

Feng YAO, Yonghao DU, Lei LI, Lining XING, Yingguo CHEN

General modeling and optimization technique for real-world earth observation satellite scheduling

© Higher Education Press 2023

Abstract Over the last two decades, many modeling and optimization techniques have been developed for earth observation satellite (EOS) scheduling problems, but few of them show good generality to be engineered in real-world applications. This study proposes a general modeling and optimization technique for common and real-world EOS scheduling cases; it includes a decoupled framework, a general modeling method, and an easy-to-use algorithm library. In this technique, a framework that decouples the modeling, constraints, and optimization of EOS scheduling problems is built. With this framework, the EOS scheduling problems are appropriately modeled in a general manner, where the executable opportunity, another format of the well-known visible time window per EOS operation, is viewed as a selectable resource to be optimized. On this basis, 10 types of optimization algorithms, such as Tabu search and genetic algorithm, and a parallel competitive memetic algorithm, are developed. For simplified EOS scheduling problems, the proposed technique shows better performance in applicability and effectiveness than the state-of-the-art algorithms. In addition, a complicatedly constrained real-world benchmark exemplified by a four-EOS Chinese commercial constellation is provided, and the technique is qualified and outperforms the in-use scheduling system by more than 50%.

Keywords earth observation satellite, scheduling, general technique, optimization algorithm, commercial constellation, real-world, benchmark

1 Introduction

Recent decades have witnessed rapid developments in satellite technology. In the areas of resource exploration, environment protection, disaster warning, and military applications, earth observation satellites (EOSs) play an increasingly important role. Management agencies must closely monitor the EOSs and appropriately schedule their tasks (Du et al., 2021; Yang, 2021; He et al., 2022). EOS scheduling has become the most troubling problem for management agencies given that the numbers of EOSs and tasks increase over the years.

The EOS scheduling problem is a combinatorial optimization problem that selects and arranges appropriate EOS tasks given a task set and certain constraints with the objective of maximizing task profits. A general modeling and optimization technique that can address different EOS scheduling problems is necessary because the EOSs of a management agency often differ in types and constraints. Otherwise, different EOS scheduling problems should be independently modeled and optimized in different systems, resulting in a “one satellite, one system” phenomenon. For example, China Center for Resources Satellite Data and Application has many types of EOSs, such as “Ziyuan”, “Gaofen”, and “Shijian”, and the agency must operate a mass of systems every day to independently schedule these EOSs, thereby increasing the workload and restricting the advantages of multi-EOS collaboration.

Although a number of modeling and optimization techniques have been developed for EOS scheduling problems, few of them show good generality for real-world applications mostly due to 1) over-simplified modeling of EOS scheduling problems and 2) seldom general-purpose development of EOS scheduling models and algorithms.

Received Dec. 13, 2022; revised May 17, 2023; accepted Jun. 5, 2023

Feng YAO, Yonghao DU (✉), Lei LI, Yingguo CHEN
College of Systems Engineering, National University of Defense Technology, Changsha 410073, China
E-mail: duyonghao15@163.com

Lining XING
School of Electronic Engineering, Xidian University, Xi'an 710075, China

This work was supported by the National Natural Science Foundation of China (Grant No. 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).

First, many models in previous studies were over-simplified to classical combinational optimization models; hence, they are unable to completely describe real-world EOS scheduling problems. For example, inspired by the vehicle routing problem (VRP) and orienteering problem (OP), Cordeau and Laporte (2005) and Peng et al. (2019; 2020) modeled the EOSs, targets, and their visible time windows (VTWs) as the vehicle (visitor), cities (vertices), and time windows in the VRP (OP), respectively. Inspired by the job-shop scheduling problem (JSP), Xiao et al. (2019) modeled the tasks and EOSs as the jobs and machines in the JSP, respectively. Although the VRP, OP, and JSP models can describe the EOS scheduling problems in an easy-to-understand manner, some simplifications in those models were overused and can hardly fit real-world applications. Those models simplified the decision process in real-world EOS scheduling problems; typically, only considered the EOS imaging scheduling but ignored, greatly simplified, or separately considered the downlink scheduling. The EOS imaging scheduling and downlink scheduling are equally important and highly coupled, and they must be simultaneously modeled. In addition, previous models often over-simplified real-world constraints. Although some simplifications are commonly used in many combinational optimization problems, real-world EOS scheduling problems are always strictly constrained; one-second time or one-byte data that violate the constraints are not allowed at all. Some well-designed and outperforming EOS scheduling algorithms cannot be transplanted or engineered for real-world use due to these over-simplifications. For example, Peng et al. (2019) proposed a dynamic programming (DP)-based local search algorithm, but the operators were designed following the first-in-first-out (FIFO) principle of an OP model that only described the EOS imaging scheduling with a few constraints. Peng et al. (2020) also proposed an exact algorithm but could only address the further simplified single-orbit EOS scheduling. In fact, the studied and many other related EOSs are still scheduled by their original systems without the proposed state-of-the-art algorithms. Therefore, despite that the previous studies brought many novel ideas, much further effort is still required to overcome the over-simplifications to promote model and algorithm applications to real-world EOS scheduling problems.

The second factor that limits the generality of previous modeling and optimization techniques for EOS scheduling is the neglect of the common characteristics among EOS scheduling problems, aiming for special cases rather than for general purpose. In addition to the VRP, OP, and JSP models mentioned above, some researchers modeled the EOS scheduling as the constraint satisfaction problem (CSP). For example, Jang et al. (2013), Wu et al. (2013), and Wang et al. (2020) viewed the VTWs of each task as the selectable resources and used decision variables to determine the selected VTW per task. Furthermore,

different models were raised to address agile EOS scheduling, where the task begin time within the selected VTW should be determined given that the VTW becomes much longer than the time required per task. For example, Lemaître et al. (2000), Liu et al. (2017), Chu et al. (2017), and Mok et al. (2019) determined this additional begin time as early as possible. Nag et al. (2018), He et al. (2018), Zhu et al. (2019), Valicka et al. (2019), and Du et al. (2020) adopted other decision variables or discretized the VTW into shorter ones that match the time required so that the sub-VTW selection can finally determine the begin time. A similarity can also be observed among various modeling methods in previous studies; the modeling was performed on the basis of EOS-task VTWs. The VTWs not only function as constraints, but also serve as important resources that represent the opportunities to execute tasks for certain EOSs in certain orbits. Thereby, with the resource characteristics of the VTWs, a general approach to support different real-world EOS scheduling modeling can be sought.

In addition, for long-term EOS management, some other real-world requirements and expectations from EOS management agencies should be considered. For example, the constraints with respect to a certain EOS may be changed over time, and new optimization algorithms may be developed. Certainly, using only one algorithm throughout the 10-year or 20-year lifetime of an EOS is impractical. Thus, an ideal modeling and optimization technique for real-world EOS scheduling problems should be appropriately structured to update constraints and algorithms. An algorithm library that continually absorbs the algorithms for EOS scheduling also makes sense; it could contribute to incremental algorithm studies and feasible algorithm applications (Du et al., 2022). Lastly, recent developments in computer hardware have motivated many algorithmic studies in a parallel and competitive manner, where computing resources can be fully used. For EOS management agencies, the use of algorithms that leave many computer threads unused in real-world applications is ineffective.

With these motivations, a general modeling and optimization technique for real-world EOS scheduling problems is developed. In this technique, a framework that decouples the modeling, constraints, and optimization of EOS scheduling problems is built, and the relationships among them are explained. Then, the EOS scheduling problems are appropriately modeled in a general manner, where the executable opportunity (EO), another format of the VTW per EOS operation, is viewed as a selectable resource to be optimized. On this basis, 10 types of optimization algorithms, as well as a parallel competitive memetic algorithm (PC-MA), are developed. The technique is then examined on four EOS scheduling cases, including three simplified benchmarks and a real-world dataset exemplified by a famous four-EOS Chinese commercial constellation called SuperView-1. The

developed technique shows good performance in applicability and effectiveness in these case studies.

2 Framework

Prior to structuring the framework of the general modeling and optimization technique for EOS scheduling problems, a premise about the role of this technique should be claimed. The acknowledged four components of a scheduling problem, including tasks, resources, constraints, and the objective, should be provided because the technique is a special tool only for scheduling, as shown in Fig. 1. Other issues that occur in EOS management, such as task generation and resource provision, are not studied here.

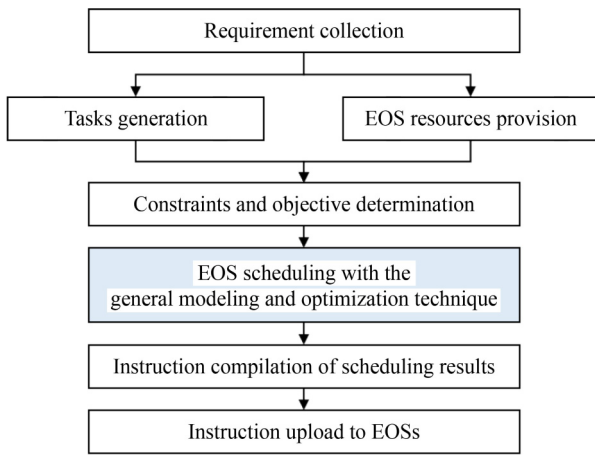


Fig. 1 Role of the general modeling and optimization technique in a brief EOS management flow.

In other words, the data inputs of the general modeling and optimization technique must contain those components of an EOS scheduling problem, and the outputs are the organized tasks and resources that satisfy the constraints with the optimized objective as much as possible.

To appropriately address the general modeling and optimization technique for EOS scheduling problems, an overall framework that decouples the modeling, constraints, and optimization is structured, as shown in Fig. 2. Contrary to the integrated organization in previous studies and engineering systems, the modeling, constraints, and optimization of EOS scheduling problems are independently organized here. More details are explained as follows.

• **Modeling.** With the inputs of an EOS scheduling problem, this process appropriately formats the real-world problem data and determines the decision variables for combinational optimization. The real-world problem contains a large amount of data about tasks and resources (such as EOSs, payloads, orbits, and tracking stations), thereby causing difficulties in the modeling process. As shown in Fig. 2, the inputted tasks and resources are generally modeled with appropriate decision variables, where all the data are instantiated and interlinked. This process presents real-world EOS scheduling solutions with instanced data and will be introduced in Section 3.1. Furthermore, the solutions are encoded into a numerical format, which will be introduced in Section 3.2. It can be seen in the figure that the numerical solution presentation and neighborhood structures function as the interfaces from real-world problem data to freely access data in the optimization loop.

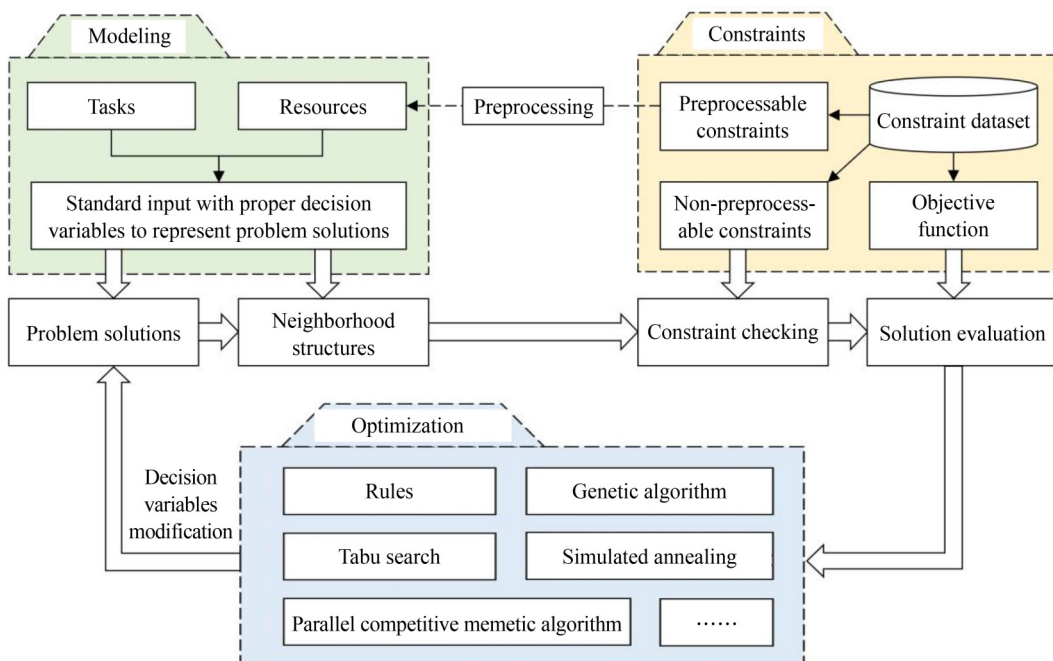


Fig. 2 Framework of the general modeling and optimization technique for EOS scheduling problems.

• **Constraints.** The constraints report the feasibility and objective of a given solution; they are the important criteria for optimization algorithms. As shown in Section 3.3, some constraints can be preprocessed, such as the visibility-related EOS limits and user requirements. If any resource certainly violates the constraints when selected, then it is directly filtered and never selected. In addition, non-preprocessable constraints affected by different combinations of tasks and resources remain in optimization. Hence, an interface that reports these constraints is provided here using the model that has free access to the problem data. The objective function is also viewed as a constraint. The objective function can be viewed as the softest one that must be satisfied as much as possible given that the constraints may have different priorities, such as the hard ones that must be satisfied and some soft ones that can be violated in certain degrees.

• **Optimization algorithms.** Based on the solution and neighborhood structures, an optimization algorithm can modify (the values of) decision variables and reproduce neighboring solutions. This approach can evaluate the reproduction and work iteratively with feedback from the constraints and objective function. Thus, as shown in Fig. 2, a closed optimization loop is structured in the decoupled framework. An algorithm library that includes 10 types of algorithms is presented in Section 4.2, providing convenience to EOS management agencies. In addition, a PC-MA is designed to further strengthen the optimization ability, and it is introduced in Section 4.3.

With this framework, the decoupled modeling, constraints, and optimization algorithms can be flexibly configured as EOS management agencies, addressing various types of EOS scheduling problems that differ in EOS types and constraints in a general manner.

3 Modeling method

The modeling method determines the applicability as an important methodology in the decoupled framework. In this section, EOS scheduling problems are described in a general manner initially. Then, the model encoding with

appropriate decision variables is provided. Moreover, real-world constraints that should be considered are introduced and cataloged.

3.1 Problem description

The EOS scheduling problem is a type of combinatorial optimization problem, which selects and arranges appropriate EOS tasks given a task set and certain constraints with the objective of maximizing task profits. In this regard, the first step of problem description is to describe the EOS tasks.

To satisfy the real-world requirements in EOS scheduling, an EOS task is described into three general operations, as shown in Fig. 3: 1) The imaging operation, which aims to image the required target by the payload(s) equipped on the EOS. This operation is singled out because it can be completed by an EOS independently, without simultaneous data communication with ground stations. 2) The downlink operation, which aims to build simultaneous data communication between the EOS and tracking stations, including ground-based stations and the relay satellites that function as space-based stations. 3) The memory-erasing operation, which is required by many real-world EOSs to erase the memory prior to the imaging operation required by common instruction templates. The downlink and memory-erasing operations were seldom considered in previous studies, but they are relatively important in real-world EOS scheduling and are fully considered in this study.

Different EOS working modes can be deduced with the combination of the three operations. For example, in view of an EOS and tracking station, different operation combinations and the corresponding EOS working modes are shown in Figs. 4(a) and 4(b), respectively. These working modes result in different downlink durations, constraints, or objectives, but they were seldom distinguished in previous studies.

The advantage of this description can be further observed in the following relationship among the real-world problem data in EOS scheduling, as shown in Fig. 5. In this figure, all the problem data are divided into

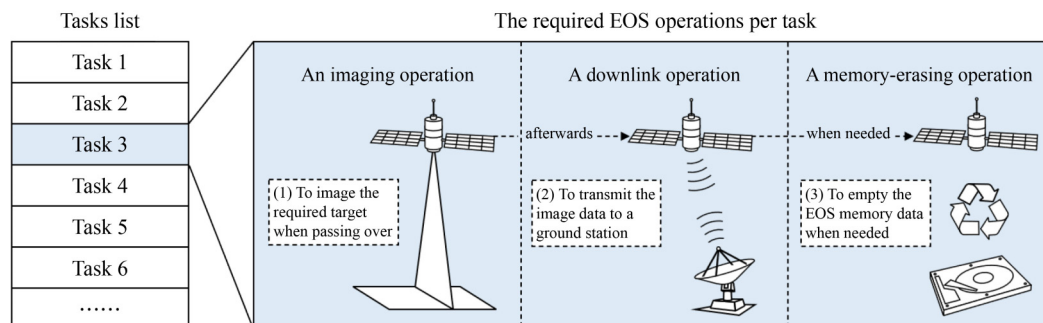


Fig. 3 Generally described EOS operations per task in EOS scheduling problems.

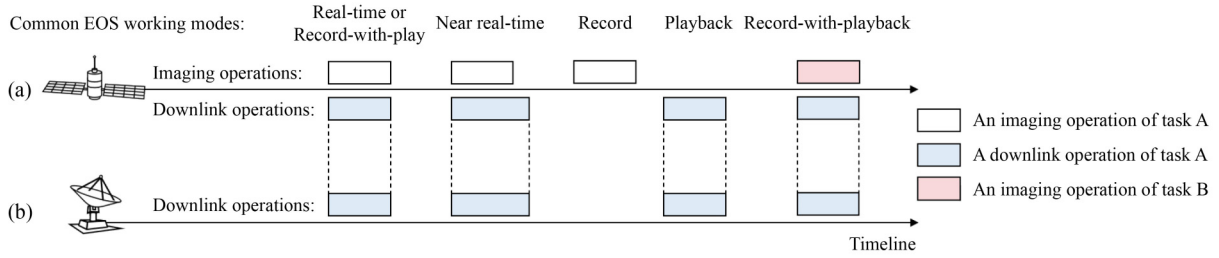


Fig. 4 Examples of EOS working modes with different combinations of imaging and downlink operations. (a) Timeline view of an EOS. (b) Timeline view of a tracking station.

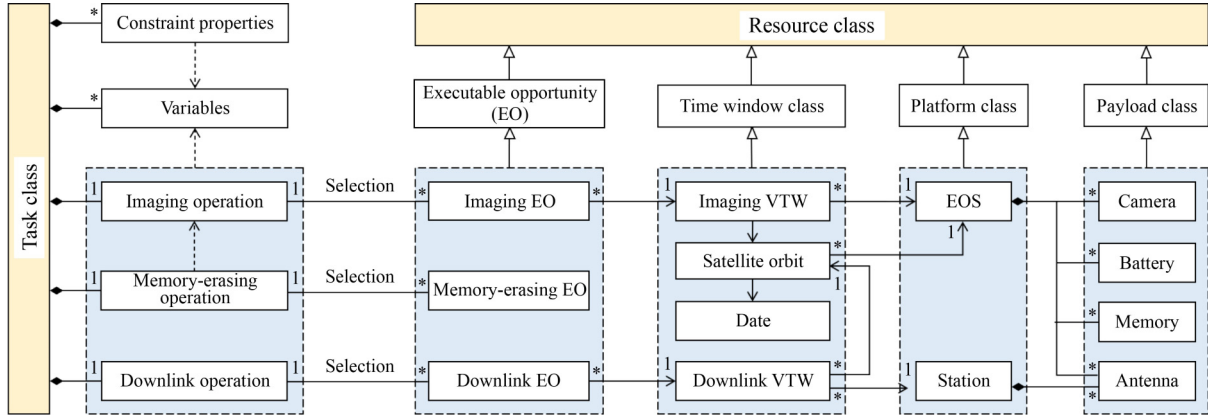


Fig. 5 Relationship among real-world problem data in EOS scheduling problems.

a task class and a resource class, the two components of a scheduling problem.

• **Task class.** The described imaging, downlink, and memory-erasing operations are involved. Then, some constant properties, such as the required imaging time and memory occupation per task, are also involved and remain unchanged throughout the scheduling process. Some other variables also depend on the constant properties, as well as the combinations of imaging and downlink operations, such as the EOS working mode, task imaging begin time, and downlink duration. These variables are updated by the given functions and used when required in the scheduling process.

• **Resource class.** The payload is the most direct resource that supports executing EOS tasks, such as the camera, battery, memory, and antenna. Then, the EOS and tracking station are required to carry those payloads, and they are cataloged to the platform class. The VTW usually functioned as an abstract resource in previous studies. Thus, a time window class that instantiates a time slot, including the imaging and downlink VTW, the EOS orbit that provides these VTWs, and the date when the scheduling is performed, is provided. Nonetheless, the begin time of an imaging or downlink operation within its VTW cannot be determined because the VTW becomes much longer. Herein, the VTW is further discretized into sub-VTWs that exactly match an imaging or downlink operation. For example, given a five-second imaging operation with a ten-second VTW, the VTW can be

discretized into six five-second sub-VTWs, as shown in Fig. 6. In this study, the discretized sub-VTW is called EO.

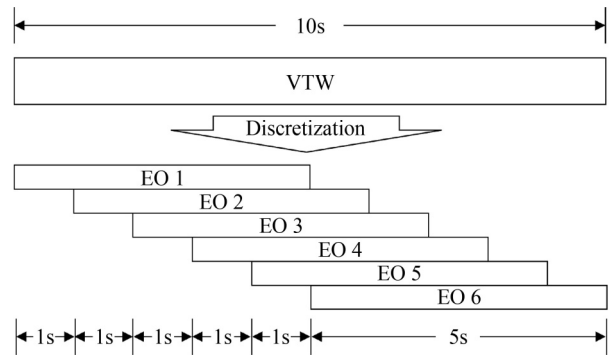


Fig. 6 Discretization from a ten-second VTW into the five-second EOs.

In Fig. 5, a single-selection relationship between the imaging, downlink, or memory-erasing operation and its EOs appears. Once an EO is selected, all real-world problem data, such as the VTW that contains this EO, the EOS that provides this VTW, and the payloads carried by this EOS, can thereby be accessed. In this manner, the EO is viewed as a selectable resource per EOS operation so that the real-world EOS scheduling is generally modeled as a combinational optimization problem.

3.2 Model encoding

With this general description and problem data relationship, the next step is to appropriately present and encode an EOS scheduling solution. The solution presentation should be clear to access problem data and beneficial for algorithmic developments. In this subsection, the model encoding is explained by a three-task EOS scheduling example, as shown in Fig. 7.

The selected EO performs as the instanced decision variable in the EOS scheduling problem because the problem data relationship has shown selectable EOs for the imaging, downlink, and memory-erasing operations per task. The three-task example shown in Fig. 7(a) indicates two and three selectable EOs for the imaging and downlink operations per task, respectively. Here, Task 1 selects the first EOs (colored in blue) for its imaging and downlink operations and erases the memory prior to imaging. Task 2 selects the second EOs for imaging and downlink and the null for memory-erasing (will not erase the memory). Task 3 does not select any EO for imaging or downlink and does not erase the memory. In this approach, an EOS scheduling solution (regardless of constraints here) can be clearly presented, and all the real-world problem data can be accessed from the selected EOs given the data relationship in Fig. 5.

Moreover, the instanced model presentation can be encoded by the normalized numbers ranging from 0 to 1, as shown in Fig. 7(b). Each [0, 1) number in the figure indicates a normalized index (from 0) of the selected EO from all the EOs and a null element. For example, 0/3 and 1/3 represent the first and second in three (two EOs and a null element), respectively, and 0/4 and 1/4 indicate the first and second in four (three EOs and a null element), respectively. If the number is set to 1, then the null element, but none of the EO, is selected. In this approach, neighboring solutions can be easily produced via a [0, 1) random-number-generation function in the programming procedure; hence, those [0, 1) numbers can effectively function as numerical decision variables and

easily encoded, cloned, and modified by algorithms in the optimization process.

In general, an n -task EOS scheduling solution can be simply represented by an $n \times 3$ decision matrix $\mathbf{D} = [d_{ij}]$, where:

- d_{i1} is the first decision variable that determines the [0, 1) normalized index of the selected EO for the imaging operation of the i th task.
- d_{i2} is the second decision variable that determines the [0, 1) normalized index of the selected EO for the downlink operation of the i th task.
- d_{i3} is the third decision variable that determines the [0, 1) normalized index of the selected EO for the memory-erasing operation of the i th task.

Suppose that m_{ij} EOs are available for the imaging, downlink, and memory-erasing operations of the i th task when j equals 1, 2, and 3, respectively. The [0, 1) normalization and denormalization between the d_{ij} and the EO selection index x_{ij} (ranging from 0 to m_{ij}) can be expressed by Eqs. (1) and (2), respectively. Herein, the normalization unifies the ranges of different decision variables so that the proposed general-purpose algorithms and neighborhood operator can be conveniently performed without out-of-range risks. In this approach, decision matrix \mathbf{D} with [0, 1) variables can work as an interface between the optimization model and algorithms so that they can be decoupled appropriately.

$$d_{ij} = \frac{x_{ij}}{m_{ij} + 1}, \quad 1 \leq i \leq n, \quad j = 1, 2, 3, \quad (1)$$

$$x_{ij} = \text{floor}(d_{ij} \cdot (m_{ij} + 1)), \quad 1 \leq i \leq n, \quad j = 1, 2, 3. \quad (2)$$

The model encoding in this subsection shows how to program the general modeling for real-world EOS scheduling problems. Notably, in cases of the oversimplified EOS scheduling problems that only consider imaging operations in previous studies, no downlink or memory-erasing operation exists; decision variables d_{i2} and d_{i3} can be fixed to 1 and never work. The solution

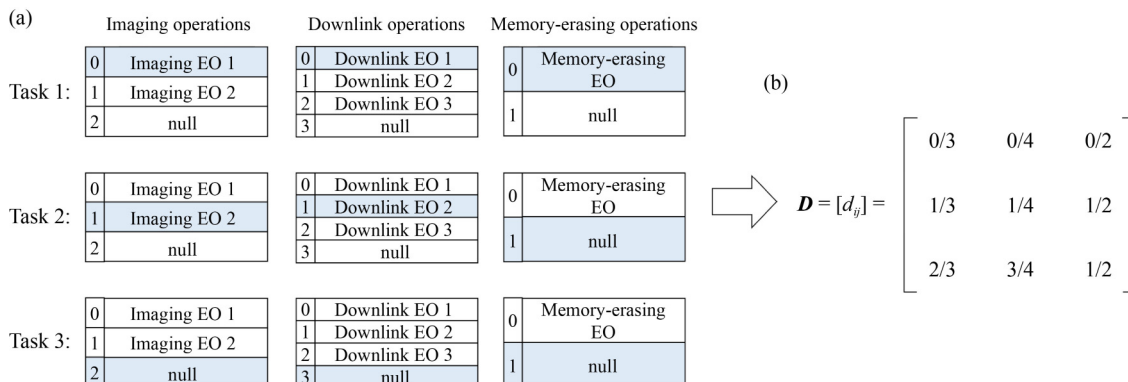


Fig. 7 Model encoding example for real-world EOS scheduling problems. (a) Instanced decision variable presentation. (b) Numerical decision variable presentation by a decision matrix.

presented here is not always feasible; hence, the constraints should report the feasibility and objective value of a given solution and offer important criteria for optimization algorithms.

3.3 Constraints and the objective

For EOS scheduling problems, the constraints function as a reporter between the solution and optimization algorithms with the decoupled framework for the general modeling and technique. Real-world EOS scheduling problems have various demands, requirements, limits, and preferences from EOS management agencies and users, which are considered in formats of constraints. For better use and incremental management of the various constraints, an elementary catalog is provided and explained in this subsection, as shown in Table 1.

Initially, the constraints are divided into the preprocessable and non-preprocessable catalogs to perform appropriate preprocessing prior to optimization. As shown in the model encoding, the preprocessing serves to filter the EOs that certainly violate the constraints, which should always be satisfied in optimization. Hereby, only the visibility-related constraints that affect the EOs are preprocessable. The visibility-related constraints also differ remarkably, and the table shows some typical examples. In real-world applications, EOS users often make requirements in real-world problems; for example, the task must be completed in a time slot by the required EOSs or tracking stations and in a special imaging angle. They can still be viewed as visibility-related constraints and can therefore be preprocessed by filtering the EOs. Here, according to the definition of the EO in Section 3.1, the selected EOs for imaging, downlink, or memory-erasing operations certainly satisfy the visibility constraint. The $[0, 1)$ normalized decision variable d_{ij} in Section 3.2 can appropriately determine the EO selection. Thus, this

constraint can be addressed as long as decision variable d_{ij} ranges according to Constraint (3) as follows:

$$0 \leq d_{ij} < 1, \quad 1 \leq i \leq n, \quad j = 1, 2, 3. \quad (3)$$

The remaining constraints cannot be preprocessed because they will be affected by different EO combinations. Hence, they determine the feasibility of a given solution and offer important criteria for optimization algorithms. As shown in Table 1, these constraints are further divided into logicity, sequence, transition, memory, and EOS protection, according to the real-world practice in EOS scheduling problems. These constraints are applicable to most real-world EOSs. However, some differences are observed in special cases. More details about the listed constraints can be accessed from the benchmark of a famous four-EOS Chinese commercial constellation called SuperView-1, which is discussed in Section 5.

A satellite scheduling solution can be judged as feasible or not according to the constraints listed in the table. As shown in Fig. 8, with the input of decision matrix $\mathbf{D} = [d_{ij}]$ that represents a scheduling solution of satellite imaging, downlink, and memory-erasing operations, the constraint check module initially works. It checks the non-preprocessable constraints shown in Table 1 one by one and outputs whether all the constraints are satisfied (true or false). If some constraints are unsatisfied in this procedure, then the conflicting tasks related to the unsatisfied constraints are output, thereby assisting in the deconflicting operation required by certain algorithms. In addition, an objective calculation module works on decision matrix \mathbf{D} . It calculates the objective function value of the scheduling solution to quantitatively evaluate its performance. Here, the objective function of the satellite scheduling problem includes the total profits of the successfully scheduled tasks in the solution represented by \mathbf{D} , similar to those in many combinational optimization problems.

Table 1 Elementary constraint catalog of EOS scheduling problems

Catalog	Sub-catalog	Explanation	
Preprocessable constraints	Visibility	The VTWs caused by EOS-task locations	
		The VTWs caused by EOS-task frequential matches	
		The VTWs caused by the abilities of EOSs, tracking stations, and their payloads	
		The VTWs caused by task requirements	
		The VTWs caused by other factors (sun, clouds, and latitudes)	
Non-preprocessable constraints	Logicity	An imaging or downlink operation can be executed once at most	
		An imaging and downlink operation must be paired per task	
		A downlink operation under certain EOS working modes must be executed within the VTW	
	Sequence	Some imaging and downlink operations must be executed in ordered sequences	
	Transition	Transition time is required between the executed operations per EOS	
		Transition time is required between the imaging and downlink operations per task	
	Memory	The stored data in the EOS memory cannot exceed the limit	
		The stored data must be downloaded prior to memory-erasing	
	EOS protection	EOS protection	Working frequency and time limits for EOS cameras per orbit and/or day
			Working frequency and time limits for EOS antennas per orbit and/or day
Maneuver frequency and time limits for EOSs per orbit and/or day			
Continuous maneuver frequency and time limits for EOSs per orbit			

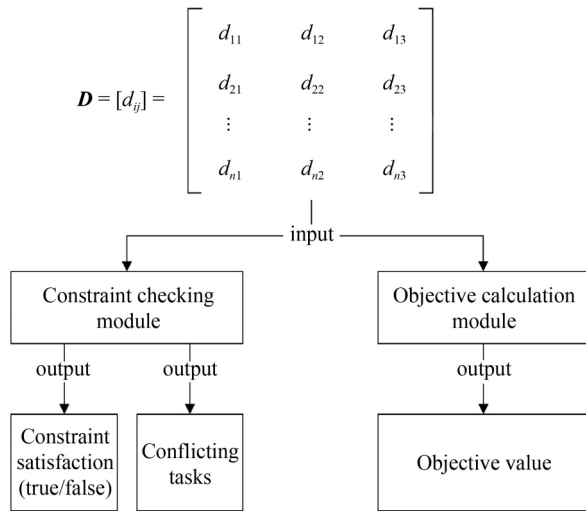


Fig. 8 Procedure of the model that judges the satellite scheduling solution as feasible or not and evaluates its objective value.

4 Algorithm library

With the decoupled modeling and optimization framework, the algorithm library is built to collect well-known and outperforming algorithms that can address benchmark and real-world cases. In this section, the general operators that reproduce neighboring solutions are provided. Then, 10 types of commonly used algorithms are collected. In addition, a PC-MA that hybridizes local search and evolutionary algorithm (EA) metaheuristics is designed.

4.1 Neighborhoods

The algorithmic design has various types of neighborhood structures and operators, but they often lack the generality in common use for EOS scheduling problems. Hence, the type of operator required should be determined. On the one hand, it must match the decision matrix as shown in the general modeling manner. On the other hand, it should function as a component to construct any complex operator further required for the algorithmic development. In this regard, two elementary operators are provided for local search and EAs, as shown in Figs. 9(a) and 9(b), respectively.

- **Move operator** produces a $[0, 1]$ random number and replaces a (random) number in the decision matrix. The produced number can always be denormalized and rounded to an EO selection index for the imaging, down-link, and memory-erasing operations per task, as shown in Eq. (2). Notably, if two near numbers in a move operator are denormalized to the same index, then the operator is re-performed. In addition, it can function as the single-point mutation operator in an EA. Furthermore, any solution can be reproduced from another with a set of move operators; hence, this operator is qualified for complex algorithmic development.

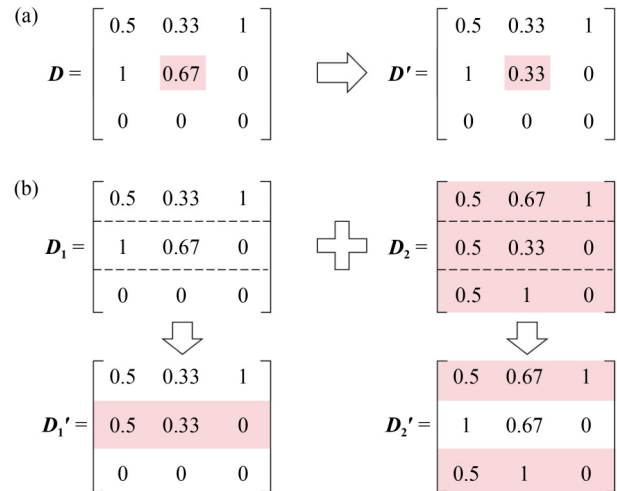


Fig. 9 Operators to reproduce neighboring solutions. (a) Move operator. (b) Crossover operator.

- **Crossover operator** swaps a (random) section of matched row vectors in two decision matrices and reproduces two new matrices. This operator is special for EAs, which exchanges the EO selection of some tasks. It can also be viewed as a set of designated move operators. Four independent solutions are obtained after this operator, including two parents and two offspring, and the selection(s) are determined by the metaheuristics or rules in algorithms.

Here, the move and crossover operators are likely to generate infeasible solutions, especially in the satellite scheduling problems with strong constraints. When a move operation on the current solution generates a lower-objective or infeasible solution, the solution is abandoned and a new move operation is performed on the last solution. When a crossover operator generates infeasible solutions, a simple deconflicting operator is performed on the solutions, canceling conflicting tasks in the priority-ascending order. Many lower-objective solutions are obtained by the deconflicting operation and the move operation, and the roulette operation is used for solution selection.

4.2 Common algorithms

Despite that many outperforming algorithms have been developed for EOS scheduling problems, few of them show good generality for real-world applications because their specific operators, neighborhoods, or solution presentations are affected by over-simplifications. In fact, engineering practice shows that some simple local search and EA metaheuristics, such as Tabu search, simulated annealing (SA), and genetic algorithm (GA), can solve the real-world problems effectively. Therefore, with the objectives of easy use and real-world application of the proposed general modeling and optimization technique, three catalogs including 10 common algorithms based on

general operators are in this subsection.

- **Rules:** First-in-first-service that selects the first EOs of imaging and downlink operations per task that satisfy the constraints. In addition, the EOS memory is not erased until the related constraints are violated. It is a common and easy-to-implement algorithm used in real-world EOS scheduling systems, and it is collected into the library initially and examined in experiments.

- **Local search algorithms:** 1) Hill climbing, which accepts non-decreasing solutions once found. 2) Tabu search, which runs hill climbing in inner loops and records local optima with an FIFO Tabu list for solution filter and escape. 3) SA, which runs hill climbing and accepts decreasing solutions based on an annealing probability. 4) Late acceptance (LA), which runs hill climbing and accepts the solutions that outperform certain history solutions. 5) Tabu SA (TSA), which adds a Tabu list to the SA for solution filter. 6) Tabu LA (TLA), which adds a Tabu list to the LA for solution filter.

- **EAs:** 1) GA, which iteratively performs crossover and mutation operators among the solution population and reserves better offspring for the next generation. 2) Tabu GA (TGA), which adds a Tabu list to the GA for offspring filter per generation. 3) Annealing GA (AGA), which adds an annealing probability to the GA for offspring filter per generation.

These algorithms are well-known; thus, their pseudocode is not explained here. Any solution that violates the constraints cannot be accepted in these algorithms, and no other special or problem-dependent operator is designed. Among these algorithms, the first-in-first-service rule, TLA, and AGA play as competitors in the case studies in Section 5.

4.3 Parallel competitive memetic algorithm

Although some well-known and easy-to-use algorithms have been provided, common local search and EAs may underperform in terms of exploration and exploitation, respectively. A high-performance algorithm that can address real-world EOS scheduling problems is also required to further strengthen the optimization ability of the proposed general modeling and optimization technique.

The algorithm hybridization often works effectively, and the advanced computer hardware motivates algorithmic developments based on parallelism and competition. Thus, a PC-MA is adopted in this subsection. As shown in Fig. 10, parallelism, competition, and evolution are the keywords in this algorithm, and the flow chart that matches these keywords is explained as follows.

- **Input and initialization:** Given the required parameters, the main computer thread is enabled, and an initial population presented by decision matrices is generated. The best solutions in the population are inputted into the follow-up local search algorithms.

- **Inner phase 1: Parallelism:** In a parallel manner, each thread probably (randomly at first) runs a local search algorithm based on its probability, and the recently obtained solutions are recorded. After the local search optimization, those threads are suspended, and they amalgamate their recently obtained solutions into a population set.

- **Inner phase 2: Competition:** Local search algorithms are graded according to the number of solutions in the population they obtained. When more solutions are obtained by the algorithms, their grades are high, and the probability that they will be used in the next-round parallelism is great.

- **Inner phase 3: Evolution:** Based on the obtained population, the evolution is performed to explore better solutions. Subsequently, the reproduced best solutions are selected and re-inputted into the local search algorithms in the next-round parallelism.

- **Output:** When the total computing time is satisfied, the best-found solution is output.

The advantages of the algorithm can also be stated in terms of its three keywords.

The parallelism leads to optimization acceleration and diversity. On the one hand, parallel algorithms can certainly perform much more searching within the same running time. On the other hand, those algorithms obtain diverse solutions in different searching trajectories. The diversity is also required by the follow-up evolution strategy. In addition, parallelism results in an easy-to-expand algorithm, and a new competitor that adopts other meta-heuristics, operators, or parameters can easily join by enabling a new thread. In addition to other software-based acceleration techniques, parallelism is a hardware-based approach that can effectively use the algorithm. The algorithm can offer good potentials because computing resources become easily accessible in the current society and in all types of large scheduling systems.

The competition leads to adaptivity because the outperforming algorithms and operators are selected and performed more frequently over time, whereas the underperforming ones are eliminated.

The evolution strengthens the exploration ability. Local search (Tabu search and SA) and EAs (GA and particle swarm optimization) often outperform in exploitation and exploration, respectively. Thus, the evolution performed along with local search optimization assists in escaping from local optima and explores a large solution space. Different from the commonly used memetic algorithms (MAs) that originate from Moscato (1989), which often adopt EAs to generate solutions in the main loop and perform local search to improve them, the proposed MA adopts local search to generate solutions in the main loop and performs EAs to improve them. In other words, it combines the EA and local search but still maintains the local search optimization ability in the strictly constrained solution space and the local search

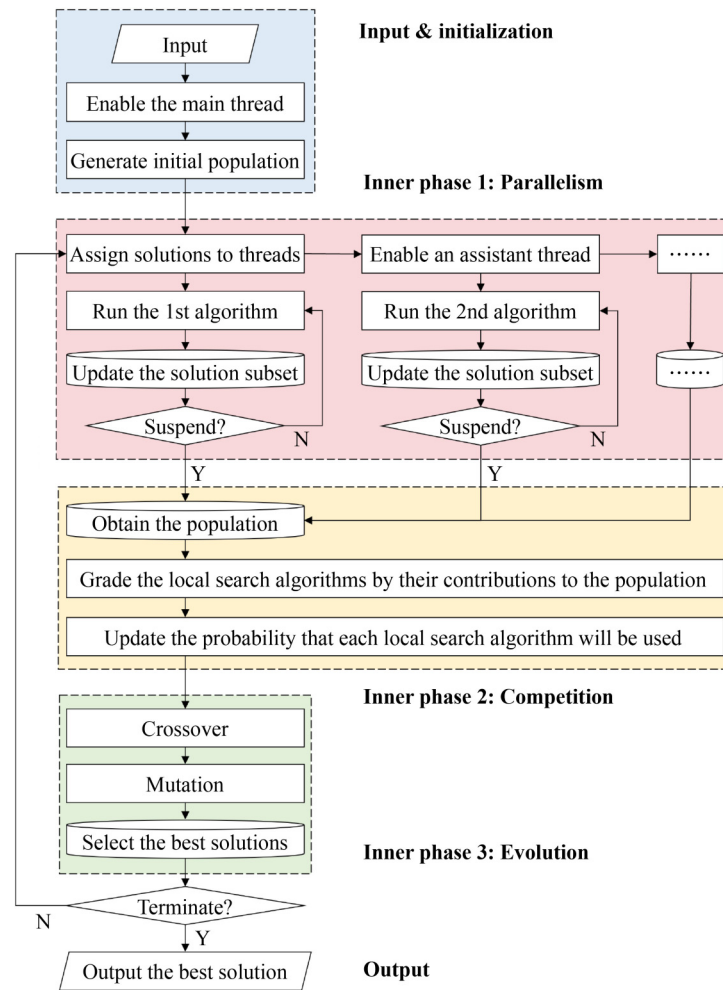


Fig. 10 Flow chart of the PC-MA.

computational complexity. Therefore, it can not only extend the advantages of traditional MAs, but also show better performance in addressing the strictly constrained and time-limited EOS scheduling problems by more general local search loops.

The algorithm library, which includes 10 types of common algorithms and the PC-MA, is built in the proposed general modeling and optimization technique. In the next section, the technique is examined on different EOS scheduling cases, and the algorithm performance is shown.

5 Case studies

Four different cases, including a new benchmark, are introduced in this section to examine the applicability and effectiveness of the general modeling and optimization technique for EOS scheduling problems in this study. In comparison with state-of-the-art algorithms, the experimental results of these cases are then presented and discussed.

5.1 Experiment setup

Four experiment cases that differ in EOSs, constraints, characteristics, and problem size are provided as follows.

- **Case 1:** A simplified EOS scheduling problem that only considers EOS imaging operations with time-dependent transition time (Liu et al., 2017). In this case, the simplification indicates that only imaging operations are required to be scheduled and only one EOS is involved. This case is highlighted in two aspects: 1) the transition time between two EOS operations is time-dependent, and 2) the VTW is much longer than the time required per task. Thus, additional begin time determination is required. As a result, exact algorithms, such as linear programming, were proven ineffective. Thus, Liu et al. (2017) designed an adaptive large neighborhood search (ALNS) algorithm. Herein, only the subcases about regional (rather than worldwide) tasks are used because the worldwide tasks were proven to be easily scheduled in this case.

- **Case 2:** An extension of Case 1 with an additional time-dependent objective function. To address this

problem, Peng et al. (2019) designed a bidirectional dynamic programming-based iterated local search (BDP-ILS) algorithm, where the iterated local search (ILS) and DP optimize the VTW and the begin time per task, respectively. Although Peng et al. (2019) discussed the importance of this time-dependent objective due to imaging qualities, some real-world EOSs, such as SuperView-1, would rather view the imaging qualities as a type of visibility constraint. Therefore, the time-dependent objective is used only here and not considered in the following real-world experiments.

• **Case 3:** A well-known satellite range scheduling benchmark (Air Force Office of Scientific Research, 2003) provided by US Air Force Satellite Control Network (AFSCN). It can also be viewed as an EOS scheduling problem that only considers downlink operations. In this problem, on the one hand, the VTW of a low-orbit satellite equals the time required per task; on the other hand, that of a high-orbit satellite is much longer, and thus, additional begin time determination is required. To address this problem, Luo et al. (2017) designed a quick conflict-resolution heuristic and obtained the best-known results in this benchmark.

• **Case 4:** A real-world EOS scheduling problem that considers imaging, downlink, and memory-erasing operations with differentiated constraints. Different from the simplified data, this case was provided by the real-world management agencies with the following real-world characteristics: 1) real data of a famous four-EOS Chinese commercial constellation called SuperView-1, 2) downlink and memory-erasing requirements that were often oversimplified in previous studies, 3) complicated constraints that differ in EOSs and tasks, and 4) subcases for single-EOS, multi-EOS, single-day, and multi-day. The dataset was provided by the SuperView-1 management agency for public studies. It can be accessed at github.com/duyonghao15/Benchmark-for-SuperView-1.git, where more constraint details are also available.

As shown in Table 2, Cases 1 and 2 only consider satellite imaging, and Case 3 only considers the downlink. In other words, Cases 1–3 simplify the real-world satellite scheduling problem (similar to Case 4) because they ignore the imaging or downlink operation. Thus, when applying the decision model proposed in Section 3.1 to Cases 1–3, the decision variable that determines the ignored imaging, downlink, or memory-erasing operation should be disused or set constantly 0.

With these cases, the proposed general modeling and optimization technique is examined in comparison with the state-of-the-art algorithms in the next subsection. A four-thread PC-MA that runs Tabu search, SA, LA, and TLA was used. Moreover, three competitive algorithms in the algorithm library were selected for experiments, including the first-in-first-service (rule), AGA, and TLA. The TLA and AGA were selected to show intensive comparison on local search and evolutionary abilities, respectively, due to the hybridization characteristic of local search and EAs in MA.

The experiments were conducted by Java 1.8.0, using an Intel (R) Core (TM) i7-7600U CPU at 2.80 GHz on Windows 10 with 8 GB RAM and four threads. The Tabu and late length in those algorithms were set to 10% and 100% of the task number per subcase, respectively. The selection and mutation probabilities in the evolution were set to 70% and 30%, respectively. The evolution in the PC-MA was performed every 12 seconds. All experiments were run 10 times except for those using the rule; the average results are listed in the following tables. The computing time for Cases 1 and 2 was set identical to that in the literature. The time for Case 3 was set to 1 minute, and that for Case 4 was set to 1 minute per EOS per day. The performance standard to compare different algorithms in each subcase is their highest average value obtained (marked in bold in the table). However, in each case, the performance standard to compare algorithms is the number of the highest solutions obtained among subcases.

5.2 Experimental results

The experimental results of Case 1 are presented in Table 3, where the best results among the algorithms are marked in bold. The table shows that the PC-MA outperforms the state-of-the-art ALNS (Liu et al., 2017) in all subcases, especially when the task number is greater than 200. In addition, the TLA shows good performance, but the AGA does not work effectively. The underperformance of the AGA should be attributed to the common crossover operators that often violate the constraints in EOS scheduling problems within limited computing time.

Hereby, the general modeling and optimization technique developed in this study is applicable and effective in addressing this problem. ALNS attempted to consider the begin time within the VTW to be as early as possible by iterative constraint checking to address the issue of a

Table 2 Four EOS scheduling cases to be studied

Case	EOS operations	State-of-the-art	Task number
Case 1	Imaging	ALNS (Liu et al., 2017)	50–400
Case 2	Imaging	BDP-ILS (Peng et al., 2019)	100–600
Case 3	Downlink	Conflict-resolution heuristic (Luo et al., 2017)	297–322
Case 4	Imaging, downlink, and memory-erasing	Seldom studied	56–1267

Table 3 Experimental results of Case 1

Subcase	Rule	AGA	TLA	PC-MA	ALNS
50	270.0	270.0	270.0	270.0	269.0
100	443.0	506.2	556.5	568.8	543.0
150	575.0	661.5	767.0	781.9	716.0
200	584.0	711.2	813.6	853.0	705.0
250	753.0	843.3	951.1	1017.9	813.0
300	809.0	855.8	986.3	1061.3	749.0
350	752.0	858.1	967.4	1066.3	841.0
400	793.0	875.7	1001.4	1133.7	739.0
Outperform	1/8	1/8	1/8	8/8	0/8

longer VTW than the time required per task in this problem. However, this process could be time-consuming and inefficient given the same computing time. In addition, the commonly used rule underperforms, even though it is the most easy-to-implement and understandable algorithm with regard to human experience. Therefore, the technique that adopts metaheuristic algorithms is necessary to displace the traditional rules in real-world EOS scheduling systems.

The experimental results of Case 2 are presented in Table 4. In the table, the PC-MA outperforms the state-of-the-art BDP-ILS (Peng et al., 2019) when the task number is 100 or 400 and slightly underperforms when the number is 200, 300, or 500. The BDP-ILS employs DP to determine the begin time within the VTW per task. Thus, it can output the optimal solution given VTWs, but it also requires considerable time. In the same computing time, the MA can perform many direct searches in a parallel manner and can show competitive performance. However, when the number reaches 600, which is relatively large for EOS scheduling problems, the MA does not show competitive performance. In spite of this condition, the easy-to-use PC-MA is qualified for addressing Case 2 with a common task number. Other competitors, such as TLA and AGA, can also solve the problem in certain degrees, and they serve as easy-to-use tools available in the algorithm library for the general modeling and optimization technique in this study.

Table 4 Experimental results of Case 2

Subcase	Rule	AGA	TLA	PC-MA	BDP-ILS
100	291.0	420.7	489.5	494.0	491.1
200	396.0	621.2	725.7	741.4	747.6
300	409.0	671.1	811.1	833.8	854.9
400	475.0	904.1	1009.1	1060.7	1053.2
500	497.0	1012.9	1081.8	1149.9	1153.7
600	589.0	921.4	1128.2	1195.6	1260.1
Outperform	0/6	0/6	0/6	2/6	4/6

The experiment results of Case 3 are shown in Table 5. The table shows that the PC-MA, TLA, and the state-of-the-art conflict-resolution heuristic (Luo et al., 2017) show similar performance in addressing this EOS range scheduling problem. The results in the literature are the best-known in this benchmark, and the PC-MA approaches the best in Days 1, 2, 3, and 4 and reaches the best in Days 5, 6, and 7. Hence, the proposed technique presents competitive performance in addressing this benchmark. Despite that the state-of-the-art conflict-resolution heuristic can output the results in a few seconds, it cannot show good adaptability to other EOS scheduling problems due to its simplification, complex implementation, and special operators. Therefore, the proposed technique also outperforms in terms of applicability, which is required in real-world EOS scheduling problems.

Table 5 Experimental results of Case 3

Subcase	Rule	AGA	TLA	PC-MA	Conflict-resolution
Day 1	269.0	309.5	313.2	314.9	316.0
Day 2	262.0	292.6	296.1	298.6	299.0
Day 3	264.0	300.2	306.1	308.0	309.0
Day 4	265.0	306.6	312.9	315.2	316.0
Day 5	255.0	295.8	300.0	303.0	303.0
Day 6	239.0	287.7	291.0	294.0	294.0
Day 7	257.0	290.0	293.0	293.0	293.0
Outperform	0/7	0/7	1/7	3/7	7/7

The experiment results of Case 4 include two versions: 1) results only constrained by VTWs and transition time given that previous studies often over-simplified other constraints except the two, and 2) results with all real-world constraints, as shown in Tables 6 and 7, respectively. In both versions of Case 4, the PC-MA outperforms other competitors in most subcases. Other competitors, such as TLA and AGA, can also address this problem. The first-in-first-service rule is one of the in-use algorithms in the SuperView-1 scheduling system. Hence, the application of the proposed general modeling and optimization technique results in a considerable improvement of over 50%. In addition, by comparing Tables 6 and 7, a great objective difference can be observed, indicating that the most considered constraints and real-world ones lead to relatively different results. Therefore, the importance of real-world modeling and constraints should receive more attention in future studies; otherwise, the well-designed and outperforming algorithms may lose value in real-world applications.

Furthermore, to show how the parallelism and competition work in the PC-MA, the mean grades (percent of the solutions obtained in the population) of the used parallel local search algorithms per 20% optimization process are shown in Fig. 11 in terms of the largest-scale subcases in

Table 6 Experimental results of Case 4 (only with VTWs and transition time constraints)

EOS	Date	Rule	AGA	TLA	PC-MA
1	Apr. 12	110.0	112.3	113.0	113.0
1	Apr. 12, 13	185.0	189.2	189.8	190.0
2	Apr. 12	143.0	146.3	148.0	148.0
3	Apr. 12	55.0	55.0	55.0	55.0
4	Apr. 12	129.0	134.4	136.8	137.0
1, 2, 3, 4	Apr. 13	331.0	337.4	344.6	344.9
1, 2, 3, 4	Apr. 14	262.0	271.3	272.0	272.0
1, 2, 3, 4	Apr. 13, 14	593.0	603.0	614.2	616.7
1, 2, 3, 4	Apr. 16	515.0	533.8	560.4	571.6
1, 2, 3, 4	Apr. 17	598.0	634.5	667.3	681.0
1, 2, 3, 4	Apr. 16, 17	1113.0	1159.7	1231.5	1246.9
Outperform		1/11	1/11	4/11	11/11

Table 7 Experimental results of Case 4 (with all constraints)

EOS	Date	Rule	AGA	TLA	PC-MA
1	Apr. 12	35.0	69.0	77.9	79.4
1	Apr. 12, 13	67.0	95.6	99.8	108.4
2	Apr. 12	29.0	72.7	84.0	86.6
3	Apr. 12	30.0	47.8	51.3	53.3
4	Apr. 12	29.0	73.6	81.1	85.5
1, 2, 3, 4	Apr. 13	113.0	219.2	253.9	260.4
1, 2, 3, 4	Apr. 14	96.0	193.4	232.1	241.0
1, 2, 3, 4	Apr. 13, 14	182.0	405.0	446.2	491.8
1, 2, 3, 4	Apr. 16	234.0	251.7	283.6	296.2
1, 2, 3, 4	Apr. 17	239.0	275.9	308.5	331.0
1, 2, 3, 4	Apr. 16, 17	357.0	455.6	534.0	544.3
Outperform		0/11	0/11	0/11	11/11

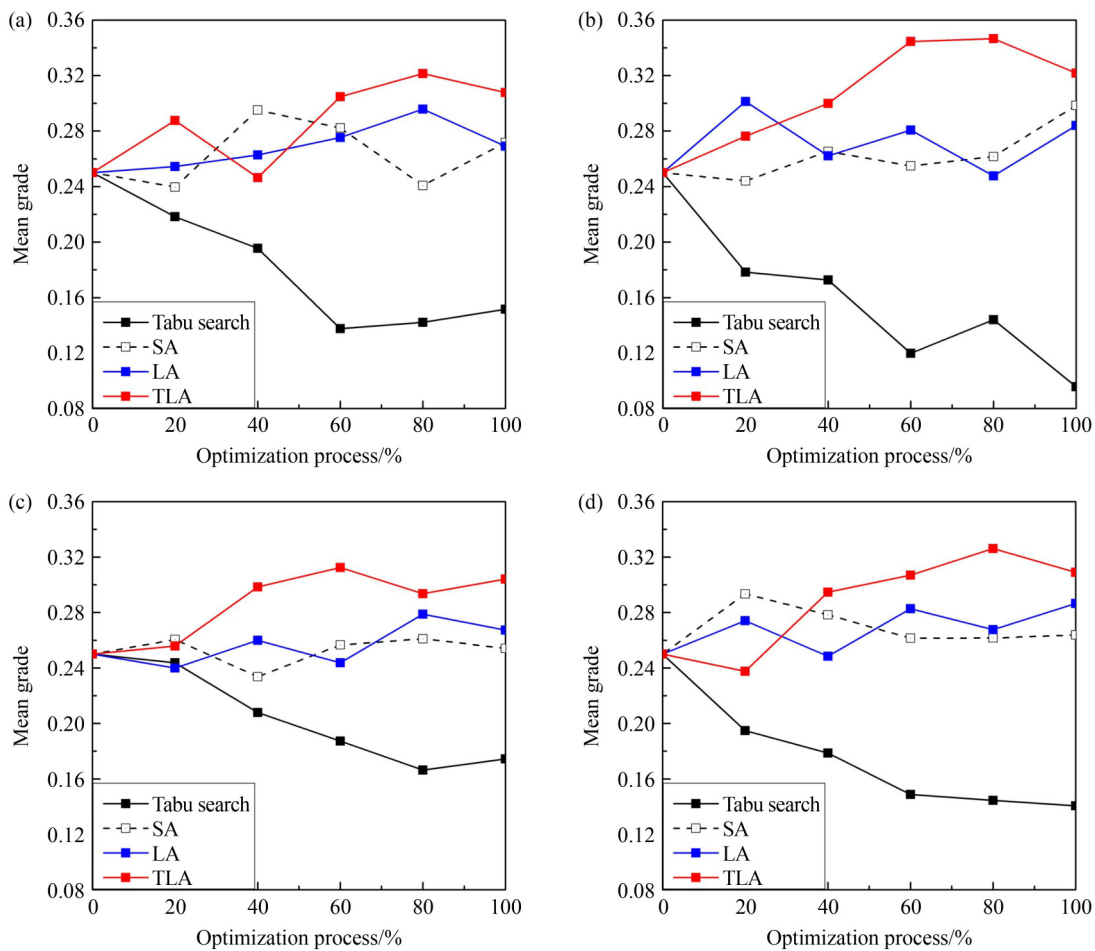


Fig. 11 Mean parallel-algorithm grades per 20% optimization process in the PC-MA. (a) Subcase 400 in Case 1. (b) Subcase 600 in Case 2. (c) Subcase Day 1 in Case 3. (d) Subcase Apr. 16, 17 with all the constraints in Case 4.

Cases 1, 2, 3, and 4. As the figure shows, four parallel algorithms were initially graded by 0.25 but differ remarkably with the optimization given that the probabilities of those algorithms were modified according to their

grades over time. Herein, the TLA (marked in red) often performed well, whereas the Tabu search (marked in black) often underperformed but still made some contributions. In other words, although some algorithms cannot

show full-time good performance, they can still assist in exploring solutions and increasing solution diversity in the PC-MA. Therefore, the PC-MA can show competitive performance in comparison with the state-of-the-art algorithms in different cases.

6 Discussion

After the experiments, the type of EOS scheduling technique required is discussed in this section. Previous studies often offered a “simplified model with complicated algorithms”, namely, the problem was over-simplified and solved by a specifically designed algorithm. However, their real-world application effects are unsatisfactory due to the great difference between the simplified and real-world problems. Therefore, EOS management agencies would rather accept a “real-world model with simple algorithms”. As a result, real-world modeling and constraints are stressed in this study.

Another highlight of the proposed technique is the decoupled framework for general real-world applications. The decoupling leads to 1) system compatibility for different EOSs, 2) integrability for multiple EOSs, and 3) extensibility for further algorithm and model update. However, the proposed technique cannot be used to effectively address all types of EOS scheduling problems, but the operation-based modeling idea for EOS scheduling can be a general-purpose approach. Hereby, in addition to the EOS scheduling studied, the proposed general modeling and optimization technique can be applied to other satellite scheduling problems, as shown in Table 8. In fact, the real-world scheduling problems shown in Table 8 have been addressed based on the proposed technique.

Table 8 Ideas of modeling other satellite scheduling problems using the proposed technique

Other satellite scheduling problem	Satellite operation contained per task
Satellite range scheduling	Downlink
Relay satellite scheduling	Downlink and memory-erasing
Time synchronization scheduling of navigation satellites	Downlink
Inter-link scheduling of navigation satellites	Inter-link (divided in timeslots)

Some necessary characteristics should be involved in the benchmark dataset for EOS scheduling studies: 1) the reality, which ensures that the modeling and optimization techniques for benchmarks are also applicable for real-world use; 2) the representativeness, which reflects the most troubling issues to EOS management agencies; and 3) the oversubscription, which raises the scheduling necessity and difficulty in this problem. The new benchmark exemplified by the Chinese commercial EOSs

effectively addresses these characteristics. Thus, it can provide a good reference to future studies of EOS scheduling problems.

7 Conclusions

General modeling and optimization techniques are urgently required by EOS management agencies to perform flexible, differentiated, and efficient management of EOSs and tracking stations. However, previous techniques cannot show excellent generality to be engineered in real-world applications. In the proposed general technique, a framework that decouples the modeling, constraints, and optimization of EOS scheduling problems is built. Then, the EOS scheduling problems are appropriately modeled in a general modeling manner. Moreover, more than 10 types of optimization algorithms, including a PC-MA, are developed. In cases of the simplified and real-world EOS scheduling problems, the developed technique shows good performance in applicability and effectiveness.

The contributions of this study include 1) the general technique for modeling and optimizing real-world EOS scheduling problems and 2) the new benchmark exemplified by real-world EOSs. With this technique, different EOSs, such as optical, electronic, and radar EOSs, no longer require a mess of independent scheduling systems. Future work based on this technique includes a system development for more than 60 EOSs. In addition to the proposed multi-thread PC-MA, more outperforming algorithms with satisfactory time and resource consumption will be studied and transplanted in the technique for real-world use.

Competing Interests The authors declare that they have no competing interests.

References

- Air Force Office of Scientific Research (2003). Exploiting elementary landscapes for search (AFSCN scheduling problems)
- Chu X G, Chen Y N, Tan Y J (2017). An anytime branch and bound algorithm for agile earth observation satellite onboard scheduling. *Advances in Space Research*, 60(9): 2077–2090
- Cordeau J F, Laporte G (2005). Maximizing the value of an earth observation satellite orbit. *Journal of the Operational Research Society*, 56(8): 962–968
- Du Y H, Wang L, Xing L N, Yan J G, Cai M S (2021). Data-driven heuristic assisted memetic algorithm for efficient inter-satellite link scheduling in the BeiDou navigation satellite system. *IEEE/CAA Journal of Automatica Sinica*, 8(11): 1800–1816
- Du Y H, Wang T, Xin B, Wang L, Chen Y G, Xing L N (2020). A data-driven parallel scheduling approach for multiple agile earth observation satellites. *IEEE Transactions on Evolutionary Computation*,

- 24(4): 679–693
- Du Y H, Xing L N, Chen Y G (2022). Satellite scheduling engine: The intelligent solver for future multi-satellite management. *Frontiers of Engineering Management*, 9(4): 683–688
- 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
- He Y M, Xing L N, Chen Y W, Pedrycz W, Wang L, Wu G (2022). A generic Markov decision process model and reinforcement learning method for scheduling agile earth observation satellites. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(3): 1463–1474
- Jang J, Choi J, Bae H J, Choi I C (2013). Image collection planning for Korea Multi-Purpose SATellite-2. *European Journal of Operational Research*, 230(1): 190–199
- Lemaître M, Verfaillie G, Jouhaud F (2000). How to manage the new generation of agile earth observation satellites. In: *Proceedings of the 6th International SpaceOps Symposium*. Toulouse: AIAA, 1–8
- 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
- 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
- Mok S H, Jo S, Bang H, Leeghim H (2019). Heuristic-based mission planning for an agile earth observation satellite. *International Journal of Aeronautical and Space Sciences*, 20(3): 781–791
- Moscato P (1989). *On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms*. Caltech Concurrent Computation Program 158-79. Pasadena, CA: California Institute of Technology
- Nag S, Li A S, Merrick J H (2018). Scheduling algorithms for rapid imaging using agile Cubesat constellations. *Advances in Space Research*, 61(3): 891–913
- 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
- Peng G S, Song G P, Xing L N, Gunawan A, Vansteenwegen P (2020). An exact algorithm for agile earth observation satellite scheduling with time-dependent profits. *Computers & Operations Research*, 120: 104946
- Valicka C G, Garcia D, Staid A, Watson J P, Hackebeil G, Rathinam S, Ntaimo L (2019). Mixed-integer programming models for optimal constellation scheduling given cloud cover uncertainty. *European Journal of Operational Research*, 275(2): 431–445
- Wang J J, Demeulemeester E, Hu X J, Wu G H (2020). Expectation and SAA models and algorithms for scheduling of multiple earth observation satellites under the impact of clouds. *IEEE Systems Journal*, 14(4): 5451–5462
- Wu G, Liu J, Ma M, Qiu D S (2013). A two-phase scheduling method with the consideration of task clustering for earth observing satellites. *Computers & Operations Research*, 40(7): 1884–1894
- 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
- Yang C F (2021). Innovation and development of BeiDou Navigation Satellite System (BDS) project management mode. *Frontiers of Engineering Management*, 8(2): 312–320
- Zhu W M, Hu X X, Xia W, Jin P (2019). A two-phase genetic annealing method for integrated earth observation satellite scheduling problems. *Soft Computing*, 23(1): 181–196