Journal of Institute of Control, Robotics and Systems (2025) 31(5):510-516
https://doi.org/10.5302/J.ICROS.2025.25.0007 ISSN:1976-5622 eISSN:2233-4335

# ChatGPT 기반 지구 관측 위성 임무 스케줄링

# ChatGPT-based Mission Scheduling for Earth Observation Satellites

김 성 준, 김 민 채, 최 한 림*
(Sung Jun Kim, Minchae Kim, and Han-Lim Choi[1,*])
[1]Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology

**Abstract:** Scheduling for Earth observation satellites has drawn significant attention through various optimization algorithms. Advancements in artificial intelligence are leading to new approaches, yet the application of large language models (LLMs) in satellite scheduling is still largely uncharted. This paper presents an approach using ChatGPT, a popular LLM, to schedule tasks for multiple low Earth orbit satellites. We formulate the problem using mixed integer linear programming and demonstrate how ChatGPT can generate near-optimal solutions in a fraction of the time required by exact solvers. Our simulations show that ChatGPT can quickly produce schedules that meet mission constraints, occasionally matching optimal outcomes. These findings suggest potential for more adaptive and efficient scheduling methods in Earth observation, opening opportunities for further exploration of LLM-based techniques.

**Keywords:** satellite mission scheduling, large language model, Earth observation, task assignment

## I. INTRODUCTION

Data collected from Earth observation satellites play a vital role in various fields, including meteorological forecasting, environmental protection, disaster monitoring, military reconnaissance, and missile early warning [1-3]. As the importance of such data grows, the demand for satellite observations has steadily increased, accompanied by a rise in the number of satellites in operation. To efficiently carry out diverse observation missions using multiple satellites, the application of scheduling techniques has become essential.

The Earth observation satellite scheduling problem is a combinatorial optimization challenge that involves generating an optimal schedule for performing observation tasks while utilizing limited resources efficiently. Due to numerous constraints and the necessity for optimization, various approaches have been explored to solve this problem, including heuristic, metaheuristic, exact optimization, and learning-based methods. Given the computational complexity and extensive constraints involved, efficiently searching for high-quality solutions is crucial in satellite scheduling. Many heuristic and metaheuristic algorithms have been proposed for this purpose. For instance, Habet et al. [4] applied Tabu Search to maximize the number of images captured by satellites, while Wang et al. [5] developed a scheduling approach using satellite constellations to ensure Earth observations remain unaffected by weather conditions. Their method improved upon traditional greedy approaches by integrating priority-based heuristics and backtracking techniques. Similarly, He et al. [6] addressed the challenge of excessive search space expansion in multi-satellite scheduling by introducing an adaptive task assignment strategy, followed by the application of Adaptive Large Neighborhood Search (ALNS) to refine scheduling solutions. Beyond maximizing observation efficiency, some studies have also focused on ensuring fair resource allocation. Tangpattanakul et al. [7,8] framed the scheduling problem as a multi-objective optimization task to prevent resource allocation bias toward specific users. Instead of using conventional genetic algorithms, they introduced an improved random-key-based genetic algorithm. Meanwhile, Xu et al. [9] tackled the problem by defining a novel priority index that accounts for the importance of each task and employed Ant Colony Optimization (ACO) to optimize scheduling accordingly. Despite the effectiveness of heuristic and metaheuristic methods, they are often tailored to specific problem instances, relying on predefined search operators and solution structures.

Exact algorithms offer a significant advantage in satellite scheduling as they do not require customized search operators or solution structures for each problem instance, while also guaranteeing optimal solutions. Due to these benefits, many studies have explored the application of exact methods to satellite scheduling problems [10,11]. For instance, Cho et al. [12] formulated the satellite scheduling problem as a Binary Integer Linear Programming (BILP) model to maximize overall mission performance in satellite constellations and optimize task allocation and start times. An optimal solution was obtained using a Mixed-Integer Linear Programming (MILP) solver. Similarly, Chen et al. [13] formulated the scheduling problem using an MILP model to maximize mission value and task execution while ensuring optimality. Kim et al. [14] addressed the scheduling of image capture missions for satellite constellations consisting of Synthetic Aperture Radar (SAR) and Electro-Optical (EO) sensors flying in formation, using an MILP approach. Subsequently, Kim et al. [15] extended this work by incorporating satellite revisit time into the scheduling process and proposing a method to minimize it. Despite their ability to guarantee optimal solutions, exact algorithms suffer from a major drawback: as

* Corresponding Author
Manuscript received January 22, 2025; revised March 24, 2025; accepted April 7, 2025
김성준: 한국과학기술원 항공우주공학과 박사과정(sungjunkim@kaist.ac.kr, ORCID 0000-0003-1949-5042)
김민채: 한국과학기술원 항공우주공학과 박사과정(mckim@lics.kaist.ac.kr, ORCID 0009-0000-2109-581X)
최한림: 한국과학기술원 항공우주공학과 교수(hanlimc@kaist.ac.kr, ORCID 0000-0003-3985-0419)
※ This work was supported by Korea Research Institute for defense Technology planning and advancement(KRIT) grant funded by the Korean government (DAPA(Defense Acquisition Program Administration)) (KRIT-CT-22-040, Heterogeneous Satellite constellation based ISR Research Center, 2022).

problem size increases, computational complexity and processing time grow exponentially. This scalability issue makes real-time operation challenging.

To address these challenges, learning-based satellite scheduling approaches have been actively studied. These algorithms make decisions based on pre-trained policies, allowing real-time decision-making without extensive computations. As a result, they can handle large-scale problems while adapting flexibly to dynamic changes. Based on this approach, Wei et al. [3] proposed a deep reinforcement learning (DRL)-based satellite scheduling method, where an encoder-decoder structured neural network directly generates optimized schedules without iterative computations. Similarly, Wang et al. [16] addressed the slow learning speed and low accuracy of conventional actor-critic algorithms by applying an encoder-decoder neural network combined with a reinforcement learning-based scheduling method. Liu et al. [17] and He et al. [18] tackled large-scale problems such as multi-satellite environments, which are difficult to handle using traditional Q-learning, by adopting Deep Q-Networks (DQN) for satellite scheduling. Furthermore, Huang et al. [19] overcame the limitation of DQN, which only operates in discrete action spaces, by implementing a Deep Deterministic Policy Gradient (DDPG)-based approach. This method prioritizes high-priority tasks while minimizing satellite rotation angles to reduce energy consumption. However, reinforcement learning-based models require training in specific environments, and additional learning is necessary when applying them to new problems. Moreover, optimizing rewards often demands a large number of simulations, which poses a significant limitation.

With the emergence of GPT, research on leveraging large language models (LLMs) has significantly expanded in recent years [20-24]. LLMs are trained on vast amounts of text data and possess emergent abilities that enable them to handle tasks beyond their training data. Notably, they excel in Few-Shot and Zero-Shot learning, allowing them to perform new tasks effectively with minimal or no additional training [25]. Building on these capabilities, LLMs are being utilized not only as text and code generators but also as central planners and macro-level decision-makers [26-28]. Accordingly, this study aims to explore the application of LLMs in satellite scheduling by proposing a ChatGPT-based scheduling algorithm.

The rest of the paper is organized as follows. Sec. II defines the Earth observation mission scheduling problem and the assumptions underlying it. Sec. III provides an MILP formulation of the problem. Sec. IV explains how ChatGPT is employed to generate feasible schedules, and Sec. V presents a performance analysis based on simulation results. Finally, Sec. VI and VII discuss the key insights gained from this research and offer a summary along with future directions.

## II. EARTH OBSERVATION SATELLITE SCHEDULING

This section describes the scheduling problem of Earth observation satellites with several assumptions regarding the mission.

### 1. Problem Description

Earth observation satellites operating in low Earth orbit can capture images within specific time windows during which their imaging sensors maintain a clear line of sight to the designated targets. In this study, imaging is assumed to be feasible whenever the satellite's elevation angle from the target's perspective exceeds 5 degrees. Fig. 1 illustrates a three-dimensional visualization of a satellite's orbit and the ground targets. Also, Fig. 2 shows an example of the two-dimensional map of the ground targets.

그림 1. 지구 관측 위성의 궤도 및 표적의 3D 시각화 예시.

Fig. 1. 3D visualization of the orbits of Earth observation satellites and targets.

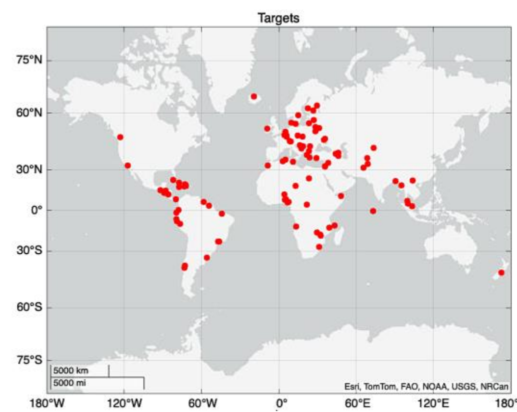



그림 2. 세계 도시 표적의 2D 지도 예시.

Fig. 2. 2D map of the world city targets.

The primary objective of scheduling in this context is to maximize the number of requested targets imaged within the allotted time. These time windows are precomputed via simulation before scheduling and are supplied in CSV file format.

### 2. Assumptions

Below are the key assumptions adopted for the scheduling problem to ensure tractability and clarity in this study:

**Constant setup time:** The setup time between two consecutive observations is assumed to be a fixed constant.

**Uniform processing time:** The processing time for each observation is considered constant across all targets.

**Single task per satellite:** Each satellite can only process one task at a time.

**Non-preemptive tasks:** Once a task has started, it must be completed without interruption.

**Exclusion of downlink and memory constraints:** Downlink operations and onboard memory limitations are not accounted for in this work.

## III. MILP FORMULATION

This section presents the MILP formulation of the problem described in the previous section. The objective function and relevant mission constraints are specified in accordance with the mission requirements, allowing the formulated problem to be solved using standard commercial optimizers.

### 1. Objective function

The objective of this observation mission is to capture images of as many targets as possible within a given time horizon. Accordingly, the MILP formulation of the scheduling problem includes an objective function designed to maximize the number of observed targets under the specified constraints.

$$J = \max \sum_{s\in S} \sum_{j\in J} \sum_{k\in K} x_{s,j,k} \tag{1}$$

Here, $x_{s,j,k}$ is a binary decision variable that is equal to 1 if a satellite $s$ is scheduled to image target $j$ at the $k^{th}$ time window of the satellite-target pair. The objective is to maximize the number of scheduled targets among all time windows by every possible satellite-target pairs.

### 2. Mission Constraints

Next, the mission requirements are represented by three sets of constraints, each addressing a different aspect of the scheduling problem.

**Time window constraints:** Each observation task must begin and conclude within a designated time window during which the satellite has visibility of the target. The constraints enforcing this requirement are formulated as follows.

$$t_{s,j,k} \le M \cdot x_{s,j,k} \tag{2}$$

$$i_{s,j,k} - t_{s,j,k} \le M(1 - x_{s,j,k}) \tag{3}$$

$$t_{s,j,k} + t^{Img} - d_{s,j,k} \le M(1 - x_{s,j,k}) \tag{4}$$

$$\forall\, s \in S, j \in J, k \in K_{s,j}$$

Here, $t_{s,j,k}$ is the start time of the observation task by a satellite $s$ to image a target $j$ at the $k^{th}$ time window. This variable is conditionally activated by the constraint (2) using the big M method when the decision variable $x_{s,j,k}$ is equal to 1. By similar method, the constraints (3) and (4) restricts the task start time to be within the start time $i_{s,j,k}$ and end time $d_{s,j,k}$ of a time window considering the imaging time $t^{Img}$.

**Non-overlap constraints:** Multiple tasks must not overlap on a single satellite. Specifically, each subsequent task must begin only after the previous task's imaging time plus the required setup time. This requirement is captured by the following set of constraints.

$$y_{s,j_1,k_1,j_2,k_2} + y_{s,j_2,k_2,j_1,k_1} = x_{s,j_1,k_1} \cdot x_{s,j_2,k_2} \tag{5}$$

$$t_{s,j_1,k_1} + t^{Img} + t^{Setup} - t_{s,j_2,k_2} \le M(1 - y_{s,j_1,k_1,j_2,k_2}) \tag{6}$$

Here, an auxiliary binary variable $y_{s,j_1,k_1,j_2,k_2}$ is the indicator that the observation task at time window $(\mathrm{s}, \mathrm{j}_1, \mathrm{k}_1)$ is done before the task at time window $(s, j_2, k_2)$. Constraint (5) leads either $y_{s,j_1,k_1,j_2,k_2}$ or $y_{s,j_2,k_2,j_1,k_1}$ to be equal to 1 when both tasks at $(s, j_1, k_1)$ and $(s, j_2, k_2)$ are assigned since only one of either direction is possible. This variable is then used for the conditional constraint (6) such that the imaging time and the setup time is secured between two tasks that are assigned.

The right-hand side of Constraint (5) is quadratic, so linearization is required to include it in an MILP. A simplified example of this linearization process is shown below.

$$y_1 + y_2 = x_1 \cdot x_2 \tag{7}$$

Now, the single quadratic constraint can be expressed as three linear constraints as follows.

$$y_1 + y_2 \le x_1 \tag{8}$$

$$y_1 + y_2 \le x_2 \tag{9}$$

$$y_1 + y_2 \ge x_1 + x_2 - 1 \tag{10}$$

These set of linear constraints function the same such that both variables $y_1$ and $y_2$ are equal to 0 when either $x_1$ or $x_2$ are equal to 0 (meaning either task is not assigned), and the left-hand side can be no more than 1 when both $x_1$ and $x_2$ are equal to 1.

**Maximum assignment constraints:** Finally, the following constraint is imposed to limit the number of assignments for each target.

$$\sum_{s\in S} \sum_{k\in K_{s,j}} x_{s,j,k} \le 1 \quad \forall\, j \in J \tag{11}$$

Constraint (11) ensures that each target can be assigned at most once across all available time windows.

## IV. ChatGPT-BASED SCHEDULING

In order to employ a language model for scheduling, user requirements must be articulated as textual prompts. If the model has not been fine-tuned for these specific tasks, it may generate schedules that deviate from the intended requirements, thus necessitating explicit inclusion of both mission objectives and constraints. Moreover, extended interactions can surpass the context window during fine-tuning, resulting in catastrophic forgetting of previously discussed content. To address this issue, the present study conducted scheduling in a one-shot manner by providing all requisite conditions simultaneously upon initiating a new ChatGPT session.

Providing instructions to ChatGPT in the form of bullet-pointed complete sentences has known to be more effective for conveying

표 1. CSV 파일을 사용하여 ChatGPT에서 임무 스케줄링을 실행하기 위한 프롬프트.

Table 1. Prompt used to perform mission scheduling on ChatGPT with a given time window data file in CSV format.

| |
|---|
| You will perform scheduling based on the data in this CSV file.<br>- The columns in the CSV file represent:<br>- Column 1: Index of the satellite<br>- Column 2: Index of the target<br>- Column 3: Index of the time window for the satellite-target pair<br>- Column 4: Duration of the time window<br>- Column 5: Start time of the time window<br>- Column 6: End time of the time window<br>- The objective is to schedule as many missions as possible within the given time while satisfying the following constraints.<br>- Once scheduled, each task will last 60 seconds.<br>- The task's start time must be within the given time window, and its end time must also fall within the time window after 60 seconds.<br>- For the same satellite, it requires 30 seconds of setup time between two consecutive tasks.<br>- Each target must be scheduled exactly once, and no other satellite can capture the same target again.<br>- A satellite cannot perform multiple tasks at the same time, meaning there should be no conflict between different tasks assigned to the same satellite. |

표 2. 스케줄 결과를 시각화하기 위한 ChatGPT 프롬프트.

Table 2. ChatGPT Prompt used to visualize the schedule results.

| |
|---|
| After scheduling, visualize the results using a bar graph:<br>- The y-axis should represent the target index.<br>- The x-axis should represent time.<br>- Use different colors for each satellite, ensuring that the colors are clearly distinguishable.<br>- Use the following color codes for the bars: #FF8080, #80FF80, #8080FF, #FFFF80<br>- Ensure the scheduled missions are clearly visible with bold-colored bars.<br>- Plot all time windows for each target. Even if they were not scheduled, using the same color as the corresponding satellite.<br>- Time windows should be drawn as boxes with the satellite's color.<br>- The height of the boxes for the scheduled missions must be longer than the boxes for the time windows.<br>- Apply black color edge on all the boxes for the scheduled mission and the time windows.<br>- The legend must contain all satellites. Don't repeat the same label in the legend. |

requirements than using standard paragraphs. Furthermore, including external data files and describing their contents in detail can enhance clarity and ensure the model accurately interprets the provided information. Table 1 lists the scheduling prompt, where time window data was supplied in a CSV file. The prompt includes the description of the time window data and the mission constraints introduced in the previous section.

In addition, Table 2 provides an example prompt for visualizing the scheduling results as a plot, where the target index appears on the y-axis and time on the x-axis. Different-colored boxes represent the time windows for each satellite, while longer boxes indicate the scheduled tasks.

## V. SIMULATION RESULTS

In this section, we present the mission scheduling results for Earth observation satellites obtained using the two methods discussed earlier and compare ChatGPT's performance against the optimal MILP solution. The time windows were computed by the Aerospace Toolbox in MATLAB. Schedules generated by the MILP approach were optimized on an Apple Macintosh system equipped with an M1 Pro processor and 32GB of RAM, using the Gurobi Optimizer (v.11.0.2). The ChatGPT 4o model, included in ChatGPT version 1.2025.007, was utilized for this study.

표 3. 사례 연구를 위한 시뮬레이션 매개변수.

Table 3. Simulation parameters for the case study.

| Parameter | Value |
|---|---|
| Mission start time | 2025. 01. 13. 12:00:00, UTC |
| Time horizon (hour) | 8.0 |
| Time step (s) | 1.0 |
| Number of targets | 10 - 250 |
| Target location | World City Database |
| Number of satellites | 4 |
| Number of orbital planes | 4 |
| Phasing between the planes | 1 |
| Inclination (deg) | 97.8 |
| Altitude (km) | 600.0 |
| Imaging time (s) | 60.0 |
| Setup time (s) | 30.0 |
| Maximum view angle (deg) | 70.0 |

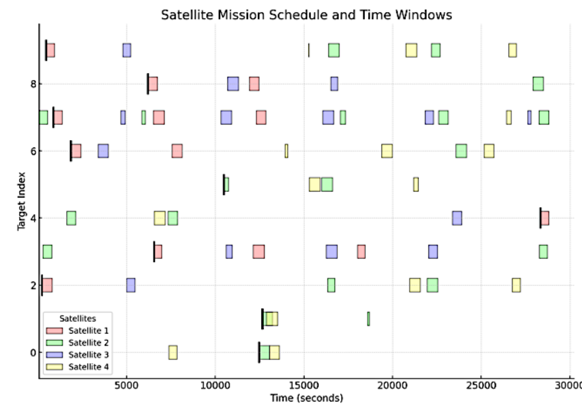



그림 3. ChatGPT로 그려진 스케줄 예시.

Fig. 3. Example schedule drawn by ChatGPT.

표 4. 그림 1에서 스케줄 예시의 세부 내용.

Table 4. Details of the example schedule shown in Fig. 1.

| Satellite | Target | TW Start Time (s) | Mission Start Time (s) | Mission End Time (s) | TW End Time (s) |
|---|---|---|---|---|---|
| 1 | 2 | 195 | 195 | 255 | 801 |
| 1 | 9 | 433 | 433 | 493 | 930 |
| 1 | 7 | 839 | 839 | 899 | 1,369 |
| 1 | 6 | 1,820 | 1,820 | 1,880 | 2,413 |
| 1 | 8 | 6,170 | 6,170 | 6,230 | 6,738 |
| 1 | 3 | 6,518 | 6,518 | 6,578 | 6,973 |
| 1 | 4 | 28,307 | 28,307 | 28,367 | 28,800 |
| 2 | 5 | 10,445 | 10,445 | 10,505 | 10,738 |
| 2 | 0 | 12,436 | 12,436 | 12,496 | 13,054 |
| 2 | 1 | 12,621 | 12,621 | 12,681 | 13,226 |

Table 3 summarizes the parameters used to simulate the satellite scheduling problem. The satellites are arranged in a Walker-Delta constellation with the specified parameters, and targets are randomly selected from a global city database containing over 10,000 entries of latitude and longitude. Figure 3 provides a visual representation of an example schedule generated by ChatGPT for ten targets with the described satellites. Table 4 details the start and end times for each satellite–target pair, demonstrating that ChatGPT successfully creates schedules that adhere to the mission constraints defined in the prompt.

The scheduling performance for varying numbers of targets was compared between ChatGPT and the MILP approach. Figure 4 shows the number of scheduled targets and the corresponding computation

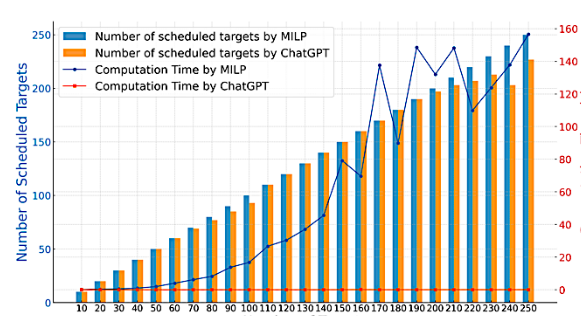



그림 4. 표적 개수에 따른 스케줄 성능 비교.

Fig. 4. Comparison of schedule performance by number of targets.

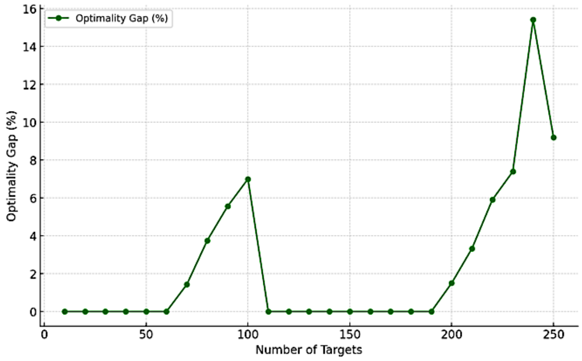



그림 5. ChatGPT를 통해 스케줄 된 표적 개수에 대한 최적해 차이 백분율.

Fig. 5. Optimality gap percentage of the number of scheduled targets by ChatGPT.

times. Despite the growing number of targets, ChatGPT consistently kept computation times under one second, whereas the exact MILP method scaled exponentially. Figure 5 presents the optimality gap for ChatGPT's schedules, calculated relative to the optimal MILP solutions. Although the gap began to widen from 60 targets onward, ChatGPT achieved optimal solutions between 110 and 190 targets—likely a result of having more feasible targets to schedule. Beyond 200 targets, however, ChatGPT's solutions diverged further from the optimum, indicating some performance limitations.

Such limitation is particularly evident when observing the outcomes of repeated experiments. Fig. 6 illustrates the scheduling results obtained by performing ten trials for each target scenario using ChatGPT, with a fixed random seed. Although the mean values of scheduled targets generally remain near-optimal or sub-optimal, the presence of outliers—depicted as circles—clearly demonstrates that scheduling performance can occasionally experience significant degradation. In particular, in the 150-target scenario, outliers exhibiting an optimality gap as high as 76% appeared twice. The exact match in the number of scheduled targets between these two occurrences strongly indicates that the same underlying scheduling algorithm was employed in both instances.

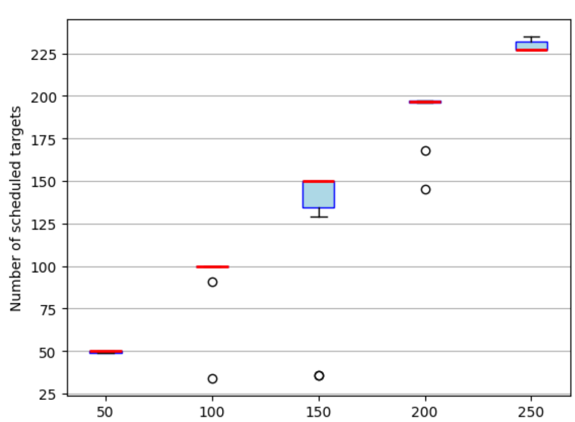



그림 6. 동일한 스케줄링 시나리오에 대해 10회 반복 실험 시 스케줄링된 표적의 개수.

Fig. 6. Number of scheduled targets over 10 trials for the same scheduling scenario.

## VI. DISCUSSIONS

This study examines the feasibility of using ChatGPT, a widely available LLM, for Earth observation satellite scheduling. Our results show that ChatGPT can satisfy scheduling constraints and reach near-or sub-optimal solutions faster than methods guaranteeing truly optimal outcomes. In some instances, it even achieved optimal solutions, indicating room for further improvement as the model or prompt engineering techniques evolve.

These findings suggest that ChatGPT could replace manually implemented heuristics or specialized schedulers, particularly when constraints and objectives change frequently. By relying on an open-source model, organizations can reduce development overhead and avoid extensive custom coding.

Nevertheless, several limitations exist. When accessing ChatGPT through an online server, the model may lose context over time, making it necessary to restate information for subsequent scheduling requests. It may also overlook constraints, such as the maximum number of assignable targets, requiring additional prompts or clarifications. In such cases, an additional one-shot prompt specifying the constraint to be satisfied normally resolved the issue and provided feasible solution. Also, we empirically speculate that the performance variability arises from ChatGPT employing multiple scheduling algorithms internally, selecting among them randomly in a black-box manner for each execution. It further appears that some of these internal algorithms may be relatively inefficient, as suggested by the identical scheduling results observed in multiple outlier cases. Future research should focus on stronger prompt engineering and local model deployments to maintain consistency and context.

Overall, this study demonstrates that LLMs can be adapted for satellite scheduling. Further work on local LLM usage and more robust prompt design could enhance reliability and pave the way for broader integration into Earth observation missions.

## VII. CONCLUSION

This paper presented an LLM-based approach for Earth observation satellite scheduling, along with corresponding simulation results. We formulated the scheduling problem for multiple low Earth orbit satellites as an MILP and demonstrated how ChatGPT, a widely used LLM, can generate schedules via a specialized prompt. The simulation findings show that, compared to optimal MILP solutions, ChatGPT can quickly produce near-optimal schedules. As a next step, we plan to explore local model deployments, apply fine-tuning for scheduling tasks, and incorporate additional constraints to further expand the utility of LLMs in the scheduling domain.

## REFERENCES

[1] G. Peng, G. Song, Y. He, J. Yu, S. Xiang, L. Xing, and P. Vansteenwegen, "Solving the agile earth observation satellite scheduling problem with time-dependent transition times," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 3, pp. 1614–1625, 2020.

[2] J. Li, C. Li, and F. Wang, "Automatic scheduling for earth observation satellite with temporal specifications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 4, pp. 3162–3169, 2020.

[3] L. Wei, Y. Chen, M. Chen, and Y. Chen, "Deep reinforcement

learning and parameter transfer based approach for the multi-objective agile earth observation satellite scheduling problem," *Applied Soft Computing*, Vol. 110, p. 107607, 2021.

[4] D. Habet, M. Vasquez, and Y. Vimont, "Bounding the optimum for the problem of scheduling the photographs of an agile earth observing satellite," *Computational optimization and applications*, Vol. 47, pp. 307–333, 2010

[5] P. Wang, G. Reinelt, P. Gao, and Y. Tan, "A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation," *Computers & Industrial Engineering*, Vol. 61, No. 2, pp. 322–335, 2011

[6] L. He, X. Liu, G. Laporte, Y. Chen, and Y. Chen, "An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling," *Computers & Operations Research*, Vol. 100, pp. 12–25, 2018.

[7] P. Tangpattanakul, N. Jozefowiez, and P. Lopez, "Multi-objective optimization for selecting and scheduling observations by agile earth observing satellites," *Parallel Problem Solving from Nature-PPSN XII: 12th International Conference*, Taormina, Italy, September 1-5, 2012, Proceedings, Part II 12, Springer, pp. 112–121, 2012.

[8] P. Tangpattanakul, N. Jozefowiez, and P. Lopez, "A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite," *European Journal of Operational Research*, vol. 245, no. 2, pp. 542–554, 2015.

[9] R. Xu, H. Chen, X. Liang, and H. Wang, "Priority-based constructive algorithms for scheduling agile earth observation satellites with total priority maximization," *Expert Systems with Applications*, vol. 51, pp. 195–206, 2016.

[10] C. H. Kim, S. J. Kim, and H. L. Choi, "Optimal Integrated Scheduling of Observation and Download Tasks for Multiple Satellites with Memory Constraints," *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, IEEE, pp. 2287–2294, 2024.

[11] S. J. Kim, M. Kim, C. H. Kim, and H. L. Choi, "Satellite scheduling framework of observation-relay-downlink with a mixed-integer linear programming model," *AIAA SCITECH 2025 Forum*, p. 2273, 2025.

[12] D. H. Cho, J. H. Kim, H. L. Choi, and J. Ahn, "Optimization-based scheduling method for agile earth-observing satellite constellation," *Journal of Aerospace Information Systems*, vol. 15, no. 11, pp. 611–626, 2018.

[13] X. Chen, G. Reinelt, G. Dai, and A. Spitz, "A mixed integer linear programming model for multi-satellite scheduling," *European Journal of Operational Research*, vol. 275, no. 2, pp. 694–707, 2019.

[14] S. J. Kim, C.-H. Kim, and H. L. Choi, "Optimal surveillance scheduling for multiple SAR-EO satellite train constellation," *AIAA AVIATION FORUM AND ASCEND 2024*, p. 4851, 2024.

[15] C. H. Kim, S. J. Kim, and H. L. Choi, "Heterogeneous satellite task scheduling with revisit time minimization using MILP formulation," *AIAA SCITECH 2024 Forum*, p. 0747, 2024.

[16] X. Wang, F. Zhao, Z. Shi, and Z. Jin, "Deep reinforcement learning-based periodic earth observation scheduling for agile satellite constellation," *Journal of Aerospace Information Systems*, vol. 20, no. 8, 2023, pp. 508–519.

[17] Y. Liu, Q. Chen, C. Li, and F. Wang, "Mission planning for Earth observation satellite with competitive learning strategy," *Aerospace Science and Technology*, vol. 118, 2021, p. 107047.

[18] Y. He, L. Xing, Y. Chen, W. Pedrycz, L. Wang, and G. Wu, "A generic Markov decision process model and reinforcement learning method for scheduling agile earth observation satellites," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 3, 2020, pp. 1463–1474.

[19] Y. Huang, Z. Mu, S. Wu, B. Cui, and Y. Duan, "Revising the observation satellite scheduling problem based on deep reinforcement learning," *Remote Sensing*, vol. 13, no. 12, 2021, p. 2377.

[20] G. Kim and J. Park, "Visual-Language Model Technology in Robot Manipulation," *Magazine of Institute of Control, Robotics and Systems (in Korean)*, vol. 30, no. 1, pp. 38–44, 2024.

[21] W.-S. Lee, T.-Y. Kim, and W.-S. Choi, "Autonomous navigation command generation system for underwater robots using LLM," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 30, no. 10, pp.1191–1196, 2024.

[22] E. Kasneci, K. Sessler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, S. Krusche, G. Kutyniok, T. Michaeli, C. Nerdel, J. Pfeffer, O. Poquet, M. Sailer, A. Schmidt, T. Seidel, M. Stadler, J. Weller, J. Kuhn, and G. Kasneci, "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, 2023, p. 102274.

[23] L. Floridi and M. Chiriatti, "GPT-3: Its nature, scope, limits, and consequences," *Minds and Machines*, vol. 30, pp. 681–694, 2020.

[24] D.-S. Jang, D.-H. Cho, W.-C. Lee, S.-K. Ryu, B. Jeong, M. Hong, M. Jung, M. Kim, M. Lee, S. Lee, and H.-L. Choi, "Unlocking robotic autonomy: a survey on the applications of foundation models," *International Journal of Control, Automation and Systems*, vol. 22, no. 8, pp. 2341–2384, 2024.

[25] J. Ruan, Y. Chen, B. Zhang, Z. Xu, T. Bao, D.G. Qing, S. Shiwei, H. Mao, Z. Li, X. Zeng, and R. Zhao, "TPTU: Task planning and tool usage of large language model-based AI agents," *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

[26] Z. Wang, S. Cai, G. Chen, A. Liu, X. Ma, and Y. Liang, "Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents," arXiv preprint arXiv:2302.01560, 2023.

[27] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 11523–11530.

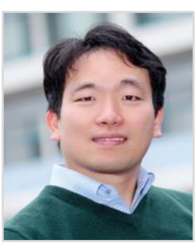
[28] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," *Proc. of the IEEE/CVF International Conference on Computer Vision, 2023*, pp. 2998–3009.

**Sung Jun Kim**

received his B.S. degree in aerospace engineering from the Pennsylvania State University, University Park, PA, USA, in 2018 and his M.S. degree in aerospace engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2020. He is currently a Ph.D. candidate in aerospace engineering at KAIST. His current research interests include satellite scheduling and decision-making under astrodynamics.

**Minchae Kim**

received her B.S. degree in aerospace engineering from Inha University, Incheon, South Korea, in 2023 and her M.S. degree in aerospace engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2025. She is currently a Ph.D. candidate in aerospace engineering at KAIST. Her research interests include decision-making under uncertainty, astrodynamics and spacecraft autonomous control.



**Han-Lim Choi**

received the B.S. and M.S. degrees in aerospace engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2000 and 2002, respectively, and the Ph.D. degree in aeronautics and astronautics from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2009. Then, he studied at MIT as a Postdoctoral Associate until he joined KAIST, in 2010. He is currently a Professor of aerospace engineering at KAIST. His current research interests include estimation and control for sensor networks and decision making for multiagent systems.