning of the network traffic between nanosatellites and ground stations. Proc. Eng. Sci. Technol. Exp. Autom. Space Veh. Small Satellites **104**, 118–130 (2015)
13. Skobelev, P., Simonova, E., Zhilyaev, A., Travin, V.: Multi-agent planning of spacecraft group for earth remote sensing. In: Borangiu, T., Trentesaux, D., Thomas, A., McFarlane, D. (eds.) Service Orientation in Holonic and Multi-Agent Manufacturing. Studies in Computational Intelligence, vol. 640, pp. 309–317 (2016)

14. Vallado, D.A.: Fundamentals of Astrodynamics and Applications, 4th edn., vol. 12. Springer, New York (2013)
15. Iacopino, C., Palmer, P., Policella, N., et al.: How ants can manage your satellites. Acta Futura **9**, 59–70 (2014)
16. He, L., Liu, X., Xing, L., Liu, K.: Hierarchical scheduling for real-time agile satellite task scheduling in a dynamic environment. Adv. Space Res. **63**(2), 897–912 (2019)
17. He, L., Liu, X., Laporte, G., et al.: An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling. Comput. Oper. Res. **100**, 12–25 (2018)
18. Peng, G., Dewil, R., Verbeeck, C., et al.: Agile earth observation satellite scheduling: an orienteering problem with time-dependent profits and travel times. Comput. Oper. Res. **111**, 84–98 (2019)
19. Niu, X., Tang, H., Wu, L.: Satellite scheduling of large areal tasks for rapid response to natural disaster using a multi-objective genetic algorithm. Int. J. Disaster Risk Reduc. **28**, 813–825 (2018)
20. Hosseinabadi, S., Ranjbar, M., Ramyar, S., et al.: Scheduling a constellation of agile Earth observation satellites with preemption. J. Qual. Eng. Prod. Optim. **2**(1), 47–64 (2017)
21. Peng, S., Chen, H., Du, C., et al.: Onboard observation task planning for an autonomous Earth observation satellite using long shortterm memory. IEEE Access **6**, 65118–65129 (2018)
22. Song, Y., Zhou, Z., Zhang, Z., et al.: A framework involving MEC: imaging satellites mission planning. Neural Comput. Appl. **32** (2019)
23. Du, Y., Wang, T., Xin, B., et al.: A data-driven parallel scheduling approach for multiple agile Earth observation satellites. IEEE Trans. Evol. Comput **24**, 679–693 (2020)
24. Bonnet, J., Gleizes, M., Kaddoum, E., et al.: Multi-satellite mission planning using a self-adaptive multi-agent system. In: Proceedings of the SASO 2015, pp. 11–20 (2015)
25. Phillips, S., Parra, F.: A case study on auction-based task allocation algorithms in multi-satellite systems. In: AIAA 2021-0185. AIAA Scitech 2021 Forum (2021)
26. Picard, G., Caron, C., Farges, J., et al.: Autonomous agents and multiagent systems challenges in earth observation satellite constellations. Proc. AAMAS **2021**, 39–44 (2021)
27. Wang, X., Wu, G., Xing, L.: Agile earth observation satellite scheduling over 20 years: formulations, methods, and future directions. IEEE Syst. J., 1–12 (2020)