symmetry

MDPI

Article

Knowledge-Guided Parallel Hybrid Local Search Algorithm for Solving Time-Dependent Agile Satellite Scheduling Problems

Yuyuan Shan [1], Xueping Wang [1], Shi Cheng [1,*], Mingming Zhang [2] and Lining Xing [3]

[1] School of Computer Science, Shaanxi Normal University, Xi'an 710119, China; shanyuyuan@snnu.edu.cn (Y.S.); wxp@snnu.edu.cn (X.W.)
[2] Department of Mechanical and Mechatronics Engineering, The University of Auckland, Auckland 1030, New Zealand; mzha509@aucklanduni.ac.nz
[3] School of Electronic Engineering, Xidian University, Xi'an 710119, China; lnxing@xidian.edu.cn
* Correspondence: cheng@snnu.edu.cn

Abstract: As satellite capabilities have evolved and new observation requirements have emerged, satellites have become essential tools in disaster relief, emergency monitoring, and other fields. However, the efficiency of satellite scheduling still needs to be enhanced. Learning and optimization are symmetrical processes of solving problems. Learning problem knowledge could provide efficient optimization strategies for solving problems. A knowledge-guided parallel hybrid local search algorithm (KG-PHLS) is proposed in this paper to solve time-dependent agile Earth observation satellite (AEOS) scheduling problems more efficiently. Firstly, the algorithm uses heuristic algorithms to generate initial solutions. Secondly, a knowledge-based parallel hybrid local search algorithm is employed to solve the problem in parallel. Meanwhile, data mining techniques are used to extract knowledge to guide the construction of new solutions. Finally, the proposed algorithm has demonstrated superior efficiency and computation time through simulations across multiple scenarios. Notably, compared to benchmark algorithms, the algorithm improves overall efficiency by approximately 7.4% and 8.9% in large-scale data scenarios while requiring only about 60.66% and 31.89% of the computation time of classic algorithms. Moreover, the proposed algorithm exhibits scalability to larger problem sizes.

Keywords: agile Earth observation satellite; knowledge-guide search; hybrid local search algorithm

check for updates

Citation: Shan, Y.; Wang, X.; Cheng, S.; Zhang, M.; Xing, L. Knowledge-Guided Parallel Hybrid Local Search Algorithm for Solving Time-Dependent Agile Satellite Scheduling Problems. *Symmetry* **2024**, *16*, 813. https://doi.org/10.3390/sym16070813

Academic Editor: Theodore E. Simos

Received: 28 May 2024
Revised: 21 June 2024
Accepted: 25 June 2024
Published: 28 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Solving scheduling problems by intelligent optimization algorithms has been utilized in many industrial and engineering applications [1]. Earth observation satellites (EOSs) are primary platforms for acquiring spatial image information. They are widely used in disaster relief, emergency monitoring, and other fields due to their broad observation coverage and absence of national boundaries [2]. Conventional non-agile Earth observation satellites have only one roll angle, which aligns the observation time window with the visible time window (VTW). In recent years, there has been a significant increase in demand for observation, leading to the deployment of a large number of agile Earth observation satellites.

As shown in Figure 1, compared with ordinary non-agile ground observation satellites, agile dispatching satellites have three rolling angles [3], namely, roll angle, pitch angle, and yaw angle. This capability allows the satellite to observe the ground target before it is directly overhead by adjusting the pitch angle. Consequently, the observation time window for the mission is narrower than the visible time window. Yet, the duration of observation is extended and is not limited to the period when the satellite is overhead but instead spans throughout the observable period. The use of agile Earth observation satellites greatly enhances observation capabilities. As the observation window expands, scheduling becomes more challenging. Therefore, selecting a reasonable start time for tasks is necessary while determining the visible time window.

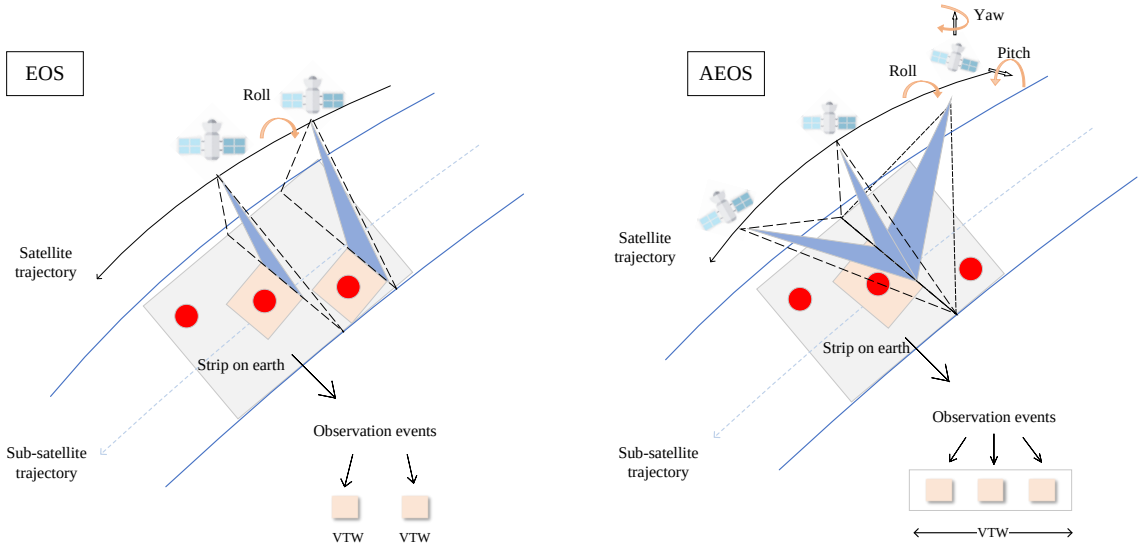


Figure 1. Comparison of non-agile and agile satellites.

Additionally, a certain transition time is required between two consecutive observation tasks of a satellite, which depends on the satellite's attitude angles during observation. Non-agile satellites have only a single side-sway angle, determined by the target's geographic location, making the required transition time fixed. Agile satellites, however, involve roll, pitch, and yaw angles, with the pitch angle determined by the observation time point. Consequently, the transition time during the imaging process of agile satellites is no longer a fixed value but varies with time. This time-dependency characteristic of agile satellites imposes higher demands on algorithm performance.

Due to its complexity properties, the AEOS task scheduling has been proven to be an NP-hard problem [4]. The objective is to maximize observation benefits by reasonably allocating tasks, considering both constraints handling and task requirements. Current approaches to solving agile satellite scheduling mainly involve heuristic algorithms, metaheuristic algorithms, and machine learning algorithms.

Heuristic algorithms demonstrate exemplary performance in small-scale instances [5,6]. Still, as the problem scales up, the algorithm design becomes more challenging, often leading to locally optimal solutions and failing to produce high-quality scheduling solutions [2,7]. Lemaître et al. initially provided a comprehensive overview of the AEOS problem and proposed various heuristic algorithms to address it [8]. Cordeau and Laporte further employed tabu search algorithms to solve the AEOS problem, albeit only considering single-orbit optimization issues [9].

Metaheuristic algorithms enhance the ability to escape local optima by guiding heuristic algorithms with meta-knowledge during the solving process. Baioletti et al. proposed an algebraic method based on repeated permutations, designing a discrete operator for the differential evolution algorithm, and successfully applied it to the job shop scheduling problem [10]. Tasgetiren et al. introduced an iterative greedy algorithm for solving the blocking flow shop scheduling problem. They generated an initial solution using a heuristic method and designed two local search algorithms, utilizing an iterative jumping probability to determine which neighborhood structure to adopt for guiding the search [11]. Zhang et al. designed a particle swarm optimization method for the resource-constrained project scheduling problem, aiming to minimize project duration. Using a parallel algorithm, they represented priorities through particles and transformed them into feasible schedules based on priority and resource constraints. Experimental results demonstrated that this method outperforms traditional algorithms [12]. Wu et al. proposed several metaheuristic-based algorithms and introduced task clustering into the Earth Observation Satellite scheduling problem [13]. Hu et al. combined the EOS problem with data transmission issues and designed a dual-population artificial bee colony algorithm to solve it [14].

Wolfe and Sorensen proposed priority scheduling methods and genetic algorithms for multi-orbit agile satellite scheduling [15]. However, most studies have ignored transition times or represented transition times as constants. Liu et al. initially proposed a time-dependent agile satellite scheduling model and developed an adaptive large neighborhood search algorithm for this purpose [16]. Building upon this work, He et al. augmented the ALNS algorithm with various operators and tabu strategies to address the AEOS problem [17,18]. Additionally, they extended this method to multi-satellite scenarios by employing heuristic techniques to design a task allocation layer. The time-dependent AEOS was modeled as a multi-objective optimization problem and solved by a memetic algorithm.

Machine learning algorithms, such as artificial neural networks (ANN), were utilized to solve satellite scheduling problems. Wu et al. integrated genetic algorithms with neural networks, developing a data-driven genetic algorithm. They used historical data to train an ANN neural network for generating superior initial populations [19]. The time-dependent property was mathematically proved using the minimum transition time instead of the actual transition time. An adaptive parallel local search algorithm combines multiple algorithms to run in parallel and adds an adaptive process to the algorithms and operators; finally, data mining techniques were utilized to enhance understanding quality and reduce computation time.

Despite extensive research on AEOS scheduling, there is a lack of literature addressing time-dependent agile satellite scheduling. Furthermore, given the complex task requirements and high real-time demands currently faced in agile satellite scheduling, existing solutions, although partially addressing the scheduling problem, still suffer from low computational efficiency and unstable solution outcomes. Learning and optimization are symmetrical processes of solving problems. Based on problem knowledge, efficient algorithms could be designed to solve optimization problems. Thus, to address these issues, we propose a knowledge-guided parallel hybrid local search algorithm (KG-PHLS) algorithm to improve solution efficiency and benefits under larger-scale and more constrained conditions.

The remaining manuscript is structured as follows. Section 2 primarily describes the model of the agile satellite scheduling problem and its relevant assumptions, while Section 3 provides a detailed explanation of each component of the knowledge-guided parallel hybrid local search algorithm. Section 4 presents the experimental data and environment, conducts algorithm testing, and analyzes the experimental results. The conclusion of the article is given in Section 5.

2. Problem Description

AEOS is a large-scale optimization problem in the satellite domain involving selecting a feasible set of tasks from a pool of candidates under constraints such as maximum memory, maximum power, and transition time. For each task, corresponding time windows are designated to maximize the revenue.

2.1. Decision Variables

Let $R = \{r_1, r_2, \cdots, r_n\}$ be a set of tasks, where N represents the number of tasks to be scheduled. For each task $r_i \in \mathcal{R}$, the key variables and parameter definitions are given in Table 1.

Table 1. Symbol of task properties.

Symbol	Explanation
p_i	The priority of task r_i
l_i	The execution duration of task r_i
m_i	The memory consumed by each t_i second
e_i	The electricity consumed by each t_i second
N	The number of tasks

The definitions of satellite parameters are shown in Table 2 in the field of satellite studies.

Table 2. Symbol of satellite properties.

Symbol	Explanation
E	The total satellite power;
M	The total satellite memory;
γ^t	The rolling angles at time t;
π^t	The pitching angles at time t;
μ^t	The yawing angles at time t;
$\|w_i\|$	The number of VTW for task r_i;
st_{ij}	The start time;
et_{ij}	The end time;
e^f	The fixed power consumption per satellite attitude adjustment;
e^s	The power consumption per degree of satellite rotation;

2.2. Assumptions

Owing to the necessity of considering numerous operational details and diverse user requirements, the daily management of satellites is rather intricate. To address the issue, we have made the following assumptions.

- The task must be completed in one continuous execution.
- The satellite can only execute one task at a time.
- The task is a standardized meta-task that can be executed at once.
- The satellite data memory usage and power consumption for meta-tasks are fixed.

2.3. Mathematical Formulation

Based on the assumptions above, the objective function and constraints for agile satellite scheduling are as follows. We define x_{ij} as a binary decision variable for determining whether to select the visible time window tw_{ij} of a task r_i. The value of x_{ij} is equal to 1 if and only if the task r_i is scheduled and selects the corresponding visible time window tw_{ij} and the observation window $ow_i \in tw_{ij}$ of r_i. If all VTWs of the task are not selected, the observation of the pair fails.

This problem aims to maximize the cumulative revenue derived from completing observation tasks. The objective function is defined as follows (see Equation (1)):

$$\max \sum_{i=1}^{N} \sum_{j=1}^{|w_i|} x_{ij} p_i \tag{1}$$

Based on the description and constraint analysis of agile satellite scheduling, we establish the following model to satisfy satellite task scheduling constraints:

$$\sum_{j=1}^{|w_i|} x_{ij} \leq 1 \tag{2}$$

$$st_{ij} \leq u_{ij} \leq u_{ij} + l_i \leq et_{ij}, \text{ if } x_{ij} = 1 \qquad \forall t_i \in T \tag{3}$$

$$\Delta g = |\gamma^{u_{ij}} - \gamma^{u_{i^*j^*}}| + |\pi^{u_{ij}} - \pi^{u_{i^*j^*}}| + |\varphi^{u_{ij}}| \qquad \forall t_i, t_{i^*} \in T \tag{4}$$

$$\Delta(t_i, t_{i^*}) = \begin{cases} 11.66, & \text{if } \Delta g \leq 10 \\ a_1 + \frac{\Delta g}{v_1}, & \text{if } 10 < \Delta g \leq 30 \\ a_2 + \frac{\Delta g}{v_2}, & \text{if } 30 < \Delta g \leq 60 \\ a_3 + \frac{\Delta g}{v_3}, & \text{if } 60 < \Delta g \leq 90 \\ a_4 + \frac{\Delta g}{v_4}, & \text{if } \Delta g > 90. \end{cases} \tag{5}$$

$$u_{ij} + l_i + \Delta(t_i, t_{i^*}) \leq u_{i^*j^*}, \text{ if } \rho_{w_{ij} w_{i^*j^*}} = 1 \qquad \forall t_i, t_{i^*} \in T, i \neq i^*, w_{ij} \in w_i, w_{i^*j^*} \in w_i^* \tag{6}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{|w_i|} x_{ij} l_i m_i \leq M \tag{7}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{|w_i|} (x_{ij} l_i e_i + \rho w_{ij} w_{i^*j^*} e^f + \rho w_{ij} w_{i^*j^*} \Delta g e^s) \leq E \qquad \forall t_{i^*} \in T, w_{i^*j^*} \in w_{i^*} \tag{8}$$

The meanings of constraints are as follows.

- Constraint (2) states that each task can only be executed once;
- Constraint (3) specifies that the executed task must fall within the visible time window;
- Constraints (4) and (5) specify the method of calculating transition time. Among them, v_1, v_2, v_3, and v_4 represent four different satellite attitude conversion speeds. Here, Δg represents the total angle change between two observations, calculated as $\Delta g = \Delta\gamma + \Delta\pi + \Delta\mu$, where $\Delta\gamma$, $\Delta\pi$, and $\Delta\mu$ represent the satellite's pitch angle, yaw angle, and roll angle, respectively. For the satellites referred to in the article, $v_1 = 1.5^\circ/s$, $v_2 = 2^\circ/s$, $v_3 = 2.5^\circ/s$, $v_4 = 3^\circ/s$, $a_1 = 5$, $a_2 = 10$, $a_3 = 16$, $a_4 = 22$, respectively.
- Constraint (6) dictates that the transition time between two tasks must not exceed the specified limit.
- Constraint (7) indicates that the memory usage must not exceed the maximum memory capacity, denoted as M.
- Constraint (8) specifies that the power consumption should not surpass the maximum power limit, denoted as E.

3. Knowledge-Guided Parallel Hybrid Local Search Algorithm

We propose a knowledge-guided parallel hybrid local search (KG-PHLS) algorithm for solving the AEOS scheduling problem based on the established agile satellite scheduling model. The framework of the proposed algorithm is illustrated in Figure 2. The algorithm is built upon the framework of a hybrid local search algorithm and integrates four strategies to enhance algorithm efficiency:

1. A knowledge-guided initial solution generation method;
2. A knowledge-guided new solution construction method;
3. Parallel search strategy;
4. Data mining techniques.

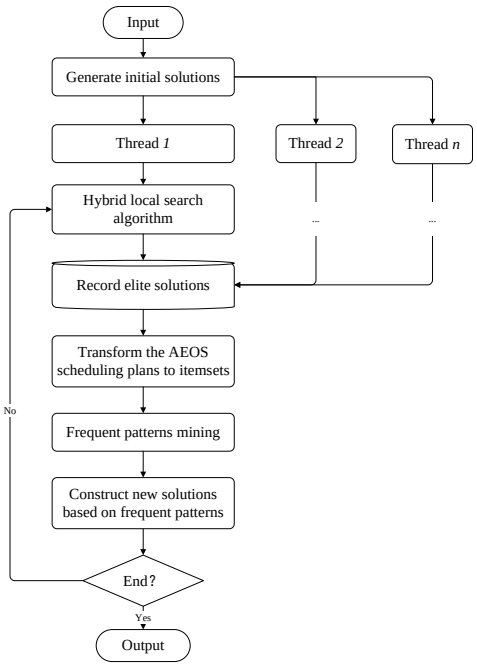


Figure 2. The flowchart of the KG-PHLS algorithm.

3.1. Knowledge-Guided Initial Solution Construction

The quality of the initial solution significantly influences the subsequent performance of the algorithm. A knowledge-based initial solution construction method is employed to speed up the search. This knowledge-guided initialization method integrates multiple heuristic rules derived from classical satellite scheduling problems to reduce ineffective searches in local search algorithms.

In constructing the initial solution guided by knowledge, four heuristic rules are used to generate the initial solution: visible time windows (VTW), task start times, task execution times, and task profits. Tasks are sorted according to the sequence of these four heuristic rules, and the scheduled tasks are inserted into the empty schedule individually. If the task does not violate the constraints, the task is started. Otherwise, the task is canceled.

3.2. Knowledge-Guided Parallel Hybrid Local Search

Meta-heuristic algorithms, such as tabu search and late acceptance, have more robust global search capabilities. They are widely used in real satellite scheduling problems due to their greater flexibility and adaptability. As shown in Algorithm 1, a hybrid local search algorithm is designed.

The algorithm mixes a late acceptance hill climbing algorithm and a tabu search algorithm and introduces a new solution construction method based on knowledge guidance. The algorithm uses the prior knowledge of the satellites to construct a new solution and decides whether to accept the current solution according to the late acceptance algorithm. If the new solution is accepted, the tabu table and the deferred table are updated, where the tabu table is used to record the tasks deleted and inserted each time a new solution is accepted. The functions in the tabu table are prohibited from being deleted or inserted in the next θ iteration, and the size of θ is set to be a random number between $[0, \sqrt{(S/2)}]$, where S is the number of tasks. The deferred table is used to record historical data.

However, within the domain of satellite resource scheduling problems, algorithm runtime stands as a crucial metric. Including multiple strategies in local search algorithms inevitably elongates the algorithm's runtime. Parallel computing is introduced to reduce search times, allowing more searches within the same runtime. During parallel computation, each thread utilizes a distinct initial solution. High-quality solutions are continuously recorded in an elite solution set throughout the iteration process. Ultimately, the elite solutions and the best solution acquired by each thread are presented, initiating a new round of parallel searches based on these newfound solutions.

Algorithm 1: Knowledge-guided parallel hybrid local search algorithm.

Input: Initial solutionR_I, elite set U_x, tabu set U_t, and deferred set U_L;
Output: The current solution R

```
R = R_I;
R_L = The last element in the overdue table;
while termination condition is not satisfied do
    Constructing new solutions based on knowledge guidance R_x;
    if the size of elite set U_x < K then
        Record the current solution R_x in U_x;
    else
        Find R_m ← {arg min{F(x) : x ∈ U_x}};
        if f(R_x) > f(R_m) or f(R_x) > f(R_L) then
            Replace R_m by R_x;
            Record the current deleted and inserted tasks to the tabu table U_t;
            Record the current solution to the overdue table U_L;
return R;
```

3.3. Knowledge-Guided New Solution Construction

Constructing new solutions is paramount in local search algorithms as it directly influences the algorithm's performance and search effectiveness. Therefore, selecting suitable methods for new solution construction based on the specific problem context is crucial. Relying solely on random generation methods would inevitably lead to low search efficiency. Hence, we embrace knowledge-based new solution construction methods, employing three heuristic rules to generate new solutions. The following three heuristic rules were used in the knowledge-guided solution construction strategy.

- Maximum Profit
  From the perspective of maximizing benefits, the satellite removes the task with the slightest benefit in the current solution. Then, the task with the highest current benefit is selected from the task pool and inserted into the schedule, thereby constructing a new solution.
- Unit Profit
  Construct the new solution by removing the k_n tasks with the most minor unit gains from the current solution. Then, tasks with more considerable unit gains are selected from the current task sequence and inserted into the solution. The unit gain is defined as the gain of a problem divided by its observation time.
- Conflict Level
  Construct the new solution by removing the k_n tasks with the highest conflict degrees from the current solution. Then, tasks with lower conflict degrees from the task pool are inserted into the solution. The conflict degree is defined as the conflict time between a current task's visible time window and other tasks' visible time windows.

3.4. Frequent Pattern Data Mining

Data mining can enhance algorithmic efficiency by uncovering specific patterns that frequently occur in solution sets. The results of data mining are treated as knowledge.

In this context, the proposed frequent pattern (FP)-based data mining method is employed [19]. The primary process can be summarized as follows:

- Step 1: Data transformation. In AEOS, the execution time of a task depends on its preceding tasks, making the identification of patterns in adjacent task executions crucial. These adjacent executions are defined as frequent patterns and are considered knowledge. As illustrated in Figure 3, this transformation process involves acquiring task execution data and identifying frequent patterns.
- Step 2: Data mining. To obtain a deep insight into AEOS scheduling, we employ the FP-growth algorithm based on the Apriori principle. This algorithm efficiently explores frequent patterns by storing the dataset in a frequent pattern tree (FP-tree), as depicted in Figure 4. The process involves constructing a tree structure to store task data, facilitating efficient mining of frequent patterns. As the scale of tasks increases, the number of frequent patterns grows, resulting in increased computation time required for the FP-growth algorithm to generate results. Tasks are partitioned into smaller subsets (as shown in Figure 5), and the FP-growth algorithm is applied to each subset individually to speed up the search. This approach reduces computation time and enhances the accuracy of frequent pattern mining.
- Step 3: New solutions construction. Based on the selected frequent patterns. We construct new solutions and employ a roulette strategy to select high-quality solutions to guide the algorithm. Initially, a set of potential new task execution sequences is generated using knowledge extracted from frequent patterns. Subsequently, using the roulette strategy, high-quality solutions are chosen based on their performance metrics, guiding the optimization process of the algorithm. This step aims to improve the overall efficiency and effectiveness of the algorithm by leveraging known frequent patterns and high-quality solutions.

elite1	1	3	4	7	8	15
elite2	1	3	5	6	8	15
elite3	2	4	7	8	15	16

⟶

elite1	1, 3	3, 4	4, 7	7, 8	8, 15
elite2	1, 3	3, 5	5, 6	6, 8	8, 15
elite3	2, 4	4, 7	7, 8	8, 15	15, 16

Figure 3. Data transformation.

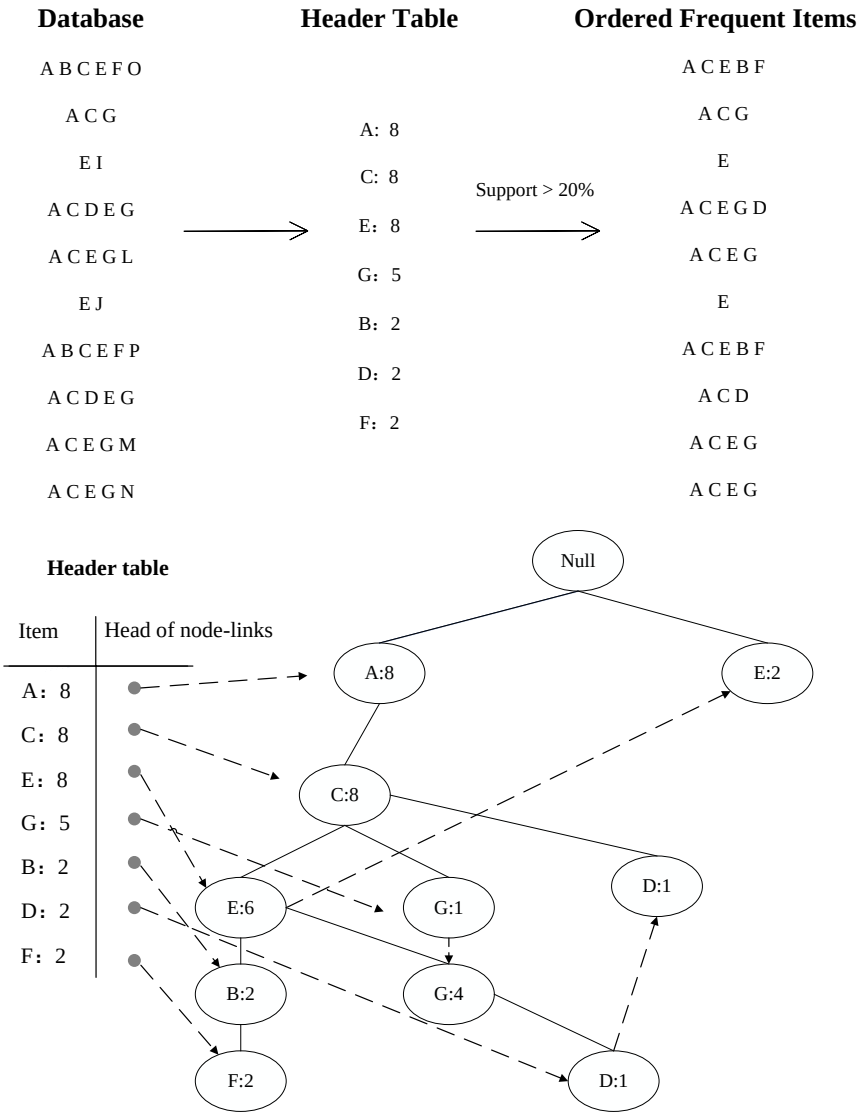


Figure 4. Construction of a frequent pattern tree.

3.5. Computational Complexity Analysis

The complexity of KG-PHLS is $O(k \cdot n \log n + n^2)$, where n denotes the number of tasks. Specifically, the heuristic-based initial solution generation method has a $O(n \log n)$ complexity. The foundational hill climbing algorithm has a time complexity of $O(k \cdot n)$. In contrast, the updated solution generation in the enhanced hill climbing algorithm has a $O(n \log n)$ complexity. The deferred strategy judgment incurs a complexity of $O(n)$, and the time complexity of the tabu search is $O(T)$, where T represents the length of the tabu list. Consequently, the overall time complexity of the parallel local search algorithm is $O(k \cdot n \log n)$. The FP-growth data mining process operates with a $O(n^2)$ complexity. Among these steps, the dominant factors contributing to the time complexity are the parallel local search algorithm and the data mining, resulting in an overall complexity of $O(k \cdot n \log n + n^2)$.

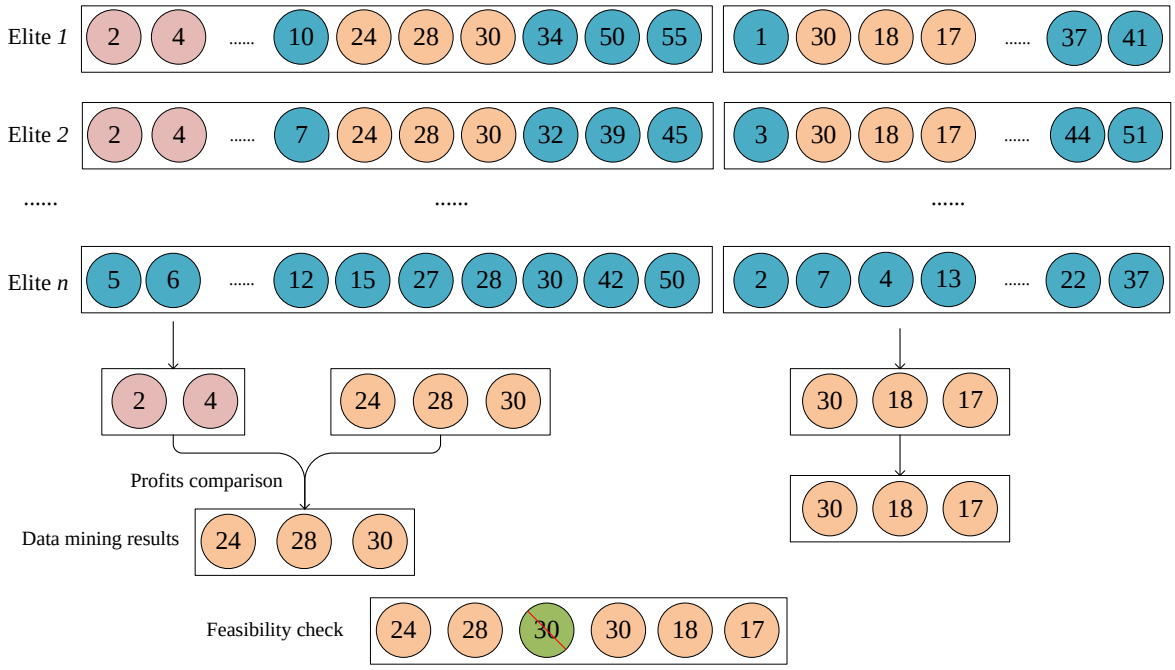


Figure 5. Frequent pattern mining to AEOS scheduling problem.

4. Experimental Study

An experimental study on a dataset with 15 scenarios validated the effectiveness of the proposed KG-PHLS algorithm. The proposed algorithm's performance was analyzed by comparing it with the other two algorithms. The experimental environment and data used are introduced first. Subsequently, the algorithm is tested using typical scenarios with various task scales, and the performance of each algorithm in terms of profit and runtime is provided.

4.1. Experimental Design

Due to the varying capabilities of different satellites and significant differences in constraints and management modes of satellite management systems among countries, providing a uniform standard for scheduling AEOS is not feasible. The experiments compared the more effective agile satellite scheduling algorithms found in the current research literature, i.e., ALNS-I [16] and ALNS-TPF [17].

In our study, we considered the performance of the same model (AS-01) agile satellite as described in ALNS-I [16]. Therefore, we adopted a similar construction method to validate the efficiency of the proposed algorithm. A total of 15 scenarios are used in the dataset, with the number of tasks ranging from 50 to 400 with an increment of 25, i.e., the numbers of the tasks are $[50, 75, 100, \cdots, 375, 400]$. The scheduling period is from 2013/04/20/00:00:00 to 2013/04/20/23:59:59, and the targets are distributed in the range of 3° N–53° N and 74° E–133° E. Each target's profit and observation duration are uniformly distributed within $[1, 10]$ and $[15, 30]$ seconds, respectively.

We evaluated the algorithms using an equal number of iterations to ensure a fair comparison of experimental results. All solution results are obtained from 10 independent experimental runs for each case. The values of other parameters of the algorithm are as follows:

- M = 2400, E = 2400;
- length for $k_n = 0.1S$, length of $U_t = 0.1S$, length of $U_L = 0.05S$, where S represents the number of tasks;
- maximum number of elite solutions = 50, maximum number of consecutive generations without improvement = 50;
- number of solutions for data mining = 10, maximum number of generations = 100, number of threads = 5.

4.2. Experimental Results

Table 3 and Figure 6 illustrate the comparison results of the profits obtained from three different algorithms. The results are the average number of 10 runs. It is evident that when the task scale is within 300, algorithm profits do not differ much. However, as the size of the instances increases, the differences between the algorithms become more pronounced. These differences are attributed to improvements in the KG-PHLS algorithm, which include enhanced methods for constructing new solutions during local search, the incorporation of tabu strategies, and the use of FP-growth-based data mining methods. These enhancements boost the algorithm's optimization and global exploration capabilities, enabling exceptional performance in solving large-scale instances.

Table 4 and Figure 7 present the comparison results of the computation time among three different algorithms. The results are the average times on ten independent runs. Concerning computation time, the ALNS-I algorithm exhibits lower efficiency than the other two algorithms when the instance size exceeds 75. The difference in efficiency between the ALNS-TPF algorithm and the KG-PHLS algorithm is insignificant.

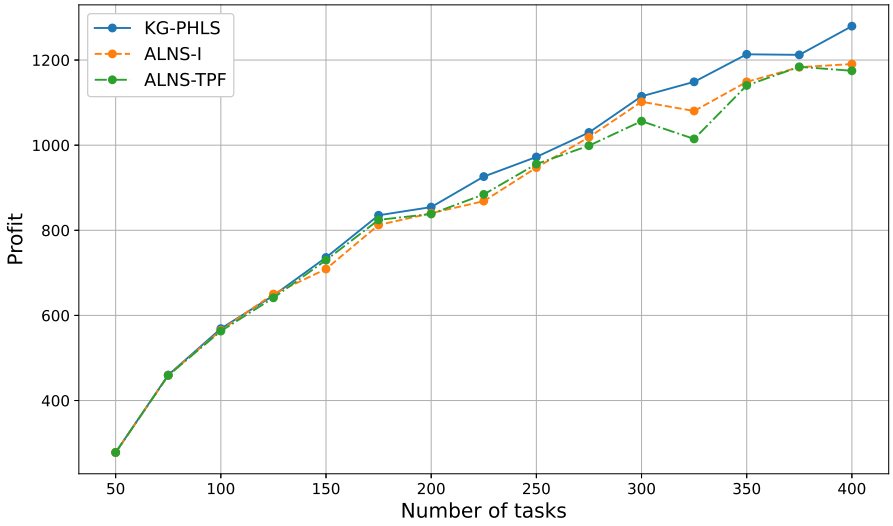


Figure 6. Results comparison on values of profit.

Table 3. Results of profit with different algorithms.

Number of Tasks	Profit		
	ALNS-I	ALNS-TPF	KG-PHLS
50	**278.0**	**278.0**	**278.0**
75	459.3	459.0	**460.0**
100	565.2	563.2	**568.9**
125	**650.2**	641.3	646.7
150	709.0	729.9	**736.1**
175	812.5	824.5	**835.3**
200	840.1	838.3	**854.5**
225	868.5	884.3	**926.2**
250	947.3	955.7	**972.3**
275	1018.7	998.8	**1029.8**
300	1102.0	1056.4	**1114.8**
325	1080.3	1014.7	**1148.8**
350	1148.8	1140.3	**1213.5**
375	1183.2	1184.1	**1212.2**
400	1190.6	1174.8	**1279.6**

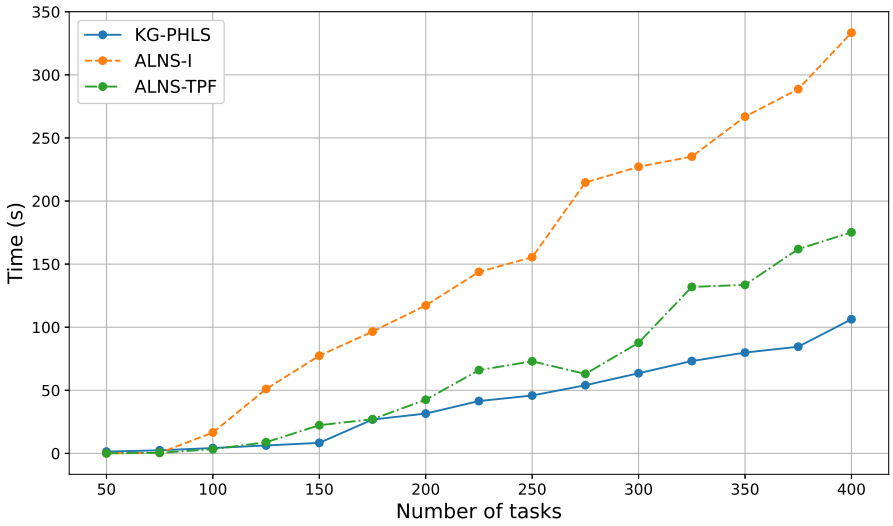


Figure 7. Results comparison on computation time.

Table 4. Computation times with different algorithms.

Number of Tasks	Computation Times (s)		
	ALNS-I	ALNS-TPF	KG-PHLS
50	0.06	**0.05**	1.40
75	**0.35**	0.47	2.42
100	16.52	**3.28**	4.22
125	51.01	8.80	**6.25**
150	77.38	22.30	**8.32**
175	96.54	27.06	**26.80**
200	117.23	42.49	**31.53**
225	143.85	65.95	**41.46**
250	155.44	72.96	**45.83**
275	214.68	62.97	**54.00**
300	227.15	87.64	**63.51**
325	235.21	131.92	**73.17**
350	266.89	133.55	**79.87**
375	288.70	161.92	**84.54**
400	333.43	175.20	**106.26**

As the instance size increases beyond 175, the KG-PHLS algorithm demonstrates significantly higher efficiency than the ALNS-TPF and ALNS-I algorithms. This outcome is due to the knowledge-guided new solution construction method and parallel computing. By leveraging domain-specific expertise or prior information, the algorithm can focus more effectively on regions of the solution space that may contain high-quality solutions, thereby reducing the search for invalid solutions. Through parallel computing, more tasks can be processed within the same time frame, accelerating the algorithm's execution speed.

Higher deletion ratios enable the algorithm to escape local optima and explore the solution space more extensively, potentially leading to superior solutions. Conversely, lower deletion ratios restrict the algorithm's exploration range, increasing the likelihood of becoming trapped in local optima and resulting in lower yields. We tested these parameters at data scales of 300 and 400. Similar characteristics apply to the size of the deferred table: a large deferred table can constrain the algorithm's exploration capabilities. At the same time, a deferred table that is too small may prematurely trap the algorithm in local optima. Moreover, deferred table size influences algorithm stability somewhat, although this effect was not consistently significant during our experiments.

From Table 5, we observed that increasing the deletion ratio and extending the deferred table did not significantly improve outcomes despite increasing computational time. Therefore, we opted for the parameter combination $k_n = 0.1S$ and $U_L = 0.05S$, where S denotes the data scale as our preferred setup.

Table 5. Sensitivity analysis of the design decisions.

Number of Tasks	Parameter Value		Profit	Computation Times (s)
	k_n	U_L		
300	0.1	0.05	1114.80	63.51
	0.05	0.05	1089.00	**62.69**
	0.2	0.05	**1149.80**	85.78
	0.1	0.02	1109.20	65.51
	0.1	0.1	1118.60	65.20
400	0.1	0.05	1279.60	106.26
	0.05	0.05	1252.20	**98.58**
	0.2	0.05	**1292.00**	130.61
	0.1	0.02	1271.80	114.12
	0.1	0.1	1273.00	109.88

5. Conclusions

This paper studied agile Earth observation satellite (AEOS) scheduling problems with temporal dependencies. The current scheduling methods make it difficult to solve large-scale AEOS scheduling problems efficiently. Thus, we have proposed a knowledge-guided parallel hybrid local search algorithm to enhance the search efficiency of AEOS scheduling.

We improved the algorithm's global search and optimization capabilities by leveraging prior satellite knowledge to enhance new solution generation in local searches and incorporating deferred mechanisms and a tabu strategy into hill climbing algorithms. Additionally, using FP-growth-based data mining methods enables the algorithm to obtain high-quality solutions quickly. The testing results have revealed that the proposed KG-PHLS algorithm outperforms the other algorithms in terms of profit and computation time on various scenario instances, particularly for large-scale problems.

More complex tasks could be completed via complex and heterogeneous satellite networks, which include various types of satellites. Our future research directions could include:

- Expanding from single-satellite scheduling to multi-satellite scheduling necessitates considering additional constraints and optimization objectives such as load balancing and integrating satellite data transmission into scheduling to unify both aspects for joint scheduling.
- We need to address more uncertainties, such as the cloud cover, urgent task arrival, and their impact on agile satellite scheduling.

Author Contributions: Conceptualization, Y.S. and S.C.; methodology, Y.S. and S.C.; coding, Y.S.; validation, Y.S. and X.W.; investigation, Y.S.; writing—original draft preparation, Y.S. and S.C.; writing—review and editing, X.W., S.C., M.Z. and L.X.; visualization, Y.S. and S.C.; supervision, S.C.; project administration, L.X.; funding acquisition, S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant no. 61806119), the Natural Science Basic Research Plan in Shaanxi Province of China (no. 2024JC-YBMS-516), Fundamental Research Funds for the Central Universities (no. GK202201014).

Data Availability Statement: The code will be open source after the paper is accepted.

Acknowledgments: Thanks to the anonymous reviewers and editor from the journal.

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

Abbreviations

The following abbreviations are used in this manuscript:

AEOS	agile Earth observation satellite
EOS	Earth observation satellite
FP	frequent pattern
KG-PHLS	knowledge-guided parallel hybrid local search algorithm
VTW	visible time window

References

1. Xu, L.; Lu, J.; Zhan, Y. Scheduling Optimization of Compound Operations in Autonomous Vehicle Storage and Retrieval System. *Symmetry* **2024**, *16*, 168. [CrossRef]
2. Wu, G.; Luo, Q.; Du, X.; Chen, Y.; Suganthan, P.N.; Wang, X. Ensemble of Metaheuristic and Exact Algorithm Based on the Divide-and-Conquer Framework for Multisatellite Observation Scheduling. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 4396–4408. [CrossRef]
3. Cui, K.; Xiang, J.; Zhang, Y. Mission planning optimization of video satellite for ground multi-object staring imaging. *Adv. Space Res.* **2018**, *61*, 1476–1489. [CrossRef]
4. Peng, G.; Dewil, R.; Verbeeck, C.; Gunawan, A.; Xing, L.; Vansteenwegen, P. Agile earth observation satellite scheduling: An orienteering problem with time-dependent profits and travel times. *Comput. Oper. Res.* **2019**, *111*, 84–98. [CrossRef]
5. Chu, X.; Chen, Y.; Tan, Y. An anytime branch and bound algorithm for agile earth observation satellite onboard scheduling. *Adv. Space Res.* **2017**, *60*, 2077–2090. [CrossRef]
6. Chen, X.; Reinelt, G.; Dai, G.; Spitz, A. A mixed integer linear programming model for multi-satellite scheduling. *Eur. J. Oper. Res.* **2019**, *275*, 694–707. [CrossRef]
7. Xiao, Y.; Zhang, S.; Yang, P.; You, M.; Huang, J. A two-stage flow-shop scheme for the multi-satellite observation and data-downlink scheduling problem considering weather uncertainties. *Reliab. Eng. Syst. Saf.* **2019**, *188*, 263–275. [CrossRef]
8. Lemaître, M.; Verfaillie, G.; Jouhaud, F.; Lachiver, J.M.; Bataille, N. Selecting and scheduling observations of agile satellites. *Aerosp. Sci. Technol.* **2002**, *6*, 367–381. [CrossRef
9. Habet, D.; Vasquez, M.; Vimont, Y. Bounding the optimum for the problem of scheduling the photographs of an Agile Earth Observing Satellite. *Comput. Optim. Appl.* **2010**, *47*, 307–333. [CrossRef]
10. Baioletti, M.; Milani, A.; Santucci, V. An Algebraic Approach for the Search Space of Permutations with Repetition. In *Evolutionary Computation in Combinatorial Optimization*; Paquete, L., Zarges, C., Eds.; Springer: Cham, Switzerland, 2020; pp. 18–34. [CrossRef]
11. Tasgetiren, M.F.; Kizilay, D.; Pan, Q.K.; Suganthan, P. Iterated greedy algorithms for the blocking flowshop scheduling problem with makespan criterion. *Comput. Oper. Res.* **2017**, *77*, 111–126. [CrossRef]
12. Zhang, H.; Li, H.; Tam, C. Particle swarm optimization for resource-constrained project scheduling. *Int. J. Proj. Manag.* **2006**, *24*, 83–92. [CrossRef]
13. Wu, G.; Liu, J.; Ma, M.; Qiu, D. A two-phase scheduling method with the consideration of task clustering for earth observing satellites. *Comput. Oper. Res.* **2013**, *40*, 1884–1894. [CrossRef]
14. Hu, X.; Zhu, W.; An, B.; Jin, P.; Xia, W. A branch and price algorithm for EOS constellation imaging and downloading integrated scheduling problem. *Comput. Oper. Res.* **2019**, *104*, 74–89. [CrossRef]
15. Wolfe, W.J.; Sorensen, S.E. Three Scheduling Algorithms Applied to the Earth Observing Systems Domain. *Manag. Sci.* **2000**, *46*, 148–166. [CrossRef]
16. Liu, X.; Laporte, G.; Chen, Y.; He, R. An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time. *Comput. Oper. Res.* **2017**, *86*, 41–53. [CrossRef]
17. He, L.; de Weerdt, M.; Yorke-Smith, N. Time/sequence-dependent scheduling: The design and evaluation of a general purpose tabu-based adaptive large neighbourhood search algorithm. *J. Intell. Manuf.* **2020**, *31*, 1051–1078. [CrossRef]
18. He, L.; Liu, X.; Laporte, G.; Chen, Y.; Chen, Y. An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling. *Comput. Oper. Res.* **2018**, *100*, 12–25. [CrossRef]
19. Wu, J.; Song, B.; Zhang, G.; Ou, J.; Chen, Y.; Yao, F.; He, L.; Xing, L. A data-driven improved genetic algorithm for agile earth observation satellite scheduling with time-dependent transition time. *Comput. Ind. Eng.* **2022**, *174*, 108823. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.