The Journal of the Astronautical Sciences (2024) 71:34
https://doi.org/10.1007/s40295-024-00455-6

ORIGINAL ARTICLE

Optimal Sensor Tasking for Space Domain Awareness Via a Beam A*-Search Algorithm

Lorenzo Federici[1] · Andrea D'Ambrosio[1] · Roberto Furfaro[1,2] · Vishnu Reddy[3]

Accepted: 29 July 2024 / Published online: 13 August 2024

Abstract

Sensor tasking for space domain awareness is a complex problem that involves scheduling observations of objects in space from one or multiple sensors, usually telescopes. This paper formulates the short-term sensor tasking problem from a single telescope as a purely combinatorial problem and uses a beam version of the A* search algorithm to efficiently search for priority-based observation schedules. Several admissible heuristics are proposed to speed up the search procedure when aiming for sub-GEO, GEO, or both kinds of objects in the same observation window. Several pruning techniques are also implemented to reduce the search space dimension and the computational time required. Simulations are conducted with real space objects retrieved from the Norad catalog and data from one of the telescopes of the Space4 Center at the University of Arizona. The objective is to evaluate the search algorithm's effectiveness, time complexity, and performance under different real-world scenarios.

Keywords Sensor tasking · A* search · Beam search · Space domain awareness

✉ Lorenzo Federici
lorenzof@arizona.edu

Andrea D'Ambrosio
dambrosio@arizona.edu

Roberto Furfaro
robertof@arizona.edu

Vishnu Reddy
vishnureddy@arizona.edu

[1] Department of Systems & Industrial Engineering, The University of Arizona, 1127 E James E. Rogers Way, Tucson, AZ 85721, USA

[2] Department of Aerospace and Mechanical Engineering, The University of Arizona, 1130 N Mountain Ave., Tucson, AZ 85721, USA

[3] Lunar and Planetary Laboratory, The University of Arizona, 1629 E University Blvd, Tucson, AZ 85721, USA

1 Introduction

Scheduling observations of human-made objects in space for space domain awareness (SDA) involves coordinating multiple ground-based sensors, like telescopes or antennas, to collect data and use them for several purposes, among which orbit determination, object catalog maintenance, and conjunction analysis. The main objective of sensor tasking is to maximize either the quantity of tracked objects within a specific time frame or an accumulated score. Indeed, each object can be assigned a priority index or score to indicate its relative significance. For instance, a suspected conjunction may result in a higher priority for the involved objects in the preceding nights, to collect as many observations as possible and reconstruct their orbits with higher accuracy. The process of tasking must consider various factors, including the telescope's geographical location, the visibility of the objects, the telescope's range of motion and speed, potential geographical limitations, lighting conditions, and the time needed for camera preparation, focusing, and executing the required exposures with the desired filters.

Achieving the best possible sensor tasking is a challenging optimization problem that generally can be classified as an NP-hard combinatorial problem. This involves determining the optimal set of objects to be observed from a larger pool and establishing the order of observation while satisfying some physical and time constraints. Practical implementation of precise integer linear programming (ILP) solution methods, like branch and bound, often becomes unfeasible due to their excessively demanding computational requirements. Numerous approaches to sensor tasking have been proposed within the existing literature. Among these, a prevalent approach entails employing either heuristic algorithms or greedy solution mechanisms [1], which operate based on predefined rules and decision trees to rapidly identify a generally suboptimal solution. To address long-term sensor planning tasks, some papers have explored dynamic programming techniques [2] and Monte Carlo-based tree search algorithms [3], although assessing all the potential observations still demands substantial computational resources. Furthermore, these methods tend to be confined to a single orbital regime, such as objects in geosynchronous Earth orbit (GEO), low Earth orbit (LEO), or medium Earth orbit (MEO). More recently, machine learning methods, such as deep reinforcement learning [4, 5], have also been adopted in sensor tasking problems, exhibiting good levels of accuracy and adaptability when facing changes in object orbits, observation windows, observer locations, and sensor characteristics. These approaches can learn from past instances of the problem and adapt to novel data. However, they cannot exploit the underlying problem's inherent mathematical structure, frequently resulting in long training periods and an absence of guarantees regarding the optimality or robustness of the final solution.

This paper focuses on short-term single-telescope tasking for observing objects in multiple orbital regimes, specifically, in sub-GEO (LEO, MEO, GTO) and GEO. The main objective is to develop a rapid tasking tool capable of scheduling telescope observations on a night-by-night basis. This involves retrieving the most current two-line elements (TLEs) of the target objects directly from the Norad catalog via SpaceTrack.com just before the beginning of each observation session, typically shortly before local nautical twilight. Given that Norad's TLEs are updated daily, the position of the objects in the sky at the time of observation can be determined by forward-propagating them for less than a day. This ensures that any errors introduced during propagation are small enough to maintain the object within the telescope's field of view and, consequently, within the exposures captured by the camera. Consequently, uncertainties in the object's position can be safely disregarded.

This deterministic sensor tasking problem can be formulated as a variant of the orienteering problem (OP), an NP-hard combinatorial problem that combines the score-maximization objective of the knapsack problem (KP) with the path-length minimization elements of the traveling salesman problem (TSP). This problem takes its name from an outdoor sport, orienteering, where the objective is to visit, in a limited amount of time, a subset of the checkpoints located on a map, starting from the home base, to maximize the total score associated with them. In the telescope tasking scenario, each checkpoint corresponds to a different observation opportunity for a given object, and the time to move from one checkpoint to the next one (i.e., the cost associated with that arc) corresponds to the time needed to physically slew the telescope to the new pointing direction, possibly wait till the beginning of the new object pass, or for a fixed preparation time, and observe the object itself to collect the desired exposures.

In this work, the sensor tasking problem is formulated as a search problem on a graph and tackled with the A* search algorithm [6]. A* search is a widely used algorithm for finding the optimal path in a graph, and it can be applied to the optimal telescope tasking to efficiently search through the vast number of possible observation schedules. The A* algorithm is a combination of two approaches: Dijkstra's algorithm for finding the shortest path (respectively, the highest scoring path, in maximization problems) in a graph and a heuristic function that estimates the "distance" (respectively, the improvement in total score) between a node and the goal. The algorithm uses a priority queue to keep track of nodes that have been visited and those that need to be explored. Each node is assigned an evaluation function based on the sum between its distance from the start node (respectively, the cumulative score since the start node) and the estimated distance (respectively, score improvement) to the goal. The algorithm selects the node with the best value of the evaluation function and explores its neighbors, updating the heuristic associated with neighboring nodes as necessary. The search continues until the goal node is reached or until no more nodes can be explored. The efficiency of A* search mainly depends on the accuracy of the heuristic function. The more accurate the heuristic, the fewer nodes are expanded, but, generally, the more complex the heuristic is to be computed. Hence, a trade-off between the accuracy of the heuristics and its complexity is usually required. If the heuristic is admissible, that is, it never overestimates (respectively, underestimates) the true cost (respectively, cumulative score) of the optimal path to the goal node, A* is proven to be both optimal and optimally efficient. Three admissible heuristics for A* are proposed in this study, the first valid for scheduling sub-GEO (i.e., LEO and MEO) objects, the second for scheduling GEO objects, and the third one combining the other two when objects in both orbital regimes are considered.

Being an exact solution algorithm, A* may encounter challenges in finding an optimal solution within a reasonable computational time, particularly when dealing with complex problem instances. To address this issue, various pruning techniques are employed during the search process itself. Specifically, schedules deemed unlikely to yield a high cumulative score are systematically eliminated from the priority queue. This elimination is achieved by implementing a beam search version of the A* algorithm [7], named Beam-A* (BA*). While this pruning strategy significantly accelerates the search, it comes at the trade-off of losing guarantees on the global optimality of the solution.

Numerical simulations are carried out using actual object data in both LEO and GEO retrieved from the Norad catalog and the real characteristics of one of the telescopes (Raptors-2) of the Space4 Center at The University of Arizona (Tucson, Arizona). First, the two A* heuristics for sub-GEO and GEO objects are compared with a "base" heuristic in terms of the number of expanded nodes and run time on increasingly bigger sets of objects to understand their efficiency and time complexity, using an optimal version of A*. The effect of varying the object score range on the heuristic efficiency and on the optimal observation schedules is also analyzed in detail. The beam search version of the algorithm, BA*, is then compared against the exact version of the A* to understand its accuracy and time complexity when dealing with different sets of objects in both the GEO and sub-GEO regimes and longer observation times. Eventually, the results obtained when scheduling objects in just the LEO, GEO, or both orbital regimes at the same time are presented and discussed in detail by varying the available observation time.

2 Problem Statement

This section introduces the telescope tasking problem (TTP) and mathematically formulates it as a search problem on a graph.

2.1 Problem Definition

Let us consider a telescope characterized by its geographic coordinates (λ, ϕ, z), which represent its latitude, longitude, and altitude above sea level. This telescope can be oriented in a specific direction identified by the azimuth ψ and altitude (or elevation) θ, by using a motorized altazimuth mount. This mount has a maximum angular rate of motion, or slew rate, denoted as $\omega_{\max}$.

On a given day, the telescope becomes operational at a specific time called epoch t_0. This time is chosen to coincide with nautical twilight, defined as the time when the Sun's elevation relative to the local horizon drops below $-12°$. The telescope remains active for a total time of $\Delta t_{\max}$, typically extending throughout the entire night until the subsequent nautical twilight when the Sun's elevation rises above $-12°$ again. Once the telescope is initiated at epoch t_0, it promptly goes to its designated “home” position marked as H. This home position is characterized by a specific azimuth ψ_H and altitude θ_H.

Throughout the night, from time t_0 to time $t_0 + \Delta t_{\max}$, the telescope is tasked with conducting a series of optical observations of artificial objects in Earth’s orbit using a camera affixed to it. To be more specific, a collection $\boldsymbol{l} = l_1, \dots, l_L$ of L known objects positioned in sub-GEO (LEO / MEO) and another collection $\boldsymbol{g} = g_1, \dots, g_G$ of G known objects in GEO are intended to be scheduled for observation during the aforementioned night. Each of these objects $d_i \in \boldsymbol{l} \cup \boldsymbol{g}$ is supposed to be observable at least once from the telescope during that night, and it is associated with a designated count of exposures, denoted as E_{d_i}, to be realized with the camera, each with the same total duration $\Delta t^{\text{exp}}_{d_i}$.

Each object in the sub-GEO category, denoted as $l_i \in \boldsymbol{l}$, undergoes P_{l_i} passes across the telescope’s field of view within the observation time $\Delta t_{\max}$. These passes are designated by the indices $p = 1, \dots, P_{l_i}$, and each starts at time $t^s_{l_i,p}$, reaches the maximum elevation point at time $t^h_{l_i,p}$, and ends at time $t^e_{l_i,p}$. The starting and ending epochs mark the times when the object reaches its minimum elevation angle $\theta_{\min}$ for visibility above the local horizon. The positions of the object l_i at these times are denoted as $(\theta^s_{l_i,p}, \psi^s_{l_i,p})$, $(\theta^h_{l_i,p}, \psi^h_{l_i,p})$, and $(\theta^e_{l_i,p}, \psi^e_{l_i,p})$, respectively. Should the sub-GEO object $l_i \in \boldsymbol{l}$ be observed during its p-th pass, $p = 1, \dots, P_{l_i}$, the telescope will continuously track the object during an arc of his trajectory in the sky centered about the highest elevation point and with duration

$$\Delta t^{\text{obs}}_{l_i,p} = \max \left\{ E_{l_i} \Delta t^{\text{exp}}_{l_i}, t^e_{l_i,p} - t^s_{l_i,p} \right\} \tag{1}$$

The initial and final time of the observed arc are then

$$t^{s,\text{obs}}_{l_i,p} = t^h_{l_i,p} - 0.5 \Delta t^{\text{obs}}_{l_i,p} \tag{2}$$

$$t^{e,\text{obs}}_{l_i,p} = t^h_{l_i,p} + 0.5 \Delta t^{\text{obs}}_{l_i,p} \tag{3}$$

The corresponding position of the spacecraft, i.e., the pointing direction of the telescope, can be obtained starting from the azimuth and elevation rates of the spacecraft

$$\theta^{s,\text{obs}}_{l_i,p} = \theta^h_{l_i,p} - 0.5\, \dot{\theta}^{s,h}_{l_i,p} \Delta t^{\text{obs}}_{l_i,p} \tag{4}$$

$$\psi^{s,\text{obs}}_{l_i,p} = \psi^h_{l_i,p} - 0.5\, \dot{\psi}^{s,h}_{l_i,p} \Delta t^{\text{obs}}_{l_i,p} \tag{5}$$

$$\theta^{e,\text{obs}}_{l_i,p} = \theta^h_{l_i,p} + 0.5\, \dot{\theta}^{h,e}_{l_i,p} \Delta t^{\text{obs}}_{l_i,p} \tag{6}$$

$$\psi^{e,\text{obs}}_{l_i,p} = \psi^{h}_{l_i,p} + 0.5\,\dot{\psi}^{h,e}_{l_i,p}\Delta t^{\text{obs}}_{l_i,p} \tag{7}$$

Those rates, in turn, can be estimated for the two halves of the observed arc by knowing the initial, highest elevation point, and final object position along the p-th pass

$$\dot{\theta}^{s,h}_{l_i,p} = 2\frac{\theta^{h}_{l_i,p} - \theta^{s}_{l_i,p}}{\Delta t^{\text{obs}}_{l_i,p}} \tag{8}$$

$$\dot{\psi}^{s,h}_{l_i,p} = 2\frac{\psi^{h}_{l_i,p} - \psi^{s}_{l_i,p}}{\Delta t^{\text{obs}}_{l_i,p}} \tag{9}$$

$$\dot{\theta}^{h,e}_{l_i,p} = 2\frac{\theta^{e}_{l_i,p} - \theta^{h}_{l_i,p}}{\Delta t^{\text{obs}}_{l_i,p}} \tag{10}$$

$$\dot{\psi}^{h,e}_{l_i,p} = 2\frac{\psi^{e}_{l_i,p} - \psi^{h}_{l_i,p}}{\Delta t^{\text{obs}}_{l_i,p}} \tag{11}$$

The score s_{l_i} linked to the sub-GEO object l_i is collected when the object is observed during any of its P_{l_i} passes.

On the other hand, the GEO objects are regarded as observable at all times throughout the night due to their fixed position in the sky $(\theta_{g_i}, \psi_{g_i})$ relative to the observatory site. Therefore, a GEO object has the potential to be observed on multiple occasions during the night, with each observation lasting a fraction of the remaining total exposure time. If a GEO object $g_i \in \boldsymbol{g}$ is observed for an integer fraction p of the total exposures E_{g_i}, $p = 1, \dots, E_{g_i}$, the duration of the observation and the resultant collected score will be as follows

$$\Delta t^{\text{obs}}_{g_i,p} = p\Delta t^{\text{exp}}_{g_i} \tag{12}$$

$$s_{g_i,p} = p\frac{s_{g_i}}{E_{g_i}} \tag{13}$$

The time required to slew the telescope from the last position of the object d_i during the observation arc along its j-th sky pass to the initial position of the next object d_k during the observation arc of its l-th sky pass amounts to

$$\Delta t^{\text{slew}}_{d_i,j,d_k,l} = \frac{\cos^{-1}\left(\boldsymbol{r}^{s,\text{obs}}_{d_i,j} \cdot \boldsymbol{r}^{e,\text{obs}}_{d_k,l}\right)}{\omega_{\max}} \tag{14}$$

where

$$\boldsymbol{r}^{a,\text{obs}}_{d_i,j} = \begin{cases} [\cos\theta^{a,\text{obs}}_{d_i,j}\cos\psi^{a,\text{obs}}_{d_i,j},\ \cos\theta^{a,\text{obs}}_{d_i,j}\sin\psi^{a,\text{obs}}_{d_i,j},\ \sin\theta^{a,\text{obs}}_{d_i,j}]^\top & d_i \in \boldsymbol{l} \\ [\cos\theta_{d_i}\cos\psi_{d_i},\ \cos\theta_{d_i}\sin\psi_{d_i},\ \sin\theta_{d_i}]^\top & d_i \in \boldsymbol{g} \end{cases} \tag{15}$$

with $a = s, e$.

Before starting to actually take exposures of an object, the telescope must stay in place at the initial position of that object in the sky for a preparation time equal to $\Delta t^{\text{prep}}_{\text{sgeo}}$, for sub-GEO objects, and $\Delta t^{\text{prep}}_{\text{geo}}$, for GEO objects. This preparation time is necessary to focus the camera, change the filter, and propagate the object's TLEs to correctly lock the telescope onto its orbit.

2.2 Search Problem

The TTP can be formulated as a search problem within a graph. A search problem encompasses a set of potential states, an initial state, a successor or transition function that maps from any given state to a collection of new states (referred to as successors), a goal test that determines whether a state qualifies as a goal state, that is, a final state within the search, and an objective function linked to either the states themselves or the path traversed within the graph. The goal of a search algorithm is to determine a sequence of transitions that initiates from the initial state of the problem, culminates in a goal state, and maximizes the value of the objective function.

By effectively defining the state space, initial state, successor function, goal test, and objective function, the TTP can be formulated as a search problem. Specifically, each state $\boldsymbol{\sigma}$ is a sequence of pairs (d_i, p_i), each giving the label of the object observed d_i, and either the pass number p_i (for sub-GEO objects) or the number of exposures taken p_i (for GEO objects):

$$\boldsymbol{\sigma} = \{(d_1, p_1), \dots, (d_N, p_N)\} \tag{16}$$

Each observation (d_i, p_i) in $\boldsymbol{\sigma}$, $i = 1, \dots, N$, can be associated with a starting and ending time $t^s_{\sigma,i}$, $t^e_{\sigma,i}$:

$$t^s_{\sigma,i} = \begin{cases} t^{s,\text{obs}}_{d_i,p_i} & d_i \in \boldsymbol{l} \\ t^e_{\sigma,i-1} + \Delta t^{\text{slew}}_{d_{i-1},p_{i-1},d_i,p_i} + \Delta t^{\text{prep}}_{\text{geo}} & d_i \in \boldsymbol{g} \end{cases} \tag{17}$$

$$t^e_{\sigma,i} = \begin{cases} t^{e,\text{obs}}_{d_i,p_i} & d_i \in \boldsymbol{l} \\ t^s_{\sigma,i} + \Delta t^{\text{obs}}_{d_i,p_i} & d_i \in \boldsymbol{g} \end{cases} \tag{18}$$

and with a corresponding score:

$$s_{\sigma,i} = \begin{cases} s_{d_i} & d_i \in \boldsymbol{l} \\ s_{d_i,p_i} & d_i \in \boldsymbol{g} \end{cases} \tag{19}$$

The total score of a state $\boldsymbol{\sigma}$ is

$$s_{\boldsymbol{\sigma}} = \sum_{i=1}^{N} s_{\boldsymbol{\sigma},i} \tag{20}$$

while the total exposure time $\Delta t_{\text{obs},\boldsymbol{\sigma}}$ (i.e., total time spent taking exposures), total slewing time $\Delta t_{\text{slew},\boldsymbol{\sigma}}$ (i.e., total time spent moving the telescope) and total observation time $\Delta t_{\text{tot},\boldsymbol{\sigma}}$ (i.e., total duration of the observation session) are

$$\Delta t_{\text{obs},\boldsymbol{\sigma}} = \sum_{i=1}^{N} \Delta t^{\text{obs}}_{d_i,p_i} \tag{21}$$

$$\Delta t^{\text{slew}}_{\text{tot},\boldsymbol{\sigma}} = \sum_{i=1}^{N} \Delta t^{\text{slew}}_{d_{i-1},p_{i-1},d_i,p_i} \tag{22}$$

$$\Delta t_{\text{tot},\boldsymbol{\sigma}} = t^{e}_{\boldsymbol{\sigma},N} - t_0 \tag{23}$$

Let us define with $\boldsymbol{d}_{\boldsymbol{\sigma}} = \{d_i\}|_{i=1,\dots,N}$ the set of objects in $\boldsymbol{\sigma}$. For each object $d_i \in \boldsymbol{d}_{\boldsymbol{\sigma}}$, the set $\boldsymbol{k}_{\boldsymbol{\sigma},i}$ is the set of indices in $\{1,\dots,N\} \setminus i$ that refer to observations of the same object d_i:

$$\boldsymbol{k}_{\boldsymbol{\sigma},i} = \{k \in \{1,\dots,N\} \setminus i \, : \, d_k = d_i \in \boldsymbol{d}_{\boldsymbol{\sigma}}\} \tag{24}$$

State $\boldsymbol{\sigma}$ is an admissible state if the following conditions are satisfied

$$d_i \in \boldsymbol{l} \cup \boldsymbol{g}, \quad i = 1,\dots,N \tag{25}$$

$$p_i \in \mathbb{N}^+, \quad i = 1,\dots,N \tag{26}$$

$$d_i \neq d_{i+1}, \quad i = 1,\dots,N-1 \tag{27}$$

$$\boldsymbol{k}_{\boldsymbol{\sigma},i} = \emptyset, \quad d_i \in \boldsymbol{l}, i = 1,\dots,N \tag{28}$$

$$p_i \leq \begin{cases} P_{d_i} & d_i \in \boldsymbol{l} \\ E_{d_i} - \sum\limits_{\substack{k \in \boldsymbol{k}_{\boldsymbol{\sigma},i} \\ k < i}} p_k & d_i \in \boldsymbol{g} \end{cases}, \quad i = 1,\dots,N \tag{29}$$

$$t^{s}_{\boldsymbol{\sigma},i} \geq t^{e}_{\boldsymbol{\sigma},i-1} + \Delta t^{\text{slew}}_{d_{i-1},p_{i-1},d_i,p_i} + \Delta t^{\text{prep}}_{\text{sgeo}}, \quad d_i \in \boldsymbol{l}, i = 2,\dots,N \tag{30}$$

$$\Delta t_{\text{tot},\boldsymbol{\sigma}} \leq \Delta t_{\max} \tag{31}$$

The number of different objects observed in state $\boldsymbol{\sigma}$ is

$$N_{\mathrm{obj},\boldsymbol{\sigma}} = N - \sum_{i=1}^{N} \frac{|\boldsymbol{k}_{\boldsymbol{\sigma},i}|}{|\boldsymbol{k}_{\boldsymbol{\sigma},i}| + 1} \tag{32}$$

So, the state space of the search problem is:

$$\mathcal{S} = \{\boldsymbol{\sigma} \,:\, (16)\text{–}(32)\} \tag{33}$$

The initial state of the search is the state $\boldsymbol{\sigma}_0$ containing just the “fake” observation $(H, 1)$, that is, the telescope at its home position at the initial epoch t_0:

$$\boldsymbol{\sigma}_0 = \{(H, 1)\} \tag{34}$$

$$t^s_{\boldsymbol{\sigma}_0} = t_0 \tag{35}$$

$$t^e_{\boldsymbol{\sigma}_0} = t_0 \tag{36}$$

$$s_{\boldsymbol{\sigma}_0} = 0 \tag{37}$$

Any successor state $\boldsymbol{\sigma}'$ of state $\boldsymbol{\sigma}$ is obtained by adding to the sequence a new observation $\left(d_{N+1}, p_{N+1}\right)$. The pair $\left(d_{N+1}, p_{N+1}\right)$ has to be selected so that $\boldsymbol{\sigma}'$ is still an admissible state, i.e., it belongs to set $\mathcal{S}$. Hence, the set of successor states of the state $\boldsymbol{\sigma}$ is:

$$\mathcal{C}(\boldsymbol{\sigma}) = \left\{\boldsymbol{\sigma}' \,:\, \boldsymbol{\sigma}' = \boldsymbol{\sigma} \cup \left(d_{N+1}, p_{N+1}\right),\, \boldsymbol{\sigma}' \in \mathcal{S}\right\} \tag{38}$$

Let us introduce the set of unscheduled observations associated with a state $\boldsymbol{\sigma}$:

$$\boldsymbol{u}_{\boldsymbol{\sigma}} = \{(\tilde{d}_1, \tilde{p}_1), \dots, (\tilde{d}_U, \tilde{p}_U)\} \tag{39}$$

This set contains all the pairs $(\tilde{d}_i, \tilde{p}_i)$ meeting the following conditions:

$$\tilde{d}_i \in \boldsymbol{l} \cup \boldsymbol{g}, \quad i = 1, \dots, U \tag{40}$$

$$\tilde{d}_i \notin \boldsymbol{l} \cap \boldsymbol{d}_{\boldsymbol{\sigma}}, \quad i = 1, \dots, U \tag{41}$$

$$\tilde{d}_i \notin \{d_j \in \boldsymbol{g} \cap \boldsymbol{d}_{\boldsymbol{\sigma}} \,:\, p_{\max\{\boldsymbol{k}_{\boldsymbol{\sigma},j}\}} = E_{d_j}\}, \quad i = 1, \dots, U \tag{42}$$

$$\tilde{p}_i \in \mathbb{N}^+, \quad i = 1, \dots, U \tag{43}$$

$$\tilde{p}_i \le P_{\tilde{d}_i}, \quad \tilde{d}_i \in \boldsymbol{l},\, i = 1, \dots, U \tag{44}$$

$$\tilde{p}_i \ne \tilde{p}_k, \quad \tilde{d}_i = \tilde{d}_k \in \boldsymbol{l},\, i \ne k = 1, \dots, U \tag{45}$$

$$\tilde{p}_i = \begin{cases} E_{\tilde{d}_i} & \tilde{d}_i \in \boldsymbol{g} \setminus (\boldsymbol{g} \cap \boldsymbol{d}_{\boldsymbol{\sigma}}) \\ E_{\tilde{d}_i} - \sum\limits_{k \in \boldsymbol{k}_{\boldsymbol{\sigma},i}} p_k & \tilde{d}_i \in \boldsymbol{g} \cap \boldsymbol{d}_{\boldsymbol{\sigma}} \end{cases}, \quad i = 1, \dots, U \tag{46}$$

$$t^{s,\mathrm{obs}}_{\tilde{d}_i,\tilde{p}_i} \geq t^{e}_{\boldsymbol{\sigma},N} + \Delta t^{\mathrm{slew}}_{d_N,p_N,\tilde{d}_i,\tilde{p}_i} + \Delta t^{\mathrm{prep}}_{\mathrm{sgeo}}, \quad \tilde{d}_i \in \boldsymbol{l},\ i = 1, \dots, U \tag{47}$$

The score associated with each unscheduled object $\tilde{d}_i$, $i = 1, \dots, U$, is

$$s_{\boldsymbol{u}_{\boldsymbol{\sigma}},i} = \begin{cases} s_{\tilde{d}_i} & \tilde{d}_i \in \boldsymbol{l} \\ s_{\tilde{d}_i,\tilde{p}_i} & \tilde{d}_i \in \boldsymbol{g} \end{cases} \tag{48}$$

The total observation time left associated with state $\boldsymbol{\sigma}$ is

$$\Delta t_{\mathrm{left},\boldsymbol{\sigma}} = \Delta t_{\max} - \Delta t_{\mathrm{tot},\boldsymbol{\sigma}} \tag{49}$$

A state $\boldsymbol{\sigma}$ is considered a goal state if the set of unscheduled object referred to $\boldsymbol{\sigma}$ is empty:

$$\boldsymbol{u}_{\boldsymbol{\sigma}} = \emptyset \tag{50}$$

This situation arises when either all the planned objects have already been observed for their total exposure time or when there is no possibility of observing any additional object without exceeding the maximum observation time.

Hence, the set of goal states of the search problem is

$$\mathcal{G} = \{\boldsymbol{\sigma} : \boldsymbol{\sigma} \in \mathcal{S}, (39)\text{–}(48), (50)\} \tag{51}$$

The objective function of the problem is just a function of the last state $\boldsymbol{\sigma}$ reached by the search process and is defined as

$$J_{\boldsymbol{\sigma}} = 10^3 s_{\boldsymbol{\sigma}} - \Delta t_{\mathrm{tot},\boldsymbol{\sigma}} + 10^{-6} N_{\mathrm{obj},\boldsymbol{\sigma}} \tag{52}$$

This objective function has been devised to prioritize solutions based on total score first, followed by total observation time in case of ties, and lastly by the number of observed objects. The three scaling factors 10^3, 1, and 10^{-6} have been specifically selected to prevent the components from mixing and to keep their contributions separate. Indeed, given the typical application scenarios, the ranges of the score $s_{\boldsymbol{\sigma}}$, observation time $\Delta t_{\mathrm{tot},\boldsymbol{\sigma}}$, and number of objects $N_{\mathrm{obj},\boldsymbol{\sigma}}$ will be approximately (0.01–300), (10^{-5}–0.35), and (1–100), respectively. By multiplying them by the scaling factors in Eq. (52), their ranges become (10–300,000), (10^{-4}–0.35), and (10^{-6}–10^{-4}), respectively. This ensures that the ranges do not overlap; thus, a solution with a lower observation time but also a lower score, or a solution with the same score, a higher number of objects but a lower observation time, will not have a higher merit index.

A goal state is said to be optimal if no other goal state in $\mathcal{G}$ has a higher objective function value. The goal of a search algorithm is to find a sequence of transitions that, starting from the initial state of the problem, reaches an optimal goal state.

If $\boldsymbol{\Sigma} = \left\{\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \dots, \boldsymbol{\sigma}_K\right\}$ represents the sequence of visited states along the optimal path, then the search problem can be stated as follows:

$$\begin{aligned} &\max_{\boldsymbol{\Sigma}} J_{\sigma_K} \\ \text{s.t.}\quad &\boldsymbol{\sigma}_i \in \mathcal{S}, && i = 1, \dots, K \\ &\boldsymbol{\sigma}_K \in \mathcal{G}, \\ &\boldsymbol{\sigma}_i \in \mathcal{C}(\boldsymbol{\sigma}_{i-1}), && i = 1, \dots, K \end{aligned} \tag{53}$$

3 Solution Approach

This section outlines the beam A*-search technique used to address the search problem in Eq. (53). Diverse heuristics are also presented to accelerate the search process when confronted with either just sub-GEO objects, GEO objects, or a combination of both types of objects.

3.1 A* Search

A* is a widely employed algorithm in the realm of artificial intelligence (AI), first developed during the Shakey project at the Stanford Research Institute in 1968 [6]. Presently, A* finds applications in various AI domains, such as parsing in natural language processing [8], path planning for robots and UAVs [9], and pathfinding in video games [10].

The A* algorithm solves search problems by constructing a search tree. Each node of the tree corresponds to a problem state, and the connections between nodes represent state-to-state transitions. The search procedure starts by generating all the successor nodes of the root node, which corresponds to the initial problem state $\boldsymbol{\sigma}_0$. These newly generated states become leaf nodes and are added to the frontier, a prioritized collection of all leaf nodes at a specific point of the search process. During each iteration, the algorithm removes the topmost leaf node from the frontier and evaluates it against the goal test. If the node's state matches a goal state, the search is concluded, and that state is returned as the problem's solution. If not, the node is expanded by generating successors based on its state. These new leaf nodes are appended to the frontier, initiating a new iteration. The search is terminated either when a goal node is reached or when no further nodes can be generated. In this last case, the last expanded node is returned as the problem solution.

In A*, the frontier is organized as a priority queue using an evaluation function f_σ combining the state's objective function J_σ with a heuristic function h_σ as the ordering criterion. In problems aiming to maximize a state-dependent objective function, like the one examined in this paper, the heuristic function h_σ estimates the potential improvement in the objective function obtained by reaching the nearest goal node from state $\boldsymbol{\sigma}$. Hence, during each iteration, the algorithm selects the node with the highest evaluation function for expansion. If the heuristic function is admissible, meaning it never underestimates the actual objective function improvement towards the closest goal node, A* ensures both optimality and optimal efficiency. Optimality means that the first expanded goal node is the global optimal solution, while optimal efficiency implies that no other search algorithm is guaranteed to explore fewer nodes than A* for the same problem. The admissibility condition can be formulated as follows, by supposing that $\boldsymbol{\sigma}_g$ is the nearest reachable goal node from $\boldsymbol{\sigma}$:

$$f_{\sigma} = J_{\sigma} + h_{\sigma} \geq J_{\sigma_g} \tag{54}$$

Being an optimal algorithm, when dealing with NP-hard problems, A*'s time and space complexity tend to grow exponentially with the problem dimension. This may make A* impractical for handling large-scale problems. However, this limitation can be alleviated by applying well-designed heuristics, which reduce the number of nodes that need to be expanded to find the optimal solution. Indeed, the more accurate the heuristic, that is, the closer its value to the actual merit index improvement to the closest goal node, the fewer nodes are expanded during the search. If the heuristic were perfect, no search iterations would be necessary, as A* would consistently choose the subsequent state in the solution of the problem $\boldsymbol{\Sigma}$ for expansion. Indeed, knowing a perfect heuristic implies knowing the problem solution itself. Hence, a trade-off emerges between the accuracy of the heuristic and its computational demands, which encompass considerations both on memory utilization and speed.

3.2 Heuristics

In this paper, three admissible heuristics are introduced. These heuristics are exact solutions derived from relaxed versions of the TTP when focusing exclusively on either sub-GEO or GEO objects or both kinds of objects. The following sections delve into the presentation of these heuristics.

3.2.1 Base Heuristic

To establish a benchmark for performance assessment of the heuristics that will be presented afterward, a "base" heuristic is introduced. The base heuristic is derived by straightforwardly summing the number of remaining observations and the scores and the overall observation and preparation time of all unscheduled objects. In cases where the cumulative observation time of unscheduled objects surpasses the remaining observation time, the latter is used in the heuristic. Thus,

$$\begin{aligned} h_{\boldsymbol{\sigma}}^{\text{base}} =& 10^3 \sum_{i=1}^{U} s_{\boldsymbol{u}_{\sigma},i} - \min\left\{ \sum_{i=1}^{U} \Delta t_{\tilde{d}_i,\tilde{p}_i}^{\text{obs}} + \sum_{\tilde{d}_i \in \boldsymbol{l}} \Delta t_{\text{sgeo}}^{\text{prep}} + \sum_{\tilde{d}_i \in \boldsymbol{g}} \Delta t_{\text{geo}}^{\text{prep}},\ \Delta t_{\text{left},\boldsymbol{\sigma}} \right\} + \\ & + 10^{-6} U \end{aligned} \tag{55}$$

This heuristic is admissible, as it assumes that all unscheduled objects are observed for their complete exposure time within the remaining observation time.

Additionally, it assumes that there are no waiting times or slewing times between successive observations. As such, the heuristic both overestimates the total score improvement and number of objects observed and underestimates the time required to execute the pending observations, thus producing a value of the objective function always higher than the one achievable from state $\boldsymbol{\sigma}$.

3.2.2 Sub-GEO Heuristic

In situations where just sub-GEO objects are still unscheduled ($\tilde{d}_i \in \boldsymbol{l}, i = 1, \dots, U$), the TTP transforms into a pure orienteering problem (OP) [11]. The OP is built around a weighted graph consisting of n nodes, labeled from 1 to n, each associated with a score s_i, linked by arcs with weights w_{ij} ($i,j = 1, \dots, n$). A designated root node $r \in \{1, \dots, n\}$ is also part of the graph. The primary objective of the OP is determining the optimal path, initiating from r, that maximizes the cumulative score of the visited nodes, while maintaining a total weight that does not exceed a predefined maximum value $w_{\max}$. In the specific variant of TTP focused solely on sub-GEO objects, each node within the graph symbolizes a distinct object-pass combination among the unscheduled objects ($i = (\tilde{d}_i, \tilde{p}_i)$, $i = 1, \dots, U$), along with the last observation in the state $\boldsymbol{\sigma}$, which corresponds to the root node ($r = (d_N, p_N)$). Consequently, if observation j can indeed follow observation i, an arc is established in the graph, connecting node i to node j. This condition is met if:

$$t^{s,\mathrm{obs}}_{\tilde{d}_j,\tilde{p}_j} \geq t^{e,\mathrm{obs}}_{\tilde{d}_i,\tilde{p}_i} + \Delta t^{\mathrm{slew}}_{\tilde{d}_i,\tilde{p}_i,\tilde{d}_j,\tilde{p}_j} + \Delta t^{\mathrm{prep}}_{\mathrm{sgeo}} \tag{56}$$

The corresponding arc weight will be

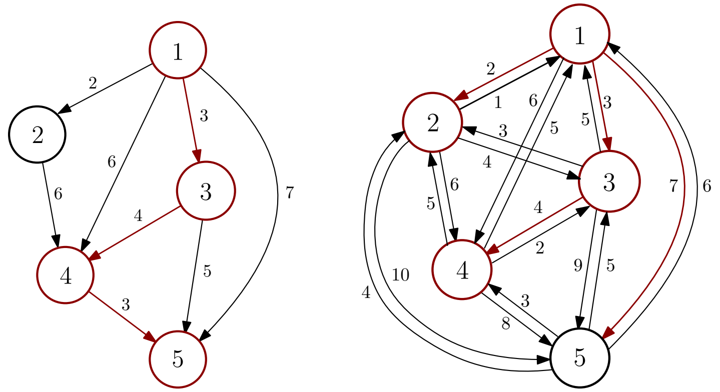
$$w_{ij} = t^{e,\mathrm{obs}}_{\tilde{d}_j,\tilde{p}_j} - t^{e,\mathrm{obs}}_{\tilde{d}_i,\tilde{p}_i} \tag{57}$$

The maximum path weight is instead

$$w_{\max} = \Delta t_{\mathrm{left},\boldsymbol{\sigma}} \tag{58}$$

When exclusively addressing observations of sub-GEO objects, the resulting graph forms a directed acyclic graph (DAG). An example of DAG is shown in Fig. 1a. In a DAG, each arc is directed from one node to the subsequent one in such a way that following these directions will never create a closed loop. This graph arrangement stems from the temporal order of non-GEO object passes, as it is impossible to backtrack to a prior object pass and thus create a loop.

If all objects possess equal scores, the OP problem on a DAG simplifies into the longest shorter path (LSP) problem, previously introduced by Federici et al. for planning active debris removal missions [12]. In the LSP problem, the objective is to find the highest integer q for which the shortest path (i.e., the one with the minimum cumulative weight) originating from the root r and traversing q nodes has a cumulative weight w_{lsp} lower than a predetermined upper limit $w_{\max}$. By supposing that the weights represent distances between the nodes, the name given to



(a) DAG, with the LSP in red, with $r = 1$ and $w_{\max} = 10$.

(b) CDG, with the MSA in red, with $r = 1$.

Fig. 1 Directed acyclic graph (DAG) (**a**) and complete directed graph (CDG) (**b**)

this problem stems from its focus on identifying the longest path (i.e., which passes through the maximum number of nodes) in a graph that is shorter than a given distance.

The LSP problem can be efficiently tackled using dynamic programming. Its time complexity is $O\left(n^2q^2\right)$, thus belonging to the P-hard class of problems. The approach involves iterating over the count of nodes k within the path, spanning from 1 to n. During each iteration, the shortest path originating from r to any other node, while traversing exactly k nodes, is calculated. If the cumulative weight of this path surpasses $w_{\max}$, the solution becomes $k - 1$. Thus, solving the LSP problem reduces to addressing $q(q + 1)/2$ shortest path problems, which are known to be P-hard.

Upon resolving the LSP problem, the heuristic can be obtained by multiplying the count of observed objects q by the average score $\bar{s}$ of the q unscheduled sub-GEO objects in $\boldsymbol{u}_{\boldsymbol{\sigma}}$ with the highest scores. Subsequently, the total observation time corresponding to these objects is subtracted and the number of observed objects q is summed, yielding to

$$h_{\boldsymbol{\sigma}}^{\mathrm{lsp}} = 10^3 q\bar{s} - w_{\mathrm{lsp}} + 10^{-6} q \tag{59}$$

This heuristic is also admissible. Supposing that the actual optimal solution comprises $M \leq U$ observations, as defined by the set of indices $\boldsymbol{m} \subset \{1, \dots, U\}$, the following inequalities hold:

$$q \geq M \tag{60}$$

$$q\bar{s} \geq \sum_{m \in \boldsymbol{m}} s_{\boldsymbol{u}_{\boldsymbol{\sigma}}, m} \tag{61}$$

Indeed, q denotes an upper limit on the maximum number of observations feasible within the remaining observation time, and $\bar{s}$ is the average score of the q highest-scoring objects.

3.2.3 GEO Heuristic

In scenarios where just GEO objects are still unscheduled ($\tilde{d}_i \in \boldsymbol{g}$, $i = 1, \dots, U$), the TTP again reduces into an OP, as the optimal solution still involves observing each object just once for the whole exposure time (i.e., collecting the total score associated to it), as this requires a lower total preparation time (just one per object, instead of one per exposure). However, in this case, the OP takes place on a complete directed graph (CDG), a graph where every distinct pair of nodes is interconnected by two arcs. This kind of graph is shown in Fig. 1b. This structure derives from the telescope's capability of slewing from any GEO object to any other GEO object at any given time during the observation session (thus, loops are indeed possible). Due to this graph structure, the LSP problem heuristic becomes less precise, although it remains admissible. This is because the LSP problem solution could potentially involve multiple traversals through the same graph arcs if their weights are particularly low.

To derive a more accurate and admissible heuristic for GEO object scheduling, the simplifying assumption that all of the unscheduled objects are observable in the remaining observation time is employed. If the exposure times are sufficiently small compared to the time length of the observation session, this assumption holds more reliably when just GEO objects are considered, as there are no mandatory waiting times between consecutive object observations, and the telescope slewing time between the satellites is extremely short. Therefore, the problem can be posed as a minimum-weight Hamiltonian path problem to accurately estimate the total required observation time, which, in this case, is the only factor influencing the value of the objective function.

This problem is a variant of the TSP that does not require closing the tour. In this context, given a weighted graph comprising n nodes, labeled from 1 to n, connected by arcs with weights w_{ij} $(i,j = 1, \dots, n)$, and a designated root node $r \in \{1, \dots, n\}$, the objective is identifying the minimum-weight path commencing from r and passing through each graph node exactly once. In the GEO-only TTP, each node within the graph corresponds to a distinct GEO object in the unscheduled objects $(i = (\tilde{d}_i, \tilde{p}_i)$, $i = 1, \dots, n)$, in addition to the last observation in the state $\boldsymbol{\sigma}$, which serves as the root $(r = (d_N, p_N))$. As for the weight of the arc going from observation i to j, it will be:

$$w_{ij} = \Delta t^{\text{slew}}_{\tilde{d}_i, \tilde{p}_i, \tilde{d}_j, \tilde{p}_j} + \Delta t^{\text{prep}}_{\text{geo}} + \Delta t^{\text{obs}}_{\tilde{d}_i, \tilde{p}_i} \tag{62}$$

Although the minimum-weight Hamiltonian path problem remains NP-hard, a recognized admissible heuristic is the minimum spanning arborescence (MSA). An arborescence is a directed graph where, if r is the root, there exists exactly one directed path from r to v for every node v. An MSA is an arborescence encompassing all graph nodes and with the minimum overall weight. This MSA total weight constitutes as a lower bound for the total weight of the minimum-weight Hamiltonian path. Indeed, the MSA solution arises from a relaxed problem formulation, which assumes that more than two arcs can insist on each node in the solution. Distinct from the minimum-weight Hamiltonian path problem, the MSA problem can be solved in polynomial time $O(mn)$, with $m = n^2$ representing the number of arcs in the graph. Edmonds' algorithm [13] is used for this purpose in this paper. By indicating with w_{msa} the total weight of the MSA, the value of the heuristic in state $\boldsymbol{\sigma}$ will be

$$h^{\text{msa}}_{\boldsymbol{\sigma}} = 10^3 \sum_{i=1}^{U} s_{\boldsymbol{u}_\sigma, i} - \min\{w_{\text{msa}}, \Delta t_{\text{left},\boldsymbol{\sigma}}\} + 10^{-6} U \tag{63}$$

3.2.4 General Heuristic

The LSP and MSA heuristics can be integrated to formulate a novel admissible heuristic in the general scenario encompassing both sub-GEO and GEO satellites. The underlying assumption of this composite heuristic is that the GEO satellites can be observed during the time intervals between successive sub-GEO object passes in the optimal solution. If the total observation time attributed to GEO objects, represented by the weight w_{msa} of the MSA, is either less than or equal to the cumulative waiting time between the different sub-GEO object observations, denoted as

$$\Delta t^{\text{wait}}_{\text{lsp}} = w_{\text{lsp}} - \sum_{i \in \boldsymbol{q}} \Delta t^{\text{obs}}_{\tilde{d}_i, \tilde{p}_i} \tag{64}$$

where $\boldsymbol{q} \subset \{1, \dots, U\}$ is the set of indices identifying the sub-GEO objects in the LSP problem solution, then it is assumed that all GEO objects can be observed within that time window. Otherwise, only a subset of GEO objects can be actually observed between the LEO object passes, and the remaining ones must be scheduled afterward, within the maximum available observation time $\Delta t_{\text{left},\boldsymbol{\sigma}}$.

Given $G_{\boldsymbol{u}_\sigma}$ as the number of GEO satellites within the unscheduled objects $\boldsymbol{u}_{\boldsymbol{\sigma}}$, the combined heuristic in state $\boldsymbol{\sigma}$ is

$$\begin{aligned} h^{\text{comb}}_{\boldsymbol{\sigma}} = {} & 10^3 \left(q\bar{s} + \sum_{\tilde{d}_i \in \boldsymbol{g}} s_{\boldsymbol{u}_\sigma, i} \right) + \min \left\{ \Delta t_{\max,\boldsymbol{\sigma}}, w_{\text{lsp}} + \left[w_{\text{msa}} - \Delta t^{\text{wait}}_{\text{lsp}} \right]^+ \right\} + \\ & + 10^{-6} (q + G_{\boldsymbol{u}_\sigma}) \end{aligned} \tag{65}$$

3.3 Search Space Pruning

Several pruning techniques have been incorporated to reduce the search space size throughout the search procedure, thereby enhancing the overall efficiency of the algorithm. These pruning techniques can be categorized into optimality-preserving ones, i.e., which do not hinder the global optimality of the solution, and beam search techniques, to quickly eliminate large portions of the search space at the cost of losing global optimality guarantees.

3.3.1 Optimality-Preserving Pruning

Two optimality-preserving pruning techniques have been implemented. The first one, occurring before the search procedure is initiated, involves removing from the list of unscheduled observations referred to the initial state of the problem, $\boldsymbol{u}_{\sigma_0}$, all the passes of the sub-GEO objects that are not completely observable for the desired exposure time within the considered observation window, that is, the object-pass pairs $(\tilde{d}_i, \tilde{p}_i)$ for which the following condition is verified

$$\left(t^{s,\text{obs}}_{\tilde{d}_i,\tilde{p}_i} < t_0\right) \vee \left(t^{e,\text{obs}}_{\tilde{d}_i,\tilde{p}_i} > t_0 + \Delta t_{\max}\right), \qquad \tilde{d}_i \in \boldsymbol{l},\ i = 1, \dots, U \tag{66}$$

The second pruning technique involves using a graph version of the A* algorithm to avoid visiting the same nodes again. In this context, in addition to maintaining the frontier, the set of nodes that have been explored up to the current iteration is also retained in memory during the search. If a node is already present in the explored set, it is disregarded, and its potential successors are not generated. Nodes are considered equal (i.e., already explored) if the corresponding states share the same score, total observation time, and list of objects observed. Furthermore, each object within the states of the two nodes must possess the same cumulative count of exposures. It is important to remark that two nodes are considered equal even if their associated states are distinct. For instance, this could occur if the same set of sub-GEO objects, but for the final one, is observed in a different order or if the same count of exposures of one or more GEO objects occurring between the same sub-GEO object passes are taken over different numbers of successive observations.

3.3.2 Beam Search

Despite the effectiveness of the A* search algorithm, it may never converge to the optimal solution when (i) dealing with problem instances with extensive object lists or (ii) a combination of different orbital regimes (sub-GEO and GEO) because of the lower accuracy of the corresponding heuristic. In both cases, the main reason for failure is A* high time and space complexity. A sub-optimal variant of A*, named beam A*-search (BA*), has been devised to address these challenges. BA* is obtained by combining standard beam search [7] and A*, and it can be seen as a heuristic-powered version of the beam best-first algorithm proposed by Zavoli et al. [14] for handling the multi-rendezvous problem released in the 10th global trajectory optimization competition (GTOC X).

In BA*, only a fixed number b_w of nodes, named the beam width, is retained in memory inside the frontier at each iteration. If the frontier width is lower than the beam width, BA* works exactly like A*. When the frontier width reaches the beam width, any additional leaf node, associated with an arbitrary state $\boldsymbol{\sigma}$, is chosen to be included in the frontier according to the following rules:

(i) With a probability p_b, named inclusion probability, the node is added in place of the last (i.e., worst) node $\boldsymbol{\sigma}_w$ in the frontier if $f_\sigma > f_{\sigma_w}$;

(ii) With a probability $1 - p_b$, the node is added in place of the last node $\boldsymbol{\sigma}_w$ in the frontier with a probability equal to

$$p_s = \frac{1}{1 + \frac{f_{\sigma_w}}{f_\sigma}} \tag{67}$$

So, the higher the value of the evaluation function of state $\boldsymbol{\sigma}$ compared to the one of the worst state $\boldsymbol{\sigma}_w$, the higher the probability of being included in the frontier.

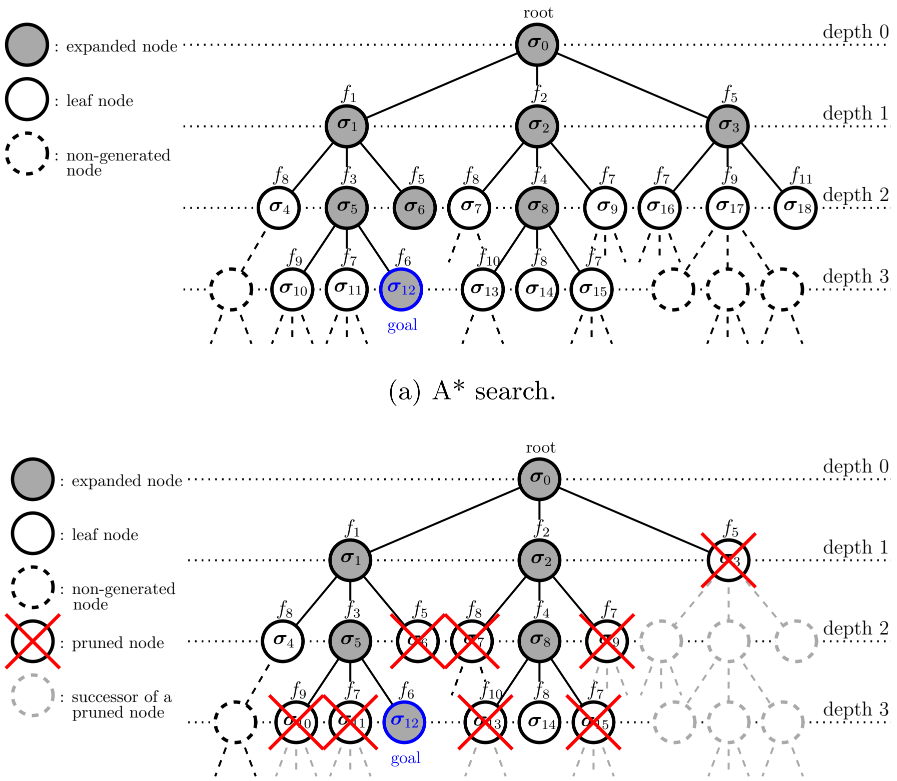
This biased replacement based on the evaluation function is used to avoid filling the frontier with just the newly generated nodes. In BA*, a maximum CPU time is also enforced by terminating the search when a predefined maximum number of nodes $D_{\max}$ has been expanded. The best solution found so far, i.e., the one with the highest J, is returned as the problem solution.

Differently from the standard breadth-first or uniform-cost version of the beam search, the BA* version here adopted relies on the evaluation function f, which includes the heuristic contribute h. So, the biased replacement within the frontier is also based on how promising the states are and not just on their current value of the objective function, thus reducing the likelihood of discarding potentially good solutions and increasing the chances of approaching a goal state close to the global optimum of the problem.

Figure 2 shows a comparison between the search tree generated by A* and BA* on the same example problem. Here, states are numbered based on their generation sequence, and evaluation function values are numbered in descending order (lower subscripts indicate higher function values and, thus, better solutions).

4 Numerical Results

This section delves into the numerical results of the manuscript. The pool of sub-GEO and GEO objects of interest is introduced, as well as the characteristics of the telescope and the relevant information about the night of observation. First, the performance of the A* algorithm by varying the heuristic and the object score range are compared in terms of the number of visited nodes and total computing time. Then, the main hyperparameters of the BA* search algorithm are tuned by looking at the average error with respect to the optimal solution and number of visited nodes obtained in both the sub-GEO-only and GEO-only scenarios. Afterward, the performance of the BA* search algorithm, in terms of total computing time and quality of the obtained solutions, is compared against a standard A* search by varying the number of objects and the total observation time. Eventually, we present the results of both the A* and BA* search algorithms when considering sub-GEO-only,



(a) A* search.

(b) BA* search.

Fig. 2 Search trees generated by A* (**a**) and BA* (**b**) on a sample problem

GEO-only, and mixed object sets by analyzing the number of observed satellites, the cumulative score, and the actual observations realized for different values of the available observation time.

4.1 Study Cases

The telescope considered in this study is Raptors-2, a member of the telescope array within the Space4 Center at the University of Arizona. Raptors-2 is classified as a Newtonian telescope and features an aperture measuring 608.9 mm and a focal length of 2831 mm. This telescope is situated at Biosphere 2, a facility affiliated with the University of Arizona, located 25 miles north of Tucson, Arizona, USA.

For precise details regarding the telescope's specifics and its geographical coordinates, please refer to Table 1.

The observation session starts on July 24th, 2023, at 8:30 pm Pacific time (PT) and is extended for a total duration of 8.5 h. During this nighttime session, the focus

Table 1 Telescope characteristics

Parameter	Value	Unit
λ	32.581	deg
ϕ	−110.847	deg
z	1172	m
ψ_h	170.934	deg
θ_h	69.144	deg
$\omega_{\max}$	2	deg/s
$\theta_{\min}$	30	deg

is on observing satellites from the Globalstar and Iridium constellations in LEO and satellites from the Intelsat and Galaxy constellations in GEO.

Relevant parameters such as the epoch, azimuth, and altitude of the objects at the beginning, highest-elevation point, and conclusion of all their passes over the telescope's location are precomputed, right before the nautical twilight of the observation day, by forward-propagating, with the SGP4 model, their most up-to-date TLEs sourced from the Norad catalog via SpaceTrack.com.

At the beginning of the observation session, the telescope is assumed to be positioned at its home location and prepared to initiate exposures 2 min before the first scheduled pass of one of the LEO objects.

Detailed specifics about the beginning of the observation session, the total count of visible LEO and GEO satellites from Raptors-2, the required exposures and exposure duration for GEO satellites, as well as the preparatory time before capturing exposures of LEO and GEO satellites, are outlined in Table 2. For the sake of simplicity, the exposure time associated with each of the LEO objects has been assumed to be slightly longer than its longest pass time, so that they are always observed for the whole pass time, that is

$$\Delta t^{\text{obs}}_{l_i,pl} = t^{e}_{l_i,p} - t^{s}_{l_i,p}, \qquad p = 1, \dots, P_{l_i},\ i = 1, \dots, L \tag{68}$$

The implementation of the A* and BA* search algorithms has been realized in C++, and the execution was performed on a workstation housing an AMD Ryzen 9

Table 2 Observation and object data

Parameter	Value	Unit
t_0	60150.1377	MJD
L	90	–
$\sum_{i=1}^{L} P_{l_i}$	107	–
$\Delta t^{\text{prep}}_{\text{leo}}$	90	s
G	33	–
E_{g_i}	5	–
$\Delta t^{\text{exp}}_{g_i}$	45	s
$\Delta t^{\text{prep}}_{\text{geo}}$	15	s

7950X 16-core processor operating at 5.8 GHz and equipped with 128 GB of RAM. The execution was parallelized across the 32 logical cores of the processor using OpenMP directives.

4.2 A* Heuristic Performance Analysis

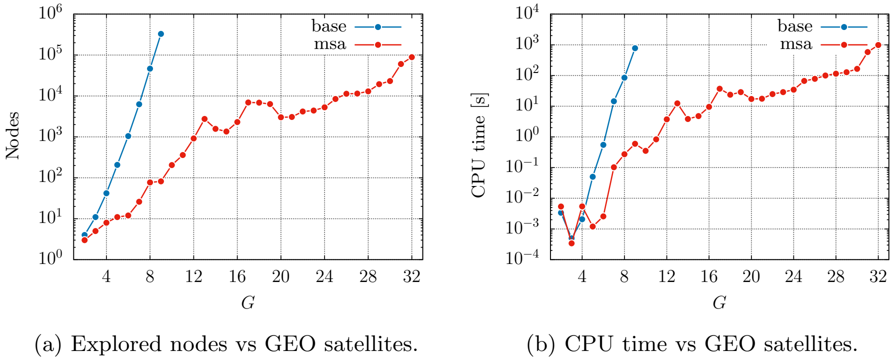
The exact (i.e., optimal) version of the A* algorithm has been considered first to study the performance of the proposed heuristics with different sets of target object, score ranges, and total observation times.

4.2.1 Effect of the Problem Dimension

First, the scores of the objects have been fixed (all equal to 1) and the performance of the heuristics has been analyzed on increasingly harder instances of the TTP by considering only GEO or LEO satellites.

Specifically, Fig. 3 shows the trends of the number of visited nodes (Fig. 3a) and total CPU time (Fig. 3b) obtained with the two heuristics for GEO objects (base and MSA) by varying the pool of GEO satellites G from 2 to 32 and keeping the total observation time fixed to $\Delta t_{\max} = 8.5\,\text{h}$. It is easy to note that both the number of explored nodes and the corresponding CPU time grow exponentially with the number of objects when the base heuristic is used. In this case, the algorithm is not even able to solve problem instances with more than 9 objects because of complete memory saturation. Conversely, when the MSA heuristic is used, the two figures of merit feature a much slower growth with the problem dimension, and the A* code can find a globally optimal solution for every problem instance considered. The maximum CPU time required for scheduling 32 GEO satellites with the MSA heuristic is about 20 min, which confirms the necessity of introducing an approximate version of the algorithm to cope with bigger problem instances in reasonable times.

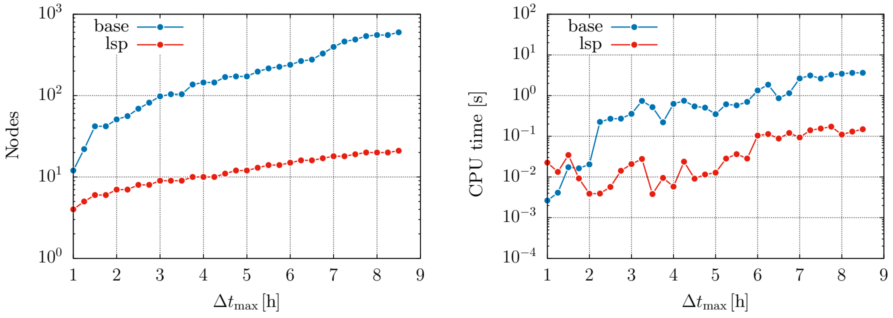
Figure 4 shows the trends of the number of visited nodes (Fig. 4a) and total CPU time (Fig. 4b) that are instead obtained with the two heuristics for LEO objects (base



(a) Explored nodes vs GEO satellites.

(b) CPU time vs GEO satellites.

Fig. 3 Performance comparison between the base and MSA heuristic for GEO satellites scheduling with A*



(a) Explored nodes vs observation time.

(b) CPU time vs observation time.

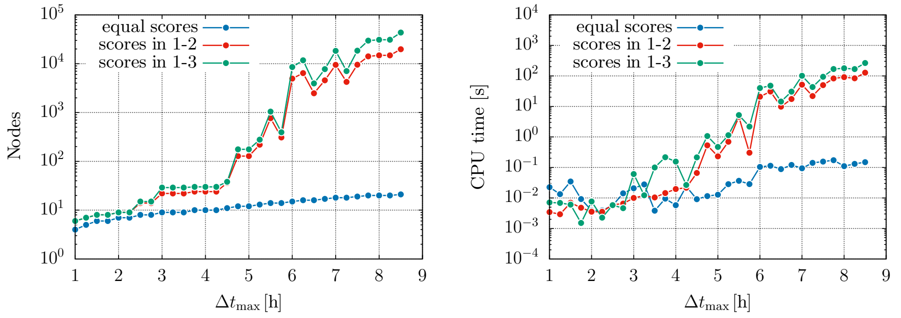
Fig. 4 Performance comparison between the base and LSP heuristic for LEO satellites scheduling with A*

and LSP) by varying the total observation time $\Delta t_{\max}$ from 1 h to 8.5 h, and always considering all the $L = 90$ satellites. As before, the A* algorithm with the base heuristic features a larger number of visited nodes and total CPU time, up to one order of magnitude higher than the LSP heuristic. However, their growth with the observation time is slower than in the GEO case, thanks to the use of a graph search version of the algorithm, which removes all the LEO object permutations that yield the same value of the merit index. The results obtained with the ad-hoc designed heuristic (LSP) are even better in this case, as the two figures of merit increase just linearly with the observation time. In this case, the CPU time is always lower than 0.2 s, even when the entire night and all the 90 target objects are considered.

It is worth mentioning that an analysis of the A* performance when varying the number of sub-GEO objects, as done in the GEO object scheduling scenarios, has not been conducted. This is because this effect is implicitly considered in the analysis focused on variations in the total observation time. Altering the total observation time influences the visibility of certain sub-GEO object passes, thereby excluding specific sub-GEO satellites from the search due to the preliminary object pruning detailed in Sect. 3.3.1. Specifically, varying the total observation time from 1 to 8.5 h corresponds to a change in the number of observable sub-GEO satellites from 18 to 90, with an increase of approximately 10 objects per additional observation hour.

4.2.2 Effect of the Score Range

Then, the effect of varying the scores of the target objects on the LSP heuristic performance has been analyzed for LEO objects only. Results with the MSA heuristic for the GEO-only case have been omitted as they produce identical metric trends, as, given the exposure times here considered, all the GEO objects are always scheduled during the 8-hour observation session, so the scores do not have an impact on the heuristic accuracy. Figure 5 shows the trends of the number of visited nodes



(a) Explored nodes vs observation time.

(b) CPU time vs observation time.

Fig. 5 LSP heuristic performance comparison by varying the LEO satellite score range with A*

(Fig. 5a) and total CPU time (Fig. 5b) obtained with the LSP heuristic by varying the total observation time $\Delta t_{\max}$ from 1 to 8.5 h and with different scores for the $L = 90$ LEO satellites (all scores equal to 1, random scores in [1, 2], and random scores in [1, 3]).

As expected, being the LSP heuristic based on an equal-score assumption, its accuracy worsens when tackling problems with different scores: the more scattered the scores, the slower the search process. However, it is worth noting that, while both the number of visited nodes and the CPU time increase considerably when switching from an equal-score to a different-score scenario, the same performance metrics do not change much when widening the score range from [1, 2] to [1, 3], thus showing a good degree of robustness of the heuristic against a larger scattering of the object scores, which, in a normal operative scenario, are not expected to be too much different between each other.

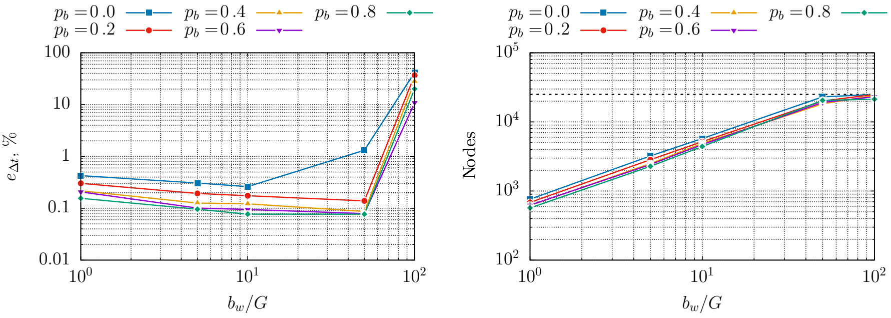
4.3 BA* Algorithm Performance Analysis

The beam version of the A* algorithm (BA*) has been then considered, and its efficiency and accuracy have been assessed against the optimal version of A* on different sets of target objects.

4.3.1 Hyperparameter Tuning

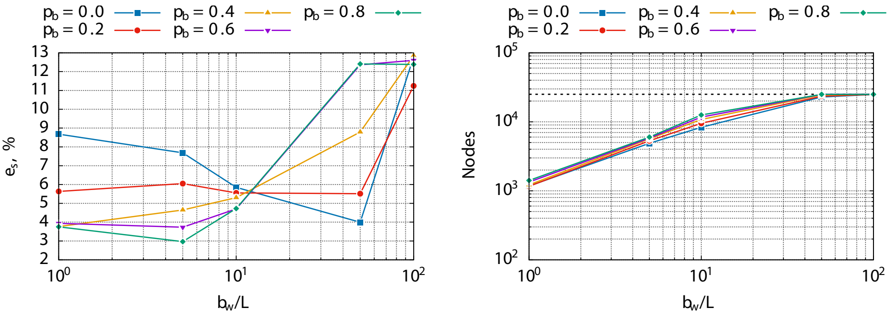
First, the effect of the two hyperparameters of BA*, namely the beam width b_w and the inclusion probability p_b, on the algorithm performance has been analyzed by considering two instances of the TTP involving either all the GEO satellites or the LEO satellites, a maximum observation time $\Delta t_{\max} = 8.5$ h, and random object scores sampled in the interval [1, 3].

Figures 6 and 7 show the trends of the algorithm performance measure (Fig. 6a and 7a) and average number of explored nodes (Fig. 6b and 7b) obtained by both varying the beam width and the inclusion probability when only GEO or LEO objects are considered, respectively. For each pair of values (b_w, p_s), 50 independent runs have been



(a) Relative error in observation time. (b) Number of explored nodes.

Fig. 6 Performance analysis of BA* for GEO satellites scheduling for different beam widths b_w and inclusion probabilities p_b



(a) Relative error in cumulative score. (b) Number of explored nodes.

Fig. 7 Performance analysis of BA* for LEO satellites scheduling for different beam widths b_w and inclusion probabilities p_b

realized to mitigate the stochastic nature of the algorithm and collect statistics. The maximum number of explored nodes has been fixed to $D_{\max} = 25000$. Two different performance measures or error metrics have been used when scheduling GEO or LEO objects, respectively. Let us name with $s^\star$ and $\Delta t^\star_{\text{tot}}$ the score and total observation time of the optimal solution, retrieved via standard A*. With GEO objects, the performance measure corresponds to the average relative error in total observation time with respect to the optimal solution:

$$e_{\Delta t} = \frac{|\Delta t^\star_{\text{tot}} - \Delta t_{\text{tot},\sigma}|}{\Delta t^\star_{\text{tot}}} \tag{69}$$

being the score always the same between the A* and BA* solutions as all scheduled GEO objects are always observed. Conversely, with LEO objects, the performance measure is the average relative error in the cumulative score with respect to the optimal solution:

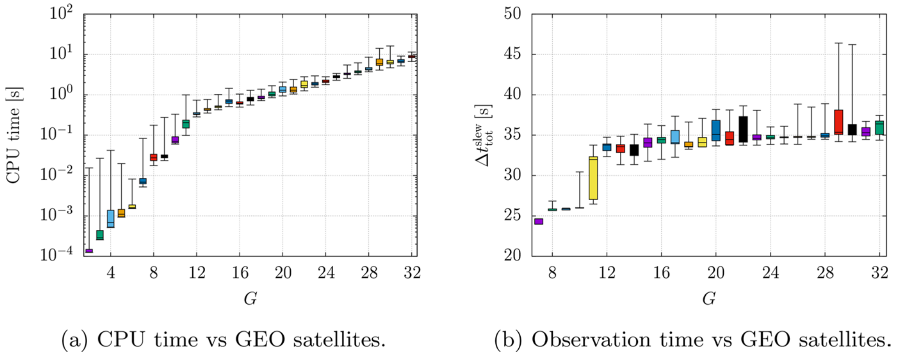
$$e_s = \frac{|s^\star - s_\sigma|}{s^\star} \tag{70}$$

An alternative and more precise error metric could be the relative error in the objective function J_σ in Eq. (52), which accounts for the score, the total observation time, and the number of objects. However, the scaling factors used to weigh the last two terms make it difficult to discern differences in these values. To address this, the effects have been separated, and only the leading one (score for LEO objects and observation time for GEOs) has been used to define the error metrics. This approach ensures that the most significant factors for each type of object are appropriately emphasized in the error analysis that follows.

Figures 6b and 7b confirm that, in either case (GEO and LEO object scheduling), the number of explored nodes grows exponentially with the beam width b_w, as a direct consequence of the NP-hardness of the combinatorial problem. The count of nodes reaches its maximum value, denoted as $D_{\max} = 25000$, when the beam width surpasses 50 times the count of scheduled objects. By looking at Figs. 6a and 7a, one can notice that, with such values of b_w, the errors increase rapidly in both the GEO-only and LEO-only scenarios. In these cases, the BA* algorithm fails to reach a goal node within the available computational time, yielding an incomplete solution with a consistently lower score compared to the optimal one. Thus, as expected, the best value of the beam width is highly dependent on the available computational budget, that is, on the value of $D_{\max}$. For lower values of the beam width ($b_w \leq 50\,G$ and $b_w \leq 50\,L$ respectively), the behavior of BA* diverges between the GEO and LEO scenarios. In the former, as expected, errors exhibit a monotonous decline as both the beam width and inclusion probability increase. This decline leads to an error value below 0.1%, which corresponds to approximately 10 s in total observation time. This level of accuracy is achieved when utilizing $p_b = 0.8$ with either $b_w = 5\,G$ or $b_w = 10\,G$. However, for the LEO case, the error features a minimum point that is dependent on the value of p_b. With lower inclusion probabilities ($p_b = 0$ and $p_b = 0.2$), a higher beam width ($b_w = 50\,L$) is required to converge to a good quality solution, as the broader pool of nodes that are kept in memory at each iteration mitigates the effect of the higher probability of discarding promising solutions. Conversely, with higher inclusion probabilities ($p_b = 0.6$ and $p_b = 0.8$), a relatively low value of the beam width ($b_w = 5L$) is sufficient to find solutions closer to the optimal one. In any case, also with LEO target objects, the minimum error value ($e_s < 3\%$) is achieved with $p_b = 0.8$ and $b_w = 5\,L$. For this reason, the analyses in the following sections have been conducted using this specific configuration of the algorithm.

4.4 Performance Statistics

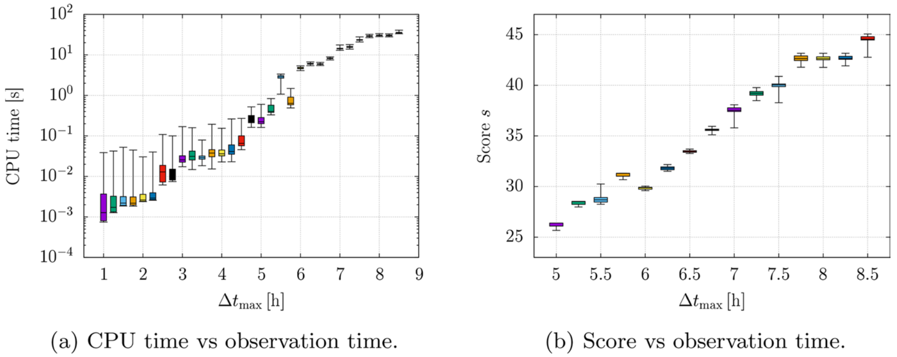
The performance of the best BA* configuration found so far has then been assessed on increasingly harder instances of the telescope tasking problem, using only GEO or LEO satellites and random scores in the interval [1, 3]. For each problem instance, 100 independent runs of BA* have been performed to collect statistics.



(a) CPU time vs GEO satellites.

(b) Observation time vs GEO satellites.

Fig. 8 Performance statistics of BA* for GEO satellites scheduling



(a) CPU time vs observation time.

(b) Score vs observation time.

Fig. 9 Performance statistics of BA* for LEO satellites scheduling

Figure 8 presents the boxplots generated from 100 BA* runs, depicting the CPU time (Fig. 8a) and the total slewing time (Fig. 8b) for GEO object scheduling. The pool of GEO satellites G varies from 2 to 32, while the total observation time is fixed at $\Delta t_{\max} = 8.5$ h. The total slewing time $\Delta t_{\text{tot}}^{\text{slew}}$ has been selected as a performance metric in this case because it allows us to better appreciate differences between subsequent runs, as the total exposure time $\Delta t_{\text{obs},\boldsymbol{\sigma}}$ and preparation time $G\Delta t_{\text{geo}}^{\text{prep}}$ remain the same across all solutions.

Conversely, Fig. 9 displays the boxplots for CPU time (Fig. 9a) and cumulative score (Fig. 9b) for LEO object scheduling. In this case, the total observation time $\Delta t_{\max}$ varies from 1 h to 8.5 h, consistently considering all $L = 90$ satellites. The boxplots corresponding to problem instances with fewer than 6 satellites for GEO object scheduling and with a total time of less than 5 h for LEO object scheduling have not been included in Figs. 8b and 9b because BA* was able to find the optimal solution in every run, as will be shown in the next section.

The results indicate that the variability in total CPU time is more pronounced in percentage terms for the easiest problem instances, where the average value is small (less than 0.1 s). However, as the problem becomes more complex, this variability decreases, demonstrating that BA* consistently finds solutions by exploring a similar number of search nodes. In both cases (GEO and LEO object scheduling), the average CPU time increases exponentially with the problem size, but with a much lower slope compared to A*.

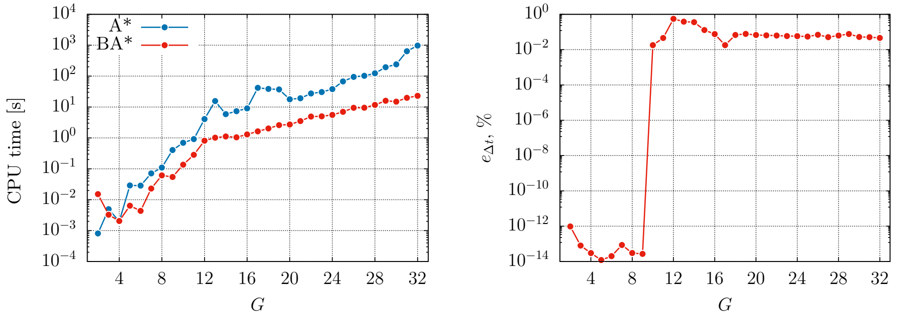
The variability in performance differs between the GEO and LEO cases. For GEO scheduling, the variability is broader in some test cases and includes some distant outliers. It is important to note, however, that the differences between solutions are on the order of a few seconds in total observation time over several hours of telescope usage. For LEO scheduling, the variability is almost absent, with BA* consistently finding solutions with very similar scores, highlighting the robustness of the proposed approach against randomness.

4.5 Performance Comparison with A*

BA* has then been systematically compared against standard A* on the same problem instances. In this case, the best solution over 50 runs of BA* has been used for comparison with A*.

Figure 10 shows the trends of the CPU time (Fig. 10a) and of the error on the total observation time (Fig. 10b) obtained with BA* and A* for GEO object scheduling by varying the pool of GEO satellites G from 2 to 32 and keeping the total observation time fixed to $\Delta t_{\max} = 8.5\,\text{h}$. Figure 11, instead, shows the trends of the CPU time (Fig. 11a) and of the error on the cumulative score (Fig. 11b) that are obtained with BA* and A* for LEO object scheduling by varying the total observation time $\Delta t_{\max}$ from 1 h to 8.5 h, and always considering all the $L = 90$ satellites.

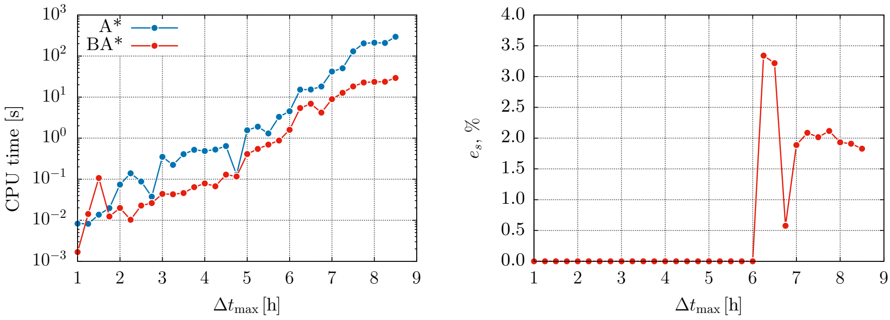
In both scenarios, it is evident that BA* consistently reduces the overall time needed to reach a goal state (Figs. 10a and 11a). Specifically, the CPU time needed to schedule all of the GEO objects within the maximum observation time diminishes by almost two orders of magnitude, going from 16 min with A* to 23 s with BA*. For LEO satellites, the time drops from 5 min with A* to less than 30 s with BA*, which corresponds to a reduction of almost one order of magnitude.



(a) CPU time vs GEO satellites.

(b) Solution error vs GEO satellites.

Fig. 10 Performance comparison between BA* and A* for GEO satellites scheduling



(a) CPU time vs observation time.

(b) Solution error vs observation time.

Fig. 11 Performance comparison between BA* and A* for LEO satellites scheduling

While this significant improvement in computational speed does come at the expense of sacrificing the guarantee of a globally optimal solution, the trade-off is justified. This justification is particularly clear when looking at Figs. 10b and 11b, which illustrate that the solutions produced by BA* deviate by less than 0.08% in the case of scheduling GEO objects, and by less than 3.5% when it comes to scheduling LEO objects. This level of difference in the merit index values supports the use of BA* instead of standard A* or similar exact solution algorithms, especially when dealing with extensive lists of target objects. It's worth highlighting that the errors introduced by the BA* algorithm do not increase alongside the problem's complexity. Instead, the errors are almost zero when sufficiently small sets of objects are considered, then rise quickly and converge towards a fixed value with larger object pools.

4.6 Optimal Observation Schedules

Eventually, this section presents and discusses the optimal observation schedules obtained with either A* or BA* when considering only GEO, LEO, or both GEO and LEO satellites together as target objects, and the entire night as observation session.

4.6.1 GEO Object Scheduling

Figure 12 shows the path of the telescope pointing direction in the sky for the optimal solution obtained with A* all the 33 GEO satellites. The total observation time is 2.21 h, with 2.06 h being the actual cumulative exposition time. It is interesting to note that the GEO objects do not follow a simple ordering in azimuth, which is the most commonly used in sensor tasking applications. Instead, they are arranged in a more complex way to minimize the total telescope slewing time, that is, the cumulative path in the sky. The obtained solution indeed corresponds to the solution of the TSP referred to the target GEO objects, departing from the telescope's home

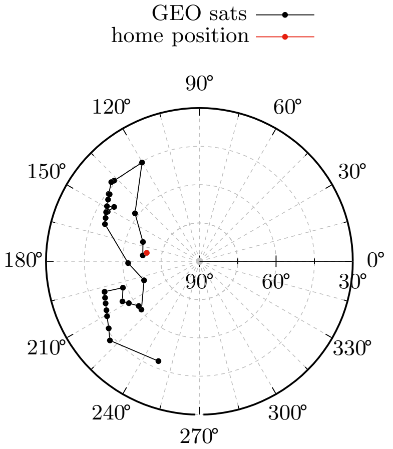


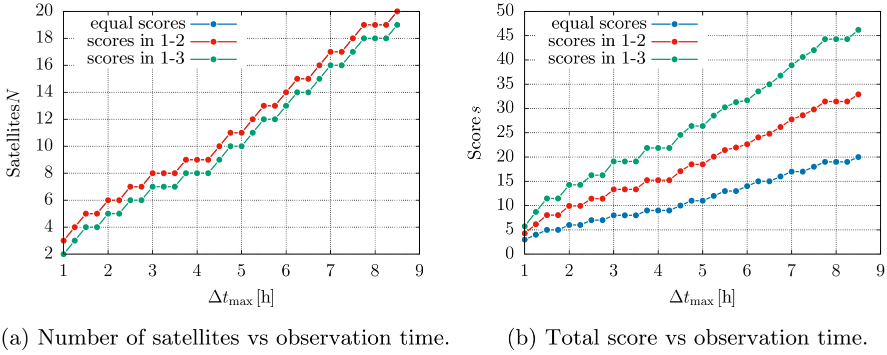
Fig. 12 GEO object observation. The concentric circles represent different values of the elevation angle (degree), while the radii represent different values of the azimuth angle (degree)

position. This time-optimal solution is also the most convenient from an energy consumption point of view, as it allows reducing the movements of the mount motors to a minimum.

4.6.2 LEO Object Scheduling

Figure 13 shows the trends of the number of observed satellites (Fig. 13a) and total score (Fig. 13b) obtained with A* and the LSP heuristic by varying the observation time $\Delta t_{\max}$ from 1 h to 8.5 h and using different scores for the 90 LEO satellites (all scores equal to 1, random scores in [1, 2], and random scores in [1, 3]). The corresponding solutions are reported in Table 3, where, for either score distribution, the list of Norad IDs, the corresponding object scores, and the time passed till the end of the observations, are listed.

Of course, both the number of observed satellites and the total score increase with the observation time, and the wider the score range, the higher the cumulative score



(a) Number of satellites vs observation time.

(b) Total score vs observation time.

Fig. 13 Optimal solutions with A* by varying the LEO satellite score range and the observation time

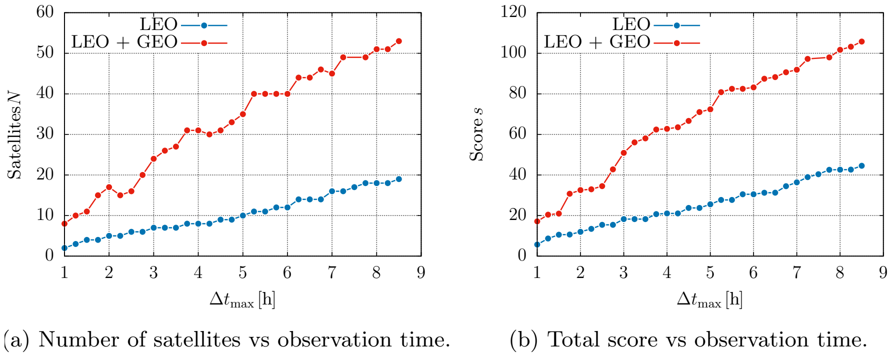
Table 3 Optimal LEO object observation schedules with different score distributions by A*

Equal scores			Scores in [1, 2]			Scores in [1, 3]		
ID	s	$t^e - t_0$ [h]	ID	s	$t^e - t_0$ [h]	ID	s	$t^e - t_0$ [h]
43929	1	0.33	37190	1.46	0.38	32264	2.87	0.43
42807	1	0.63	42807	1.07	0.63	43927	2.86	0.78
25273	1	0.84	24944	1.65	0.85	24948	2.97	1.07
24948	1	1.07	24948	1.98	1.07	31574	2.77	1.40
43922	1	1.38	31574	1.88	1.40	31573	2.80	1.96
37744	1	1.89	31573	1.90	1.96	25946	1.99	2.42
25676	1	2.35	25946	1.49	2.42	25306	2.83	2.83
25306	1	2.82	25306	1.91	2.83	25943	2.78	3.75
25307	1	3.60	25943	1.89	3.75	39076	2.69	4.32
25853	1	4.30	39076	1.85	4.32	37743	1.83	4.74
37743	1	4.74	37743	1.42	4.74	25910	2.14	5.17
25623	1	5.14	25910	1.57	5.17	25963	2.78	5.57
31571	1	5.47	25963	1.89	5.57	25944	1.26	6.03
38044	1	5.85	25944	1.13	6.03	32265	2.42	6.47
25679	1	6.20	32265	1.71	6.47	43478	1.78	6.70
41923	1	6.66	43478	1.39	6.70	41919	2.13	6.98
41919	1	6.98	41919	1.56	6.98	43479	2.61	7.27
43479	1	7.27	43479	1.80	7.27	43481	2.77	7.57
43481	1	7.57	43481	1.89	7.57	43254	1.91	8.44
37741	1	8.25	43254	1.46	8.44	–	–	–

achieved. It is interesting to note that, with either equal scores or scores in [1, 2], the number of observed satellites is the same for every observation time $\Delta t_{\max}$. However, by looking at the optimal sequences for $\Delta t_{\max} = 8.5$ h (Table 3), they just share 7 satellites out of 20, thus confirming that the objects with highest scores indeed drove the search in the latter case. With scores in [1, 3], although the optimal solutions have one satellite less, the cumulative score is always higher than in the other two cases. So, the algorithm preferred to sacrifice one of the satellites to be able to observe objects with a higher score, but which also require longer waiting and/or pass times. Indeed, one can notice that, in the optimal solution with $\Delta t_{\max} = 8.5$ h, although the last 17 objects are observed in the same order and at the same epochs as in the solution with scores in [1, 2], the time used to observe the first three objects in this latter solution is exploited in the solution with scores in [1, 3] to track just two high-scoring objects.

4.6.3 Mixed LEO–GEO Object Scheduling

Figure 14 shows the trends of the number of observed satellites (Fig. 14a) and the total score (Fig. 14b) obtained with BA* by varying the observation time $\Delta t_{\max}$ from 1 h to 8.5 h and by considering both the LEO and GEO objects together, with



(a) Number of satellites vs observation time.

(b) Total score vs observation time.

Fig. 14 Optimal solutions by varying the observation time for LEO-only and LEO+GEO object sets

random scores in the interval [1, 3]. For the sake of comparison, the same solutions but by just considering the LEO objects have been also reported.

It is interesting to note that, when objects in both orbital regimes are considered, the count of observed satellites does not increase monotonically with the observation time, as, for some observation times, the algorithm was able to find a solution where some low-scoring satellites are replaced by a lower number of higher-scoring objects, thus anyway leading to an improved cumulative score. In this case, the telescope is able to observe a higher number of satellites and collect a higher cumulative score in the same observation time by efficiently scheduling GEO objects between successive passes of LEO objects. Thus, the contemporary scheduling of both LEO and GEO objects optimizes the utilization of the available observation time. Specifically, in 8.5 h, the telescope can observe a total of 53 objects, which include all 33 GEO satellites plus 20 LEO satellites. Figure 15 shows the number of observations and number of observed satellites for the mixed LEO–GEO scenario for different observation times $\Delta t_{\max}$. It is interesting to note that the number of observations realized differs from the number of distinct satellites tracked, especially for longer observation times (in 8.5 h, 61 observations of 53 distinct satellites), as opposed to the LEO-only

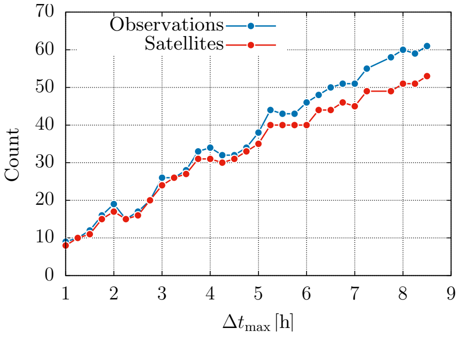


Fig. 15 Number of observations and satellites observed by varying the observation time for the LEO+GEO object set

scenario. This phenomenon occurs since some of the GEO objects are observed multiple times for fractions of the total number of exposures to better exploit the waiting time between LEO observations and further optimize the cumulative score collected.

5 Conclusion

This paper addressed the problem of optimal telescope tasking for space domain awareness purposes. The main focus was the prioritized scheduling of observations for known Earth-orbiting satellites using a single telescope. The problem was posed as a purely combinatorial problem and solved with the application of a beam variant of the A* search algorithm, termed beam A*-search (BA*).

Three novel heuristics were introduced to expedite the scheduling process via A*, each tailored to specifically deal with a distinct orbital regime (sub-GEO, GEO) or with all of them. These heuristics were devised as exact solutions of relaxed versions of the underlying combinatorial problem, ensuring their admissibility and, thus, the global optimality of the search procedure. Different pruning techniques were also proposed to reduce the dimension of the solution space and further speed up the search process. Among them, the beam search framework was applied to standard A* search to create a sub-optimal variant of the algorithm, wherein only a fixed number of promising solutions is retained in memory at each iteration. The solutions to be included within the frontier are chosen according to a biased random sampling based on the value of the node evaluation function, which contains information coming both from the problem objective function and A* heuristic.

The results obtained showcase the effectiveness of the proposed heuristics in terms of time complexity, as compared to a benchmark heuristic tailored for the specific problem when using the exact version of A* algorithm. Specifically, the proposed heuristics lead to a decrease of one order of magnitude in overall run time when scheduling LEO objects, and of up to four orders of magnitude when observing GEO objects. The effect of varying the range of priority indices, or scores, for the scheduled objects, is also investigated in detail to understand how it affects the performance of the different A* heuristics. Results show that, while broader ranges do increase the problem complexity, the A* search algorithm remains capable of finding optimal solutions within a reasonable amount of time.

The effectiveness and accuracy of the beam version of A* algorithm, BA*, have been then investigated. First, the main hyperparameters of the BA* search method, namely the beam width and the inclusion probability, have been properly tuned to identify the pair of values that corresponds to the best compromise between the quality of the obtained solution and the cumulative run time of the search. The results obtained show that a value of inclusion probability of 0.8 and a beam width equal to five times the number of scheduled objects yield the best results when considering only GEO objects or LEO objects.

BA* has been then compared against standard A* in terms of computing time and solution accuracy on object sets of different sizes involving just LEO or GEO satellites and by also varying the total observation time. Results showcase that, at the cost of converging to a sub-optimal solution with a value of the merit index that is less than 0.08% lower than the optimal one in the GEO-only case, and less than 3% lower in the LEO-only case, BA* can cut down the overall run time up to a factor 100 in the hardest problem scenarios considered.

A test set including all the target LEO and GEO objects has been eventually considered to understand how it affects the number of objects observed, the number of observations realized, and the total score collected. The obtained solutions show that when LEO and GEO objects are considered together, the latter kind of satellites are scheduled within successive passes of LEO objects, thus yielding a more efficient use of the available observation time.

As possible future research directions, it would be worthwhile to investigate alternative heuristics that can potentially perform better on different test sets. For example, when larger sets of GEO objects (in the order of hundreds) are considered, using the undirected version of the minimum spanning arborescence, called the minimum spanning tree, may yield faster solutions. This is because it is quicker to compute, albeit slightly less accurate. Additionally, if some GEO objects are associated with very long total exposure times, a new possible heuristic could be based on solving the knapsack problem. This approach prioritizes scores over total observation time, better fitting the situation where the assumption that all GEO objects are observable no longer holds.

Data Availability The authors declare that the data supporting the findings of this study are available within the paper. The TLEs of the LEO and GEO satellite constellations used in the results section can be freely retrieved from Space-Track.org at the following link: https://www.space-track.org/.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no Conflict of interest.

References

1. Frueh, C., Fielder, H., Herzog, J.: Heuristic and optimized sensor tasking observation strategies with exemplification for geosynchronous objects. J. Guid. Control Dyn. **41**(5), 1036–1048 (2018). https://doi.org/10.2514/1.G003123
2. Sunberg, Z., Chakravorty, S., Erwin, R.S.: Information space receding horizon control for multisensor tasking problems. IEEE Trans. Cybern. **46**(6), 1325–1336 (2016). https://doi.org/10.1109/TCYB.2015.2445744
3. Fedeler, S., Holzinger, M.: Monte Carlo tree search methods for telescope tasking. In: AIAA Scitech 2020 Forum, AIAA 2020-0659 (2020). https://doi.org/10.2514/6.2020-0659
4. Siew, P.M., Jang, D., Roberts, T.G., Linares, R., Fletcher, J.: Cislunar space situational awareness sensor tasking using deep reinforcement learning agents. In: 2022 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS), Maui, Hawaii (2022)

5. Siew, P.M., Linares, R.: Optimal tasking of ground-based sensors for space situational awareness using deep reinforcement learning. Sensors **22**(20), 7847 (2022). https://doi.org/10.3390/s22207847
6. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. **4**(2), 100–107 (1968). https://doi.org/10.1109/TSSC.1968.300136
7. CMU Speech Group: Speech understanding systems: summary of results of the five-year research effort at Carnegie-Mellon University (1977). https://doi.org/10.1184/R1/6609821.v1
8. Klein, D., Manning, C.D.: A* parsing: Fast exact viterbi parse selection. In: Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pp. 119–126 (2003)
9. Meng, B.-B., Gao, X.: UAV path planning based on bidirectional sparse A* search algorithm. In: 2010 International Conference on Intelligent Computation Technology and Automation, vol. 3, pp. 1106–1109. IEEE (2010). https://doi.org/10.1109/ICICTA.2010.235
10. Cui, X., Shi, H.: A*-based pathfinding in modern computer games. Int. J. Comput. Sci. Netw. Secur. **11**(1), 125–130 (2011)
11. Gunawan, A., Lau, H.C., Vansteenwegen, P.: Orienteering problem: a survey of recent variants, solution approaches and applications. Eur. J. Oper. Res. **255**(2), 315–332 (2016). https://doi.org/10.1016/j.ejor.2016.04.059
12. Federici, L., Zavoli, A., Colasurdo, G.: On the use of A* search for active debris removal mission planning. J. Space Saf. Eng. **8**(3), 245–255 (2021). https://doi.org/10.1016/j.jsse.2021.07.003
13. Edmonds, J.: Optimum branchings. J. Res. Natl. Bureau Stand. B **71**(4), 233–240 (1967). https://doi.org/10.6028/jres.071b.032
14. Zavoli, A., Federici, L., Benedikter, B., Casalino, L., Colasurdo, G.: GTOC X: solution approach of Team Sapienza-PoliTo (AAS 19-894). In: Horneman, K.R. et al. (eds.) Astrodynamics 2019. Advances in the Astronautical Sciences, vol. 171, pp. 3327–3342. Univelt, San Diego (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.