Front. Eng. Manag. 2022, 9(4): 683–688
https://doi.org/10.1007/s42524-022-0222-4

COMMENTS

Yonghao DU, Lining XING, Yingguo CHEN

Satellite scheduling engine: The intelligent solver for future multi-satellite management

1 Introduction

With the rapid development and popularization of worldwide aerospace industries over the recent decades, the optimization requirements of multi-satellite management have exploded significantly. The latest data show 4852 operational satellites orbiting the earth, of which the US, China, and Russia own 2944, 499, and 169, respectively. Therefore, how to manage and schedule effectively hundreds of satellites to conform to the developing and popularizing aerospace tendency emerges as a worldwide problem.

The satellite scheduling problem is a real-world combinational optimization problem that models satellite resources and tasks and then maximizes their benefits via optimization algorithms, subject to certain constraint conditions. Compared with classical combinational optimization problems in operational research, satellite scheduling problems often show great real-world characteristics of high complexity, uncertainty, and modeling difficulty (Karapetyan et al., 2015; Wang et al., 2021; Zhang and Xing, 2022), which could hardly be addressed by mainstream optimization techniques in theoretical studies. Furthermore, the hundreds of satellites that differ in payload modes, orbital maneuver abilities, and energy using rules have made a great deal of management trouble, particularly in the scheduling system development and application. Currently, various satellite scheduling systems are independently developed and working without interaction, resulting in great difficulties in multi-satellite operation, management, and coordination. Troubled by these issues, the satellite management agencies strongly need an intelligent and overall engine to solve multi-satellite scheduling problems. Thus, the development of the engine turns into an important research and engineering topic.

In the computer field, “engine” refers to the core program or system component developed on computer platforms, such as game and search engines, which can support developers to build program and system functions quickly. In the optimization field, a “scheduling engine”, also known as a “scheduler”, refers to a general-purpose solver for certain types of optimization and scheduling problems, such as the well-known CPLEX (by International Business Machines Corporation (IBM)) and Gurobi. Despite the fact that CPLEX and some satellite scheduling solvers such as Europa2 and STK/Scheduler have been developed, they did not show qualified compatibility, extensibility, and performance that can assist in operating satellite scheduling with complex real-world constraints. Under such circumstances, to fulfil current and future requirements, satellite management agencies prefer a flexible, compatible, expandable, and easy-to-use solver with state-of-the-art algorithms, which is named “satellite scheduling engine” in this study. With this engine, multi-satellite management could be promoted efficiently and conveniently with more powerful optimization techniques and less rework, contributing to the prosperity and development of worldwide aerospace industries in the new decades.

In the remainder of this paper, the related solvers in terms of mathematical programming, spacecraft scheduling, and heuristics will be reviewed. Then, several key techniques of models, algorithms, and frameworks will

Received January 1, 2022; accepted June 12, 2022

Yonghao DU, Yingguo CHEN
College of Systems Engineering, National University of Defense Technology, Changsha 410073, China

Lining XING (✉)
School of Electronic Engineering, Xidian University, Xi’an 710075, China
E-mail: xinglining@gmail.com

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61773120 and 72201272), the Technical Field Foundation in 173 Program of National Defense Technology (Grant No. 2021-JCJQ-JJ-0049), and the Science Foundation of National University of Defense Technology (Grant No. ZK22-48).

be discussed to develop the satellite scheduling engine. A framework of this study is shown in Fig. 1, where the main opinions of the paper are outlined for smooth reading.

2 Related solvers for satellite scheduling problems

2.1 Mathematical programming solvers

Mathematical programming solvers, such as IBM CPLEX, Gurobi, FICO Xpress, and Cardinal Optimizer, can model the satellite scheduling problems as linear and other programming problems and solve them by efficient built-in algorithms such as Simplex and branch-and-bound. Among those solvers, CPLEX developed by IBM is best-known and has been adopted in many satellites scheduling studies. It is concise, easy to understand, and compatible with mainstream programming languages such as C++, Java, and Python. In many benchmark problems of operational research, CPLEX keeps the best records.

However, mathematical programming solvers could hardly tackle complex nonlinear, large-scale, and over-constrained satellite scheduling problems with real-world aerospace characteristics. For example, Xiao et al. (2019) formulated the satellite imaging and downlink scheduling problems as a flexible job-shop model, but CPLEX spent at least 7200 s solving this model with only 20 tasks. Liu et al. (2017) simplified and linearized the ladder time constraints of agile satellite orbital maneuver, but CPLEX can only address 12 tasks at most. Wang et al. (2010) and Xu et al. (2016) combined CPLEX with local search heuristics, but the 100-task satellite scheduling still took up to 1800 s. Thus, mathematical programming solvers could not efficiently address real-world satellite scheduling problems within acceptable running time, especially as the size and complexity of these problems would be continuously increased in the recent future.

2.2 Spacecraft-oriented solvers

Some spacecraft scheduling software has been developed and applied to solve satellite scheduling problems, such as Europa2 and STK/Scheduler. Europa2 is the second

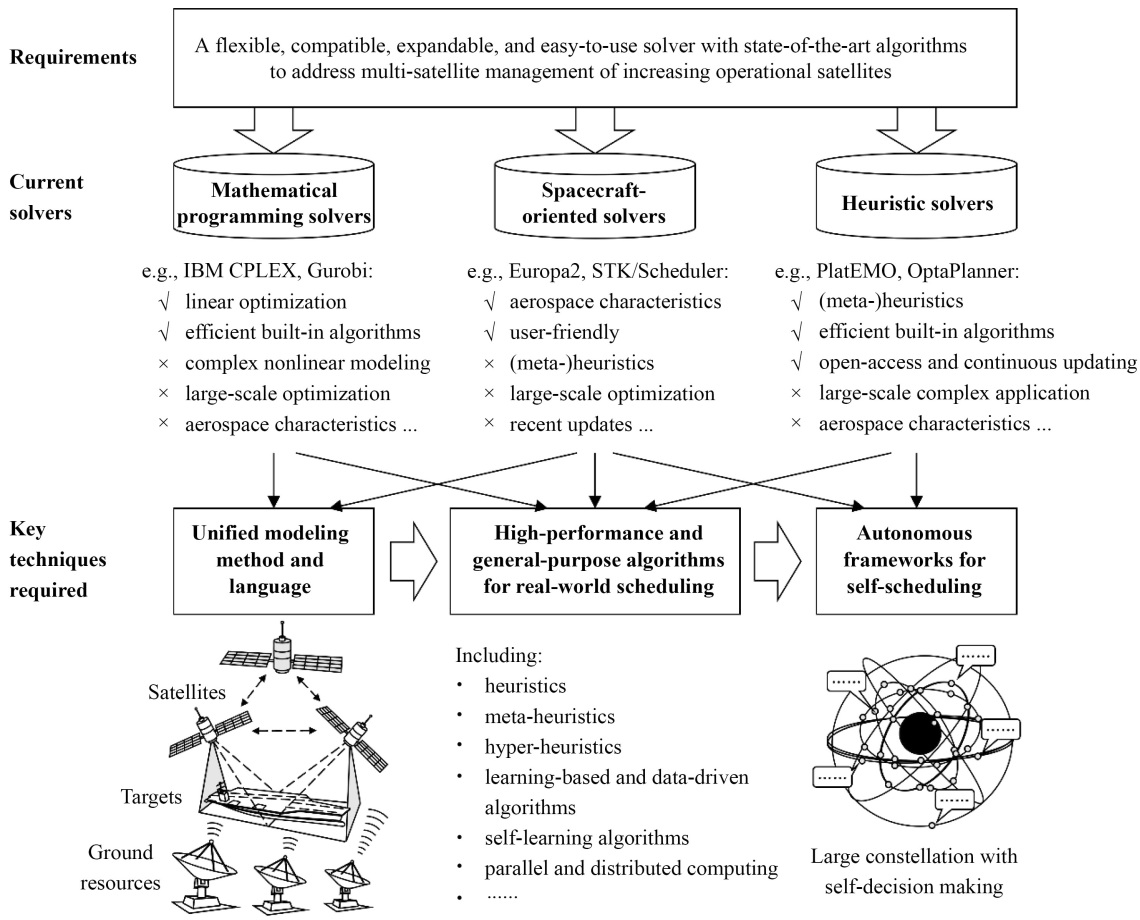

Fig. 1 Summary of current solvers for satellite scheduling problems and key techniques required to develop a satellite scheduling engine.

generation of extensible universal remote operations architecture developed by the US National Aeronautics and Space Administration (NASA) for spacecraft planning problems. It integrates new domain description language and constraint propagation techniques and has been applied to some operating satellite systems (Muscettola et al., 1998; Tran et al., 2004; Liu et al., 2012). However, Europa2 was initially designed for spacecraft action planning problems that arranges action sequences to complete short-term missions rather than the long-term spacecraft scheduling problems that properly allocate resources, including payloads, energy, and data storage space over time. To address the multi-satellite scheduling problems, Europa2 shows the following: 1) It lacks the optimization mechanism that iteratively produces better solutions driven by certain objective functions. 2) The constraint propagation techniques built in Europa2 are only efficient in small-size problems in real-world applications. 3) The latest version of Europa2 (2.6) was released in 2011 without continuous updating. Despite that NASA released a new open-access scheduling and planning interface for exploration (OpenSPIFe) in 2015 based on Europa2, few further developments or applications occurred in recent years.

STK/Scheduler is a general-purpose solver developed by the US Orbit Logic Corporation for satellite scheduling problems. As a plug-in of Space Tool Kit (STK), it can easily construct satellite scheduling models with STK data and solve them using built-in algorithms. Orbit Logic describes STK/Scheduler as a solver "for any type of space system scheduling", and attributes its advantages to the "unique approach to task and resource definition along with the powerful algorithm implementation". With this solver, Li et al. (2012) modeled the relay satellite scheduling problem considering some basic constraints. Bai et al. (2010) pointed out that multi-pass and random algorithms among the five built-in algorithms outperformed in his STK/Scheduler application scenarios. Gokhale et al. (2019) applied STK/Scheduler to perform communication scheduling among US Landsat satellites. Meanwhile, practical applications indicate that: 1) Although STK/Scheduler is user-friendly and standardized, it is still difficult to address complex real-world modeling and secondary developments. 2) It lacks the dynamic-response function that adjusts tasks, time windows, payloads, and satellites. 3) The built-in algorithms exclude current mainstream optimization (meta-) heuristics, such as Tabu search, simulated annealing, evolutionary algorithms, and memetic algorithms, and the user's guide has not been updated since 2006. Nonetheless, an online version of STK/Scheduler was released (Herz et al., 2013) and applied to satellite range scheduling problems, where the core functions such as scenario modeling and scheduling can be accessed online by remote users, presenting a new way for the future applications of satellite scheduling solvers.

2.3 Heuristic solvers

With the recent development of (meta-)heuristic algorithms, some general-purpose heuristic solvers, such as PlatEMO and OptaPlanner, are developed for addressing combinational optimization problems. PlatEMO is an open-access evolutionary multi-objective optimization platform developed by Anhui University and University of Surrey (Tian et al., 2017) in Matlab. It integrates more than 150 algorithms and continuously updates state-of-the-art algorithms. With PlatEMO, Du et al. (2019) and Wei et al. (2021) modeled the simplified satellite range scheduling and imaging scheduling problems and solved them by built-in algorithms. However, current PlatEMO applications based on evolutionary algorithms have not shown enough applicability and effectiveness in addressing complex real-world satellite scheduling problems. Thus, more algorithmic studies and examinations are required.

Different from PlatEMO, OptaPlanner is developed in Java based on another kind of heuristics, local search algorithms, including hill climbing, Tabu search, simulated annealing, and late acceptance. It is also open-access and frequently updated to improve its accessibility and performance in real-world applications. The latest OptaPlanner released in 2022 integrates over 30 well-designed optimization demos, such as vehicle routing, employee rostering, conference scheduling, and school timetabling, but the aerospace application area has not been involved. Despite the current application issues of PlatEMO and OptaPlanner, these kinds of heuristic solvers with state-of-the-art algorithms show great potential and achievable solutions for the multi-satellite management discussed in this paper.

3 Key techniques in the development of the satellite scheduling engine

Regarding the complementary advantages and shortcomings of the solvers mentioned above and to implement the flexible, compatible, expandable, and easy-to-use satellite scheduling engine with state-of-the-art algorithms, some new techniques ought to be developed, including but not limited to the following.

3.1 Unified modeling method and language for satellite scheduling problems

Unified modeling methods and languages are necessary to describe, model, and solve problems in a compatible and easy-to-use manner. Du et al. (2021) indicated that many satellite scheduling models have great similarities, concentrating on the visibility among tasks and resources in domains of time, space, or frequency. Thus, the unified modeling can fit well and greatly assist in the efficient re-use of satellite scheduling basic models. For example, in different satellite scheduling problems, users can easily format targets, satellites, payloads, orbits, and time windows (including the relationship among them) via a unified modeling method and language and construct standardized combinational optimization models in a user-friendly way. Nonetheless, given the lack of special characteristics of combinational optimization and satellite/spacecraft applications, commonly used unified modeling languages are difficult to model complex real-world satellite scheduling problems. Consequently, developing a unified modeling method and language is necessary. They integrate satellite scheduling and combinational optimization characteristics, providing standardized and general-purpose modeling basis for further algorithmic development for the satellite scheduling engine.

3.2 High-performance and general-purpose algorithms for real-world satellite scheduling

High-performance optimization algorithms are also required, especially in real-world and large-scale satellite management scenarios, to solve the satellite scheduling problems modeled by the unified modeling method and language. Despite many well-designed algorithms with good performance in previous studies, such as local search (meta-)heuristics (Luo et al., 2017; He et al., 2018), evolutionary algorithms (Song et al., 2020; Chen et al., 2021), and memetic algorithms (Du et al., 2019), few of them have been applied to operating satellites because the real-world complexity and constraints cannot be ignored or simplified. For example, Peng et al. (2019) designed several efficient algorithm operators when they studied that an agile imaging satellite scheduling problem could be simplified as an orientating problem. However, those problem-dependent operators probably cannot work well considering the downlink and onboard data-erasing optimization required by operating satellites. Furthermore, the increasing operational satellites cause "combinatorial explosion" and require state-of-the-art algorithms with higher performance and lower time consumption. The algorithms include but are not limited to heuristics, meta-heuristics, hyper-heuristics, learning-based and data-driven algorithms, self-learning algorithms, and parallel and distributed computing. Developing high-performance and general-purpose algorithms for satellite scheduling problems, especially for real-world and large-scale problems often omitted in previous studies, is of great importance in implementing the satellite scheduling engine for operating satellite systems.

3.3 Autonomous frameworks and algorithms for future large constellation self-scheduling

When the operating satellite number increases to a certain extent, such as the thousands of satellites in the current StarLink constellation, conventional ground-based scheduling frameworks would be incompetent to support the large constellation scheduling with acceptable computing resources and time. Under such circumstances, new autonomous satellite scheduling frameworks that perform thousands of satellite self-scheduling, negotiation, organization, and decision-making would be requisite. Recent studies (Zheng et al., 2019; Yang et al., 2021) have clarified the autonomous satellite scheduling tendency and made efforts on some small constellations with special configurations or purposes. However, few show general-purpose solutions for future large constellations where many technical difficulties need to be addressed. Besides, the autonomous frameworks and algorithms ought to be developed via the unified modeling method and language for satellite scheduling problems mentioned above; otherwise, the autonomous scheduling and management of future constellations could not be unified and compatible with each other. Autonomous satellite scheduling should also depend on the aforementioned high-performance optimization algorithms given that the self-scheduling of certain satellites onboard still belongs to combinational optimization problems and requires further development of state-of-the-art optimization algorithms.

3.4 A modular system framework to support and integrate the developing key techniques

With the developing key techniques, a flexible, compatible, expandable, and easy-to-use system framework is finally needed for the satellite scheduling engine development and application. According to the studies and application experience over the years, a preliminary framework that decouples the modeling, constraints, and optimization in satellite scheduling problems is given in Fig. 2. The framework includes three main modulars: 1) Modeling modular applies the unified modeling method and language to format the scheduling problem for combinational optimization. In this modular, the inputted tasks and resources will be generally modeled with proper decision variables first, where all the real-world problem data can be instantiated and interlinked. Thus, the solutions can be encoded into a numerical format and function as interfaces so real-world problem data can be freely accessed via neighborhood structures in the optimization loop over time. 2) Constraints modular reports the feasibility and objective of given solutions via interfaces and provides important criteria for optimization algorithms. The constraints, especially those diverse and complex constraints, are separated from the modeling modular herein because they have made a great deal of trouble and must be managed and processed in an independent and expandable manner. 3) Optimization modular adopts the high-performance and autonomous algorithms and other optimization techniques to modify (the values of) decision

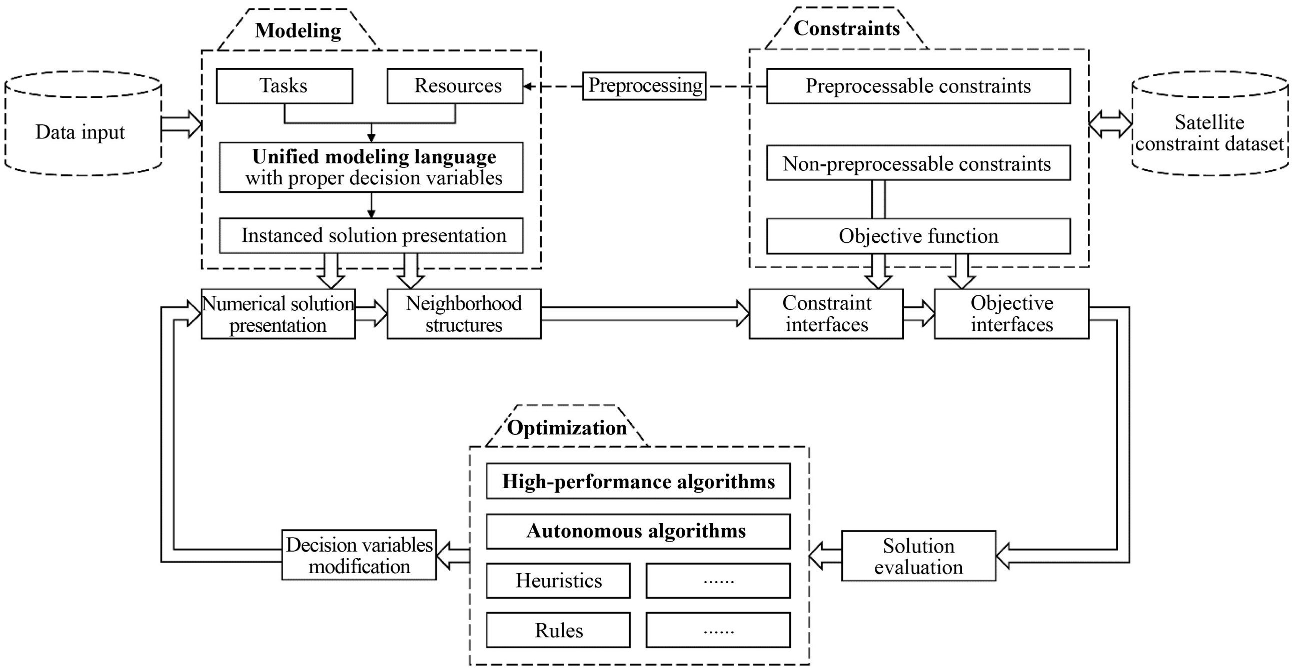

Fig. 2 A preliminary system framework for the satellite scheduling engine.

variables intelligently, reproduce neighboring solutions iteratively, and output satellite scheduling results. With feedback from the constraints and objective function, a closed optimization loop can be implemented among the decoupled modulars in this framework. Based on this kind of framework, the developing modeling and optimization techniques can be flexibly configured in the satellite scheduling engine to support overall and efficient multi-satellite scheduling in the recent future.

4 Conclusions

This study proposes to develop an intelligent solver called a satellite scheduling engine to address the overall and efficient multi-satellite management of increasing operational satellites. This study made the following contributions. First, the necessity to develop the satellite scheduling engine under current and future multi-satellite management background is clarified. Second, the advantages and shortcomings of the current three types of solvers for satellite scheduling problems are reviewed. Finally, three key techniques required and a preliminary system framework to implement this engine are discussed and illustrated.

Future work should include more efforts to address technical difficulties to develop and implement the satellite scheduling engine's key techniques and system frameworks. Also, the techniques and frameworks should be verified by more real-world satellite scheduling problems rather than simplified problems. With these development and future applications, multi-satellite management would be efficiently and conveniently conducted with more powerful optimization techniques and less rework, contributing to the prosperity and development of worldwide aerospace industries in the new decades.

References

Bai J P, Yan H, Gao Y M, Wang Z M (2010). Application of space mission scheduling based on STK/Scheduler. Journal of Equipment Academy, 21(3): 71–75 (in Chinese)

Chen M, Wen J, Song Y J, Xing L N, Chen Y W (2021). A population perturbation and elimination strategy based genetic algorithm for multi-satellite TT&C scheduling problem. Swarm and Evolutionary Computation, 65: 100912

Du Y H, Xing L N, Yao F, Chen Y G (2021). Survey on models, algorithms and general techniques for spacecraft mission scheduling. Acta Automatica Sinica, 47(12): 2715–2741 (in Chinese)

Du Y H, Xing L N, Zhang J W, Chen Y G, He Y M (2019). MOEA based memetic algorithms for multi-objective satellite range scheduling problem. Swarm and Evolutionary Computation, 50: 100576

Gokhale N, Callis K, Herz E, Bishop R (2019). Mission planning and scheduling software for Landsat 8/9. Online Paper

He L, Liu X L, Laporte G, Chen Y W, Chen Y G (2018). An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling. Computers & Operations Research, 100: 12–25

Herz A F, Stoner F, Hall R, Fisher W (2013). SSA sensor tasking approach for improved orbit determination accuracies and more efficient use of ground assets. In: Proceedings of the Advanced

Maui Optical and Space Surveillance Technologies Conference. Maui, HI: The Maui Economic Development Board, E74

Karapetyan D, Mitrovic Minic S, Malladi K T, Punnen A P (2015). Satellite downlink scheduling problem: A case study. Omega, 53: 115–123

Li Y X, Liu Y, Fang Q (2012). Realization of TDRSS mission scheduling based on STK/Schedule. Modern Electronics Technique, 35(10): 122–125 (in Chinese)

Liu X L, Laporte G, Chen Y W, He R J (2017). An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time. Computers & Operations Research, 86: 41–53

Liu Y C, Zhong X Y, Fang Y S, Chen J B (2012). Dynamic simulation and modeling for AI planning based on Europa. Computer Engineering and Applications, 48(17): 211–214, 219 (in Chinese)

Luo K P, Wang H H, Li Y J, Li Q (2017). High-performance technique for satellite range scheduling. Computers & Operations Research, 85: 12–21

Muscettola N, Nayak P P, Pell B, Williams B C (1998). Remote agent: To boldly go where no AI system has gone before. Artificial Intelligence, 103(1–2): 5–47

Peng G S, Dewil R, Verbeeck C, Gunawan A, Xing L N, Vansteenwegen P (2019). Agile earth observation satellite scheduling: An orienteering problem with time-dependent profits and travel times. Computers & Operations Research, 111: 84–98

Song Y J, Xing L N, Wang M, Yi Y, Xiang W, Zhang Z S (2020). A knowledge-based evolutionary algorithm for relay satellite system mission scheduling problem. Computers & Industrial Engineering, 150: 106830

Tian Y, Cheng R, Zhang X Y, Jin Y C (2017). PlatEMO: A Matlab platform for evolutionary multi-objective optimization. IEEE Computational Intelligence Magazine, 12(4): 73–87

Tran D, Chien S, Sherwood R, Castano R, Cichy B, Davies A, Rabideau G (2004). The autonomous sciencecraft experiment onboard the EO-1 spacecraft. In: Proceedings of the 19th National Conference on Artificial Intelligence. San Jose, CA: AAAI Press, 1040–1041

Wang P, Li J F, Tan Y J (2010). Comparison of earth observation scheduling model for satellite formation. Systems Engineering and Electronics, 32(8): 1689–1694 (in Chinese)

Wang X W, Wu G H, Xing L N, Pedrycz W (2021). Agile earth observation satellite scheduling over 20 years: Formulations, methods, and future directions. IEEE Systems Journal, 15(3): 3881–3892

Wei L N, Chen Y N, Chen M, Chen Y W (2021). Deep reinforcement learning and parameter transfer based approach for the multi-objective agile earth observation satellite scheduling problem. Applied Soft Computing, 110: 107607

Xiao Y Y, Zhang S Y, Yang P, You M, Huang J Y (2019). A two-stage flow-shop scheme for the multi-satellite observation and data-downlink scheduling problem considering weather uncertainties. Reliability Engineering & System Safety, 188: 263–275

Xu R, Chen H P, Liang X L, Wang H M (2016). Priority-based constructive algorithms for scheduling agile earth observation satellites with total priority maximization. Expert Systems with Applications, 51(1): 195–206

Yang W Y, He L, Liu X L, Chen Y W (2021). Onboard coordination and scheduling of multiple autonomous satellites in an uncertain environment. Advances in Space Research, 68(11): 4505–4524

Zhang J W, Xing L N (2022). An improved genetic algorithm for the integrated satellite imaging and data transmission scheduling problem. Computers & Operations Research, 139: 105626

Zheng Z X, Guo J, Gill E (2019). Distributed onboard mission planning for multi-satellite systems. Aerospace Science and Technology, 89: 111–122