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

Lorenzo Federici, Andrea D'Ambrosio, Roberto Furfaro
Vishnu Reddy
The University of Arizona

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 problem 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 for scheduling just LEO, GEO, or both kinds of objects. Simulations are conducted with real space objects and telescope data to evaluate the effectiveness, time complexity, and performance of the search algorithm under different real-world scenarios.

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 a number of 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, objects with a few passes might receive a higher priority on the nights they are visible from the selected sensors.

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 designated filters.

Achieving the best possible sensor tasking is a challenging optimization problem that, in general, 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 solution that is generally suboptimal. 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, emerging machine learning methods, such as deep reinforcement learning,[4,5] have also been adopted and exhibited good levels of accuracy and adaptability when facing changes in object orbits, observation windows, observer locations, and sensor characteristics. These approaches are capable of learning from past instances of the problem and adapting to novel data. However, they lack the ability to exploit the inherent mathematical structure of the underlying problem, frequently resulting in long training periods and an absence of guarantees regarding the optimality or robustness of the final solution.

The formulation of the sensor tasking problem used in this paper is 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, so that 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 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 in a graph, and a heuristic function that estimates the "distance" 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 a cost based on its distance from the start node and the estimated distance to the goal. The algorithm selects the node with the lowest cost and explores its neighbors, updating the costs of 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 simplicity is usually required. Three admissible heuristics for A* are proposed in this study, the first valid for scheduling LEO objects, the second for scheduling GEO objects, and the third one combining the other two when objects in different orbital regimes are considered (LEO, GEO). Being A* an exact solution algorithm, it can struggle to find an optimal solution in a reasonable computational time based on the problem's complexity. To reduce the size of the graph, pruning techniques will be used to eliminate, from the priority queue, schedules that are unlikely to yield a high scientific output, by implementing a beam search version of the A* search algorithm.[7] This pruning of unpromising schedules allows for a considerable speed-up of the search, at the cost of losing guarantees on the global optimality of the solution.

This beam search algorithm, named beam A*-search (BA*), is compared against an exact version of the A* search to understand its accuracy and time complexity when dealing with increasingly bigger sets of objects or longer observation times involving just GEO or LEO satellites. Eventually, the results obtained when scheduling objects in mixed orbital regimes (both LEO and GEO) are presented and discussed in detail by varying the available observation time. Simulations are carried out using actual object data 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, AZ).

2. PROBLEM STATEMENT

In this section, the telescope tasking problem (TTP) is introduced and mathematically formulated 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, and it is capable of being adjusted using a motorized altazimuth mount. This mount has a maximum rate of motion, or slew rate, denoted as $\omega_{\max}$.

During a given day, the telescope becomes operational at a specific time called epoch t_0. This timing is chosen to coincide with nautical twilight, defined as the time when the Sun's elevation relative to the local horizon drops below $-12\,\text{deg}$. The telescope remains active for a duration of $\Delta t_{\max}$, typically extending throughout the entire night until the subsequent nautical twilight, when the Sun's elevation rises above $-12\,\text{deg}$ 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 assigned the task of 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 $\boldsymbol{l} = l_1, \ldots, l_L$ of L objects positioned in LEO and another collection $\boldsymbol{g} = g_1, \ldots, g_G$ of G objects in GEO are intended to be scheduled for observation during the aforementioned night.

Each object in the LEO 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 $p = 1,\ldots,P_{l_i}$, and each starts at time $t^s_{l_i,p}$ while ending at time $t^e_{l_i,p}$. These 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 $(\theta^s_{l_i,p}, \psi^s_{l_i,p})$, $(\theta^e_{l_i,p}, \psi^e_{l_i,p})$, respectively. Should LEO object l_i be observed during its p-th pass, the telescope will continuously follow and monitor the object throughout the entire duration of that pass. As a result, the total duration of observation for object l_i will be

$$\Delta t^{\text{obs}}_{l_i,p} = t^e_{l_i,p} - t^s_{l_i,p}, \qquad l_i \in \boldsymbol{l},\, p = 1,\ldots,P_{l_i} \tag{1}$$

It is supposed that the duration of passes for LEO objects is sufficiently long to enable the telescope to acquire all the necessary exposures for the given object, denoted as E_{l_i}. The score s_{l_i} linked to the LEO 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. Each GEO object $g_i \in \boldsymbol{g}$ is linked to a designated count of exposures, denoted as E_{g_i}, each with the same total duration $\Delta t^{\text{exp}}_{g_i}$. 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 is observed for an integer fraction p of the total exposures E_{g_i}, $p = 1,\ldots,E_{g_i}$, the duration of the observation and the resultant collected score will be as follows

$$\Delta t^{\text{obs}}_{g_i,p} = p\Delta t^{\text{exp}}_{g_i} \tag{2}$$

$$s_{g_i,p} = p\frac{s_{g_i}}{E_{g_i}} \tag{3}$$

The time required for the telescope to move from the concluding position of the object i during its j-th sky pass to the initial position of the object k during its l-th sky pass amounts to

$$\Delta t^{\text{slew}}_{i,j,k,l} = \frac{\cos^{-1}\left(\boldsymbol{r}^s_{i,j} \cdot \boldsymbol{r}^e_{k,l}\right)}{\omega_{\max}} \tag{4}$$

where

$$\boldsymbol{r}^a_{i,j} = \begin{cases} [\cos\theta^a_{i,j}\cos\psi^a_{i,j},\ \cos\theta^a_{i,j}\sin\psi^a_{i,j},\ \sin\theta^a_{i,j}]^\top & i \in \boldsymbol{l} \\ [\cos\theta_i\cos\psi_i,\ \cos\theta_i\sin\psi_i,\ \sin\theta_i]^\top & i \in \boldsymbol{g} \end{cases} \tag{5}$$

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 $\Delta t^{\text{prep}}_{\text{leo}}$, for LEO objects, and $\Delta t^{\text{prep}}_{\text{geo}}$, for GEO objects.

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 LEO objects) or the number of exposures taken p_i (for GEO objects):

$$\boldsymbol{\sigma} = \{(d_1, p_1), \ldots, (d_N, p_N)\} \tag{6}$$

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

$$t^s_{\boldsymbol{\sigma},i} = \begin{cases} t^s_{d_i,p_i} & d_i \in \boldsymbol{l} \\ t^e_{\boldsymbol{\sigma},i-1} + \Delta t^{\text{slew}}_{d_{i-1},p_{i-1},d_i,p_i} + \Delta t^{\text{prep}}_{\text{geo}} & d_i \in \boldsymbol{g} \end{cases} \tag{7}$$

$$t^e_{\boldsymbol{\sigma},i} = \begin{cases} t^e_{d_i,p_i} & d_i \in \boldsymbol{l} \\ t^s_{\boldsymbol{\sigma},i} + \Delta t^{\text{obs}}_{d_i,p_i} & d_i \in \boldsymbol{g} \end{cases} \tag{8}$$

and with a corresponding score:

$$s_{\boldsymbol{\sigma},i} = \begin{cases} s_{d_i} & d_i \in \boldsymbol{l} \\ s_{d_i,p_i} & d_i \in \boldsymbol{g} \end{cases} \tag{9}$$

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

$$s_{\boldsymbol{\sigma}} = \sum_{i=1}^{N} s_{\boldsymbol{\sigma},i} \tag{10}$$

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

$$\Delta t_{\text{obs},\boldsymbol{\sigma}} = \sum_{i=1}^{N} \Delta t^{\text{obs}}_{d_i,p_i} \tag{11}$$

$$\Delta t_{\text{tot},\boldsymbol{\sigma}} = t^{e}_{\boldsymbol{\sigma},N} - t_0 \tag{12}$$

Let us define with $\boldsymbol{d}_{\boldsymbol{\sigma}} = \{d_i\}|_{i=1,\ldots,N}$ 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 $\{1,\ldots,N\} \setminus i$ that refer to observations of the same object d_i:

$$\boldsymbol{k}_{\boldsymbol{\sigma},i} = \{k \in \{1,\ldots,N\} \setminus i : d_k = d_i \in \boldsymbol{d}_{\boldsymbol{\sigma}}\} \tag{13}$$

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

$$d_i \in \boldsymbol{l} \cup \boldsymbol{g}, \qquad i = 1,\ldots,N \tag{14}$$

$$p_i \in \mathbb{N}^+, \qquad i = 1,\ldots,N \tag{15}$$

$$d_i \neq d_{i+1}, \qquad i = 1,\ldots,N-1 \tag{16}$$

$$\boldsymbol{k}_{\boldsymbol{\sigma},i} = \emptyset, \qquad d_i \in \boldsymbol{l}, i = 1,\ldots,N \tag{17}$$

$$p_i \leq \begin{cases} P_{d_i} & d_i \in \boldsymbol{l} \\ E_{d_i} - \sum\limits_{\substack{k \in \boldsymbol{k}_{\boldsymbol{\sigma},i} \\ k<i}} p_k & d_i \in \boldsymbol{g} \end{cases}, \qquad i = 1,\ldots,N \tag{18}$$

$$t^{s}_{\boldsymbol{\sigma},i} \geq t^{e}_{\boldsymbol{\sigma},i-1} + \Delta t^{\text{slew}}_{d_{i-1},p_{i-1},d_i,p_i} + \Delta t^{\text{prep}}_{\text{leo}} \qquad d_i \in \boldsymbol{l}, i = 2,\ldots,N \tag{19}$$

$$\Delta t_{\text{tot},\boldsymbol{\sigma}} \leq \Delta t_{\text{max}} \tag{20}$$

So, the state space of the search problem is:

$$\mathscr{S} = \{\boldsymbol{\sigma} : \text{(6), (7)-(8), (13)-(20)}\} \tag{21}$$

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:

$$\boldsymbol{\sigma}_0 = \{(H,1)\} \tag{22}$$

$$t^{s}_{\boldsymbol{\sigma}_0} = t_0 \tag{23}$$

$$t^{e}_{\boldsymbol{\sigma}_0} = t_0 \tag{24}$$

$$s_{\boldsymbol{\sigma}_0} = 0 \tag{25}$$

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

$$\mathscr{C}(\boldsymbol{\sigma}) = \left\{\boldsymbol{\sigma}^{'} : \boldsymbol{\sigma}^{'} = \boldsymbol{\sigma} \cup (d_{N+1}, p_{N+1}), \boldsymbol{\sigma}^{'} \in \mathscr{S}\right\} \tag{26}$$

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

$$\boldsymbol{u}_{\boldsymbol{\sigma}} = \{(\tilde{d}_1, \tilde{p}_1), \ldots, (\tilde{d}_U, \tilde{p}_U)\} \tag{27}$$

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

$$\begin{aligned}
&\tilde{d}_i \in \boldsymbol{l} \cup \boldsymbol{g}, && i = 1, \ldots, U && (28)\\
&\tilde{d}_i \notin \boldsymbol{l} \cap \boldsymbol{d}_{\boldsymbol{\sigma}}, && i = 1, \ldots, U && (29)\\
&\tilde{d}_i \notin \{d_j \in \boldsymbol{g} \cap \boldsymbol{d}_{\boldsymbol{\sigma}} : p_{\max\{\boldsymbol{k}_{\boldsymbol{\sigma},j}\}} = E_{d_j}\}, && i = 1, \ldots, U && (30)\\
&\tilde{p}_i \in \mathbb{N}^+, && i = 1, \ldots, U && (31)\\
&\tilde{p}_i \leq P_{\tilde{d}_i}, && \tilde{d}_i \in \boldsymbol{l}, i = 1, \ldots, U && (32)\\
&\tilde{p}_i \neq \tilde{p}_k, && \tilde{d}_i = \tilde{d}_k \in \boldsymbol{l}, i \neq k = 1, \ldots, U && (33)\\
&\tilde{p}_i = \begin{cases} E_{\tilde{d}_i} & \tilde{d}_i \in \boldsymbol{g} \setminus (\boldsymbol{g} \cap \boldsymbol{d}_{\boldsymbol{\sigma}}) \\ E_{\tilde{d}_i} - \sum\limits_{k \in \boldsymbol{k}_{\boldsymbol{\sigma},i}} p_k & \tilde{d}_i \in \boldsymbol{g} \cap \boldsymbol{d}_{\boldsymbol{\sigma}} \end{cases}, && i = 1, \ldots, U && (34)\\
&t^s_{\tilde{d}_i, \tilde{p}_i} \geq t^e_{\boldsymbol{\sigma},N} + \Delta t^{\text{slew}}_{d_N, p_N, \tilde{d}_i, \tilde{p}_i} + \Delta t^{\text{prep}}_{\text{leo}}, && \tilde{d}_i \in \boldsymbol{l}, i = 1, \ldots, U && (35)
\end{aligned}$$

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

$$s_{\boldsymbol{u}_{\boldsymbol{\sigma}},i} = \begin{cases} s_{\tilde{d}_i} & \tilde{d}_i \in \boldsymbol{l} \\ s_{\tilde{d}_i, \tilde{p}_i} & \tilde{d}_i \in \boldsymbol{g} \end{cases} \tag{36}$$

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

$$\Delta t_{\text{left},\boldsymbol{\sigma}} = \Delta t_{\max} - \Delta t_{\text{tot},\boldsymbol{\sigma}} \tag{37}$$

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

$$\boldsymbol{u}_{\boldsymbol{\sigma}} = \emptyset \tag{38}$$

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

$$\mathscr{G} = \{\boldsymbol{\sigma} : \boldsymbol{\sigma} \in \mathscr{S}, \text{(27)-(35), (38)}\} \tag{39}$$

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

$$J_{\boldsymbol{\sigma}} = 10^3 s_{\boldsymbol{\sigma}} - \Delta t_{\text{tot},\boldsymbol{\sigma}} + 10^{-6} N \tag{40}$$

The three weights 10^3, 1, and 10^{-6} have been selected to keep the three contributions, that is, the score, total observation time, and number of objects observed, as separate as possible. The objective is to first rank the states by score, then, when the score is the same, by total observation time, and, when both the score and the observation time are equal, by number of observations. Indeed, the score $s_{\boldsymbol{\sigma}}$ is kept on the order of 10^0, $\Delta t_{\text{tot},\boldsymbol{\sigma}}$ on the order of 10^{-1} (in days), and N on the order of 10^2.

A goal state is optimal if no other goal state in $\mathscr{G}$ has a higher value of the objective function.

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

$$\begin{aligned}
&\max_{\boldsymbol{\Sigma}} J_{\boldsymbol{\sigma}_K} \\
\text{s.t. } &\boldsymbol{\sigma}_i \in \mathscr{S}, && i = 1, \ldots, K \\
&\boldsymbol{\sigma}_K \in \mathscr{G}, \\
&\boldsymbol{\sigma}_i \in \mathscr{C}(\boldsymbol{\sigma}_{i-1}), && i = 1, \ldots, K
\end{aligned} \tag{41}$$

3. SOLUTION APPROACH

In this section, the beam A*-search technique used to address the search problem in Eq. (41) is outlined. Diverse heuristics are also presented to accelerate the search process when confronted with either LEO 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. In this tree, each node corresponds to a problem state, and the connections between nodes represent state-to-state transitions. The search starts by generating successor nodes for the root node, which correspond 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 selects 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 serves 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 $f_{\boldsymbol{\sigma}}$. In problems aiming to maximize a state-dependent objective function, like the one examined in this paper, $f_{\boldsymbol{\sigma}}$ combines the state's objective function $J_{\boldsymbol{\sigma}}$ with a heuristic function $h_{\boldsymbol{\sigma}}$, estimating the potential enhancement in the objective function 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 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 $\boldsymbol{\sigma}$:

$$f_{\boldsymbol{\sigma}} = J_{\boldsymbol{\sigma}} + h_{\boldsymbol{\sigma}} \geq J_{\boldsymbol{\sigma}_g} \tag{42}$$

As an optimal algorithm, A* exhibits exponential growth in both time and space complexity when confronted with NP-hard problems. This characteristic can render A* impractical for managing large-scale problems. However, this limitation can be alleviated through the application of well-crafted heuristics, which serve to reduce the number of nodes that need to be expanded. The precision of the heuristic is a critical factor in A*; a heuristic that closely approximates the actual objective function improvement to the nearest goal node leads to a reduction in the number of expanded nodes. 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. In this paper, we introduce three admissible heuristics. These heuristics are exact solutions derived from relaxed versions of the TTP when focusing exclusively on either LEO or GEO objects or both kinds of objects. The forthcoming sections delve into the presentation of these heuristics.

3.2 LEO Heuristic

In situations where only observations of LEO objects remain ($\tilde{d}_i \in \boldsymbol{l}$, $i = 1,\dots,U$), 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 LEO 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:

$$t^{s}_{\tilde{d}_j,\tilde{p}_j} \geq t^{e}_{\tilde{d}_i,\tilde{p}_i} + \Delta t^{\text{slew}}_{\tilde{d}_i,\tilde{p}_i,\tilde{d}_j,\tilde{p}_j} + \Delta t^{\text{prep}}_{\text{leo}} \tag{43}$$

The corresponding arc weight will be

$$w_{ij} = t^{e}_{\tilde{d}_j,\tilde{p}_j} - t^{e}_{\tilde{d}_i,\tilde{p}_i} \tag{44}$$

The maximum path weight is instead

$$w_{\max} = \Delta t_{\text{left},\boldsymbol{\sigma}} \tag{45}$$

When exclusively addressing observations of LEO objects, the resulting graph forms a directed acyclic graph (DAG). An example of DAG is shown in Figure 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 arrangement of the graph stems from the temporal order of LEO object passes, as it is not possible to backtrack to a prior object pass and thus create a loop.

If all the objects possess equal scores, the OP problem on a DAG simplifies into the longest shorter path (LSP) problem. This combinatorial problem has previously been introduced by Federici et al. for the planning of 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 combined weight w_{lsp} that remains within a predetermined upper limit $w_{\max}$. The name given to 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 (or weight).

The LSP problem can be efficiently tackled using dynamic programming. Its time complexity is $O\left(n^2 q^2\right)$, 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 LEO objects 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

$$h^{\text{lsp}}_{\boldsymbol{\sigma}} = 10^3 q\bar{s} - w_{\text{lsp}} + 10^{-6} q \tag{46}$$

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

$$q \geq M \tag{47}$$

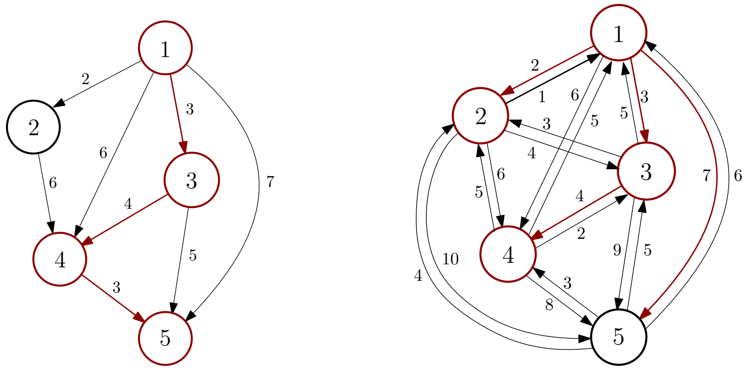
$$q\bar{s} \geq \sum_{m \in \boldsymbol{m}} s_{\boldsymbol{u_\sigma},m} \tag{48}$$

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.

3.3 GEO Heuristic

In scenarios where only observations of GEO objects remain ($\tilde{d_i} \in \boldsymbol{g}$, $i = 1,\ldots,U$), the TTP again reduces into an OP, as the optimal solution still involves observing each object just once for the whole exposure time, 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 Figure 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. This assumption holds



(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).

more reliably when just GEO objects are considered, as there are no mandatory waiting times between consecutive object observations. Therefore, it is much more likely that all the scheduled objects are actually observable in the remaining time. To achieve a more precise estimation of the total required observation time, the problem is posed as a minimum-weight Hamiltonian path problem.

This problem is a variant of the TSP that does not require closing the cycle. In this context, given a weighted graph comprising n nodes, labeled from 1 to n, connected by arcs with weights w_{ij} $(i, j = 1, \ldots, n)$, and a designated root node $r \in \{1, \ldots, n\}$, 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 $(i = (\tilde{d}_i, \tilde{p}_i), i = 1, \ldots, n)$, 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:

$$w_{ij} = \Delta t^{\text{slew}}_{\tilde{d}_i,\tilde{p}_i,\tilde{d}_j,\tilde{p}_j} + \Delta t^{\text{prep}}_{\text{geo}} + \Delta t^{\text{obs}}_{\tilde{d}_i,\tilde{p}_i} \tag{49}$$

Although the minimum-weight Hamiltonian path problem remains NP-hard, a recognized admissible heuristic called minimum spanning arborescence (MSA) can be utilized to establish a lower bound on the path weight. 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's total weight serves as a lower bound for the total weight of the minimum-weight Hamiltonian path. Indeed, the MSA's solution arises from a relaxed problem formulation, which assumes that more than two edges 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

$$h^{\text{msa}}_{\boldsymbol{\sigma}} = 10^3 \sum_{i=1}^{U} s_{\boldsymbol{u}_{\boldsymbol{\sigma}},i} - \min\{w_{\text{msa}}, \Delta t_{\text{left},\boldsymbol{\sigma}}\} + 10^{-6} U \tag{50}$$

3.4 Mixed LEO-GEO Heuristic

In the general scenario encompassing both LEO and GEO satellites, the LSP and MSA heuristics can be integrated to formulate a novel admissible heuristic. The underlying assumption of this composite heuristic is that the GEO satellites can be observed during the time intervals between LEO object passes. 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 LEO object observations, denoted as

$$\Delta t^{\text{wait}}_{\text{lsp}} = w_{\text{lsp}} - \sum_{i \in \boldsymbol{q}} \Delta t^{\text{obs}}_{\tilde{d}_i,\tilde{p}_i} \tag{51}$$

where $\boldsymbol{q} \subset \{1, \ldots, U\}$ is the set of indices identifying the 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 $G_{\boldsymbol{u}}$ as the number of GEO satellites within the unscheduled objects $\boldsymbol{u}_{\boldsymbol{\sigma}}$, the combined heuristic in state $\boldsymbol{\sigma}$ is

$$h_{\boldsymbol{\sigma}}^{\text{comb}} = 10^3 \left(q\bar{s} + \sum_{\tilde{d}_i \in \boldsymbol{g}} s_{\boldsymbol{u}_{\boldsymbol{\sigma}},i} \right) + \min \left\{ \Delta t_{\max,\boldsymbol{\sigma}}, w_{\text{lsp}} + \left[w_{\text{msa}} - \Delta t_{\text{lsp}}^{\text{wait}} \right]^+ \right\} + 10^{-6}(q + G_{\boldsymbol{u}}) \tag{52}$$

3.5 Search Space Pruning

Several pruning techniques have been incorporated to reduce the size of the search space 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.

3.5.1 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, $\boldsymbol{u}_{\boldsymbol{\sigma}_0}$, all the passes of the LEO objects that are not completely observable within the considered observation window, that is, the object-pass pairs $(\tilde{d}_i, \tilde{p}_i)$ for which the following condition is verified

$$\left(t^s_{\tilde{d}_i,\tilde{p}_i} < t_0 \right) \vee \left(t^e_{\tilde{d}_i,\tilde{p}_i} > t_0 + \Delta t_{\max} \right), \qquad \tilde{d}_i \in \boldsymbol{l}, i = 1, \ldots, U \tag{53}$$

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 if the corresponding states share the same score, total observation time, and list of objects observed. Furthermore, each object within these states 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 LEO objects, but for the final one, is observed in a different order or if the exposures of one or more GEO objects, occurring between the same LEO object passes, are taken over different numbers of successive observations.

3.5.2 Beam Search

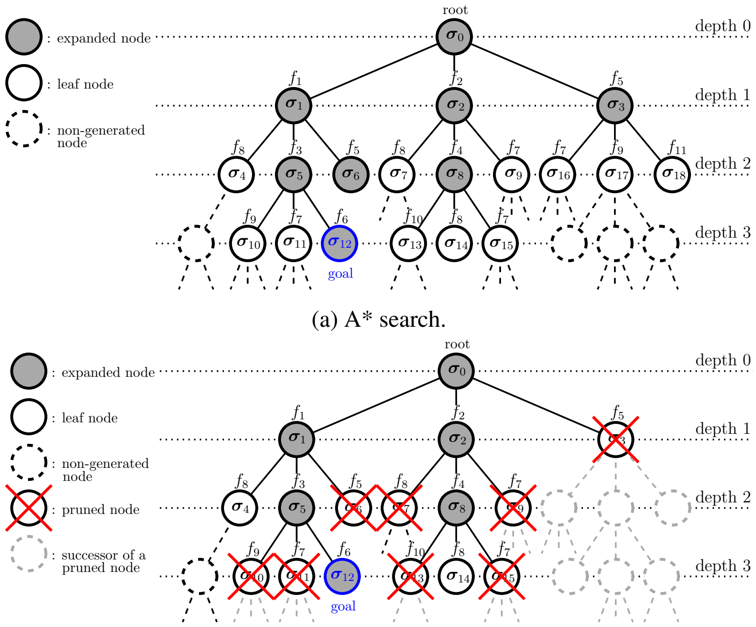
Despite the effectiveness of the A* search algorithm, it may never converge to the optimal solution when dealing with problem instances with extensive object lists or a combination of different orbital regimes (LEO and GEO) because of its high time and space complexity. To address these challenges, a sub-optimal variant of A*, named beam A*-search (BA*), has been devised. 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 $f_{\boldsymbol{\sigma}} > f_{\boldsymbol{\sigma}_w}$;

(i) 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

$$p_s = \frac{1}{1 + \frac{f_{\boldsymbol{\sigma}_w}}{f_{\boldsymbol{\sigma}}}} \tag{54}$$



(a) A* search.

(b) BA* search.

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

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,[15] 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 LEO 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 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 LEO-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 the BA* search algorithm when considering both LEO and GEO objects together 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 under investigation 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.

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
ω_{max}	2	deg/s
θ_{min}	10	deg

Table 2: Observation and object data.

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

The observation session starts on July 24th, 2023, at 8:30 pm, according to Arizona time, and is extended for a total duration of 8.5 hours. During this nighttime session, the focus is on observing satellites from the Globalstar and Iridium constellations in LEO, as well as satellites from the Intelsat and Galaxy constellations in GEO.

Relevant parameters such as the epoch, azimuth, and altitude of the objects at the beginning and conclusion of their passes over the telescope's location have been precomputed. This data has been generated by forward-propagating their two-Line elements (TLEs), sourced from the Norad catalog via SpaceTrack.com, employing the SGP4 model.

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

Detailed specifics about the initiation 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 prior to capturing exposures of LEO and GEO satellites, are outlined in Table 2.

The implementation of the BA* search algorithm has been realized in C++, and the execution was conducted on a workstation housing an AMD Ryzen 9 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 Hyperparameter Tuning

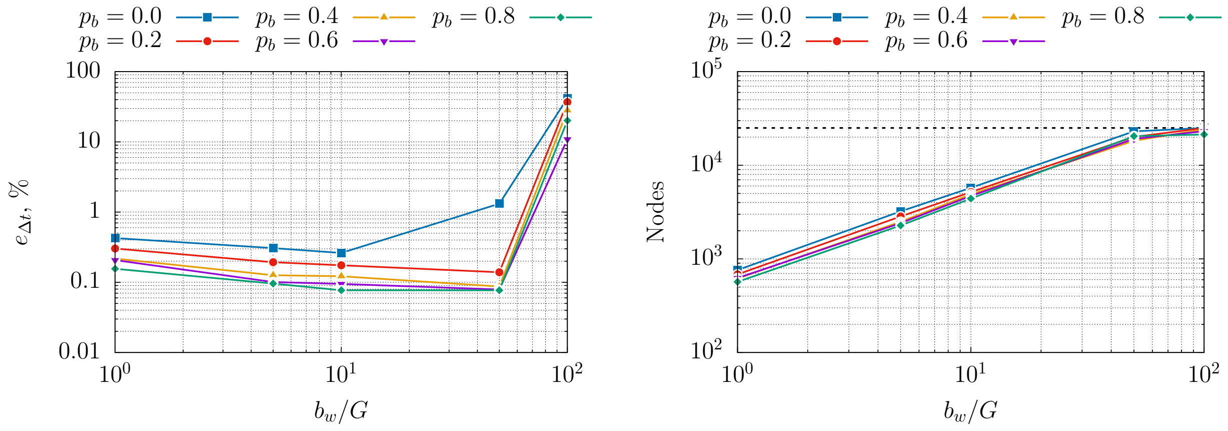
First, the effect of the two hyperparameters of BA*, namely the beam width b_w and the inclusion probability p_b, on the algorithm performance has been analyzed by considering two instances of the TTP involving either all the GEO satellites or the LEO satellites, a maximum observation time $\Delta t_{\text{max}}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

Figures 3 and 4 show the trends of the algorithm performance measure (Fig. 3a and 4a) and average number of explored nodes (Fig. 3b and 4b) 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 realized to mitigate the stochastic nature of the algorithm and collect statistics. The maximum number of explored nodes has been fixed to $D_{\text{max}} = 25000$. Two different performance measures have been used when scheduling GEO or LEO objects, respectively. Let us name with $s^\star$ and $\Delta t_{\text{tot}}^\star$ 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:

$$e_{\Delta t} = \frac{|\Delta t_{\text{tot}}^\star - \Delta t_{\text{tot},\boldsymbol{\sigma}}|}{\Delta t_{\text{tot}}^\star} \tag{55}$$

being the score (almost) always the same between the A* and BA* solutions. Conversely, with LEO objects, the performance measure is the average relative error in the cumulative score with respect to the optimal solution:

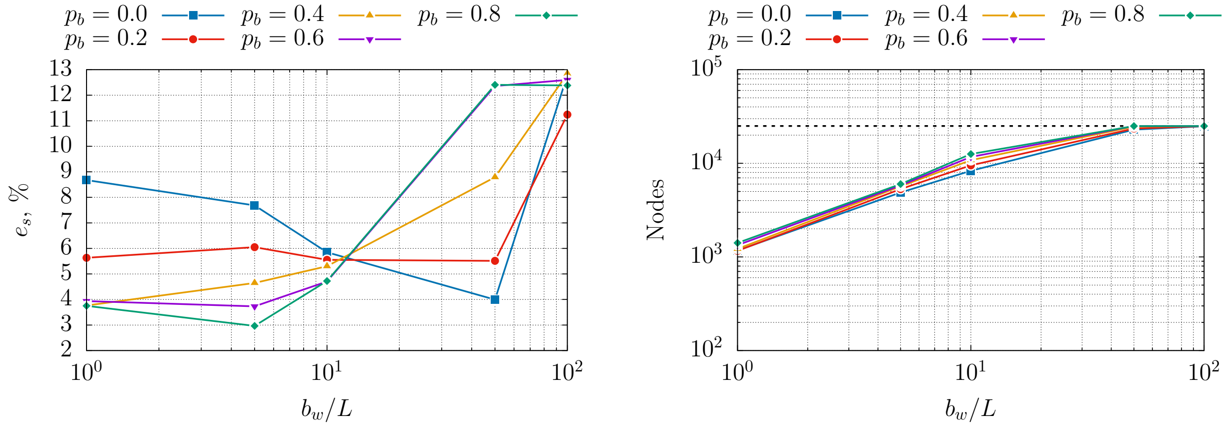
$$e_s = \frac{|s^\star - s_{\boldsymbol{\sigma}}|}{s^\star} \tag{56}$$



(a) Success rate.

(b) Number of explored nodes.

Fig. 3: Performance analysis of BA* for GEO satellites scheduling for different beam widths b_w and inclusion probabilities p_b.



(a) Success rate.

(b) Number of explored nodes.

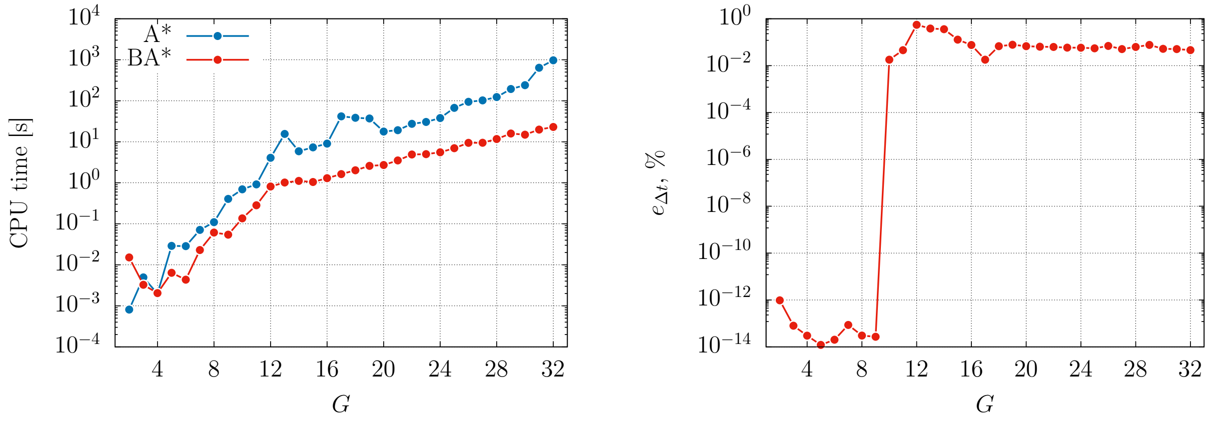
Fig. 4: Performance analysis of BA* for LEO satellites scheduling for different beam widths b_w and inclusion probabilities p_b.

Figures 3b and 4b 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. 3a and 4a, 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. For lower values of the beam width ($b_w \leq 50G$ and $b_w \leq 50L$ 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 seconds of observation time. This level of accuracy is achieved when utilizing

$p_b = 0.8$ with either $b_w = 5G$ or $b_w = 10G$. 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 ($b_w = 50L$) 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 $b_w = 5L$. For this reason, the analyses in the following sections have been conducted using this specific configuration of the algorithm.

4.3 Algorithm Performance Analysis

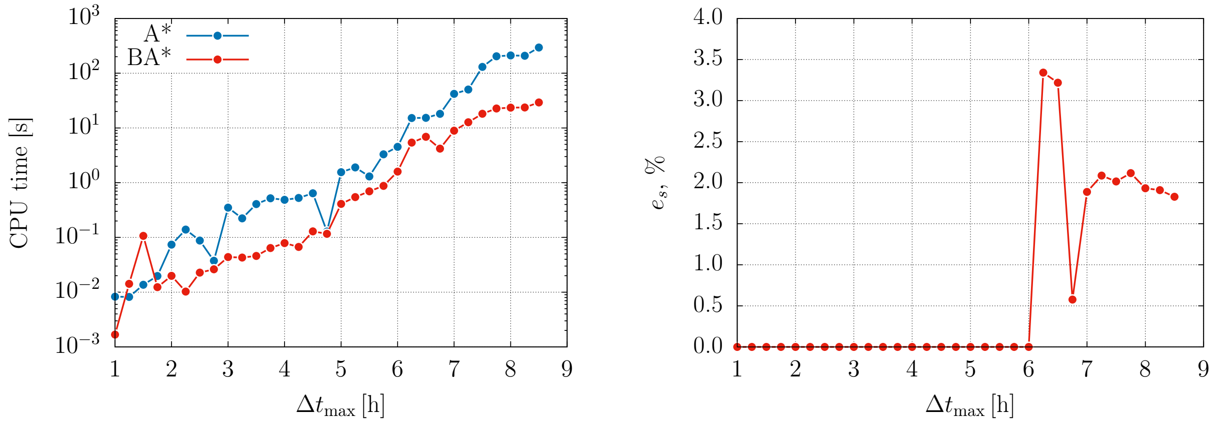
The performance of the best BA* configuration found so far has been then systematically compared against standard A* on increasingly harder instances of the telescope tasking problem with only GEO or LEO satellites with random scores in the interval $[1,3]$. For each problem instance, the best solution over 50 runs of BA* has been used for comparison with A* to reduce performance fluctuations due to the stochastic nature of the algorithm.



(a) CPU time vs GEO satellites.

(b) Solution error vs GEO satellites.

Fig. 5: Performance comparison between BA* and A* for GEO satellites scheduling.



(a) CPU time vs observation time.

(b) Solution error vs observation time.

Fig. 6: Performance comparison between BA* and A* for LEO satellites scheduling.

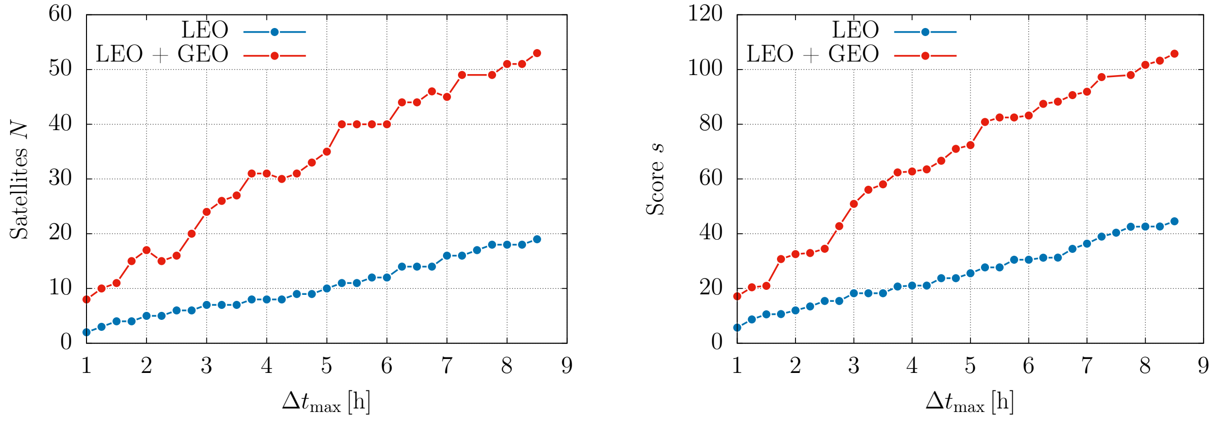
Figure 5 shows the trends of the CPU time (Fig. 5a) and of the error on the total observation time (Fig. 5b) obtained with BA* and A* for GEO object scheduling by varying the pool of GEO satellites G from 2 to 32 and keeping the total observation time fixed to $\Delta t_{\max} = 8.5\,\mathrm{h}$. Figure 6, instead, shows the trends of the CPU time (Fig. 6a) and of the error on the cumulative score (Fig. 6b) that are obtained with BA* and A* for LEO object scheduling by varying the total observation time Δt_{max} from 1 hour to 8.5 hours, and always considering all the $L = 90$ satellites.

In both scenarios, it's evident that BA* consistently reduces the overall time needed to reach a goal state (Figs. 5a and 6a). 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 minutes with A* to 23 seconds with BA*. For LEO satellites, the time drops from 5 minutes with A* to less than 30 seconds with BA*, which corresponds to a reduction of almost one order of magnitude.

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. 5b and 6b, 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.

4.4 Optimal Observation Schedules

Eventually, this section presents and discusses the optimal observation schedules obtained with the BA* search algorithm when considering both GEO and LEO satellites together.



(a) Number of satellites vs observation time.

(b) Total score vs observation time.

Fig. 7: Optimal solutions by varying the observation time for LEO-only and LEO+GEO object sets.

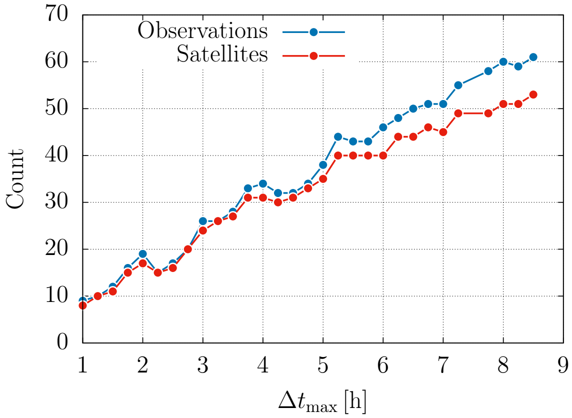


Fig. 8: Number of observations and satellites observed by varying the observation time for the LEO+GEO object set.

Figure 7 shows the trends of the number of observed satellites (Fig. 7a) and the total score (Fig. 7b) obtained with BA* by varying the observation time Δt_{max} from 1 hour to 8.5 hours and either considering just the LEO objects or both the LEO and GEO objects together. In both cases, the object scores are sampled randomly in the interval $[1,3]$.

As expected, the overall score increases monotonically with the observation time. However, this is not true for the count of observed satellites, 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. When objects in both orbital regimes are considered, 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 hours, the telescope is able to observe a total of 53 objects, which include all 33 GEO satellites plus 20 LEO satellites. Figure 8 shows the number of observations and number of observed satellites for the mixed LEO-GEO scenario for different observation times Δt_{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 hours, 61 observations of 53 distinct satellites), as opposed to the LEO-only 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 (LEO, 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's evaluation function, which contains information coming both from the problem objective function and A*'s heuristic.

First, the main hyperparameters of the BA* search method, namely the beam width and the inclusion probability, have been properly tuned in order 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 5 times the number of scheduled objects yield the best results both 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 3.5% lower in the LEO-only case, BA* is able to 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 the scalability of the current approach in managing tasking scenarios involving multiple telescopes and a broader spectrum of celestial objects. This extension could encompass objects situated in either medium Earth orbit (MEO), geostationary transfer orbit (GTO), or beyond the geostationary belt, such as those within the cislunar space region. Furthermore, an exploration of alternative optimization algorithms, such as genetic algorithms or ant colony optimization, and their comparison against the BA* algorithm could serve as a means to establish benchmarks and gain valuable insights for upcoming investigations within this domain

REFERENCES

[1] Carolin Frueh, Hauke Fielder, and Johannes Herzog. Heuristic and optimized sensor tasking observation strategies with exemplification for geosynchronous objects. *Journal of Guidance, Control, and Dynamics*, 41(5):1036–1048, 2018.

[2] Zachary Sunberg, Suman Chakravorty, and Richard Scott Erwin. Information space receding horizon control for multisensor tasking problems. *IEEE Transactions on Cybernetics*, 46(6):1325–1336, 2016.

[3] Samuel Fedeler and Marcus Holzinger. *Monte Carlo Tree Search Methods for Telescope Tasking*. 2020.

[4] Peng Mun Siew, Daniel Jang, Thomas G Roberts, Richard Linares, and Justin Fletcher. 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] Peng Mun Siew and Richard Linares. Optimal tasking of ground-based sensors for space situational awareness using deep reinforcement learning. *Sensors*, 22(20), 2022.

[6] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[7] Carnegie-Mellon University.Computer Science Dept. Speech understanding systems: summary of results of the five-year research effort at Carnegie-Mellon University. 2015.

[8] Dan Klein and Christopher D Manning. 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*, pages 119–126, 2003.

[9] Bo-bo Meng and Xiaoguang Gao. UAV path planning based on bidirectional sparse A* search algorithm. In *2010 International Conference on Intelligent Computation Technology and Automation*, volume 3, pages 1106–1109. IEEE, 2010.

[10] Xiao Cui and Hao Shi. A*-based pathfinding in modern computer games. *International Journal of Computer Science and Network Security*, 11(1):125–130, 2011.

[11] Aldy Gunawan, Hoong Chuin Lau, and Pieter Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332, 2016.

[12] Lorenzo Federici, Alessandro Zavoli, and Guido Colasurdo. On the use of a* search for active debris removal mission planning. *Journal of Space Safety Engineering*, 8(3):245–255, 2021.

[13] Jack Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards B*, 71(4):233–240, 1967.

[14] Alessandro Zavoli, Lorenzo Federici, Boris Benedikter, Lorenzo Casalino, and Guido Colasurdo. Gtoc x: Solution approach of team sapienza-polito. *Advances in the Astronautical Sciences*, 171:3327–3342, 2019.

[15] Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.