Dynamic Scheduling for Space-Based Space Situational Awareness

Xiang Li[a]*, Stefan Thorsteinson[b]

[a] *Defence Research and Development Canada, xiang.li@drdc-rddc.gc.ca*
[b] *Defence Research and Development Canada, stefan.thorsteinson@drdc-rddc.gc.ca*
* Corresponding Author

Abstract

Space situational awareness (SSA) or space domain awareness is becoming increasingly challenging as a growing number of government and private users have had increased access to space and space services. Space-based SSA provides unobstructed observation without interferences and distortions imposed by weather and atmospheric conditions that can significantly impact ground stations. However, optical sensors with a narrow field of view must slew from object to object in various orbits at different distances. This leads to sensor tasking challenges that have been traditionally solved as a combinatorial optimization problem for ground-based sensors. With the recent advancement in space-based capabilities, there is a growing interest in tracking, monitoring and characterizing space-objects on a recurring basis using space-based sensors. As a result, Canada is developing a research satellite mission called Redwing that can perform resolved optical imaging for close encounters with Low Earth Orbit (LEO) Resident Space Objects (RSOs) and gather metric and photometric data for both LEO and Deep-space RSOs. Space-based SSA presents a distinct scheduling challenge with limited onboard resources and link budget when dealing with increased dynamic behaviour of objects in different orbits, as opposed to ground-based SSA. Consequently, we developed a multi-purpose scheduler capable of managing the challenging tasks imposed on space-based sensors seen on the Redwing mission. In this paper, we address the dynamic scheduling problem for space-based SSA with the added constraints of observation availability, ad-hoc tasks at fixed intervals and revisit requirements taken into consideration. The performance of two greedy-based solvers is compared using the cumulative number of objects observed as well as the total computation time. Several scheduling scenarios are also simulated and discussed.

Keywords: space situational awareness, combinatorial optimization, satellite scheduling, Redwing Mission

1. Introduction

Canada has developed space-based SSA capabilities, namely the Sapphire satellite that operationally contributes to the US Space Surveillance Network. A dual-use optical SSA and astronomy research satellite, NEOSSat [1] has been undertaking space-based technical assessments for Defence Research and Development Canada (DRDC) and international defence scientific communities for over twelve years. Historically, GEO belt surveillance has been prioritized as GEO satellites host critical assets such as weather, communication and navigation systems. The rapid expansion of LEO constellations demands more robust monitoring of this regime, where satellites and space debris are more numerous and dynamically positioned. To advance and extend Canada's future SSA capabilities with the changing and increasingly contested orbital environment, DRDC is developing a research satellite with advanced optical sensors and increased communication link. To be launched in late 2026, Redwing is an agile satellite with three axes of control, allowing the satellite to observe different targets in three-dimensional space in contrast to Earth-Observing Satellites (EOS) which focus on targets along-track across a strip of area on the earth surface. Space-based observations present unique attitude maneuvering behaviors compared to Earth-based observations. Existing scheduling methods [2]-[6] assume a limited problem space by either applying grid-based viewing angles or selecting across imaging windows blocks of time that represent a continuous observation period—rather than examining every discrete second. While the simplified search space may work with GEO objects due to their near-static appearance or with EOS observations, applying the same methods to LEO RSOs are more challenging because of the high dynamics involved.

SSA sensors in orbit are continuously slewing between targets, with target detectability being time-dependent. Multiple targets may become detectable simultaneously, leading to overlapping visibility windows and conflicting task priorities as can be seen in Figure 1. Each target requires a finite number of observations per day, and redundant observations strain both on-board memory and link budgets. Additionally, the scheduler must be able to incorporate user-defined tasks at fixed time periods which further congests the often already crowded problem space. Consequently, an adaptive scheduling system is essential to maximize the number of observations while adhering to these constraints. Three scenarios reflecting a modern satellite's, such as Redwing, day-in-the-life are experimented with two greedy-based optimization methods.

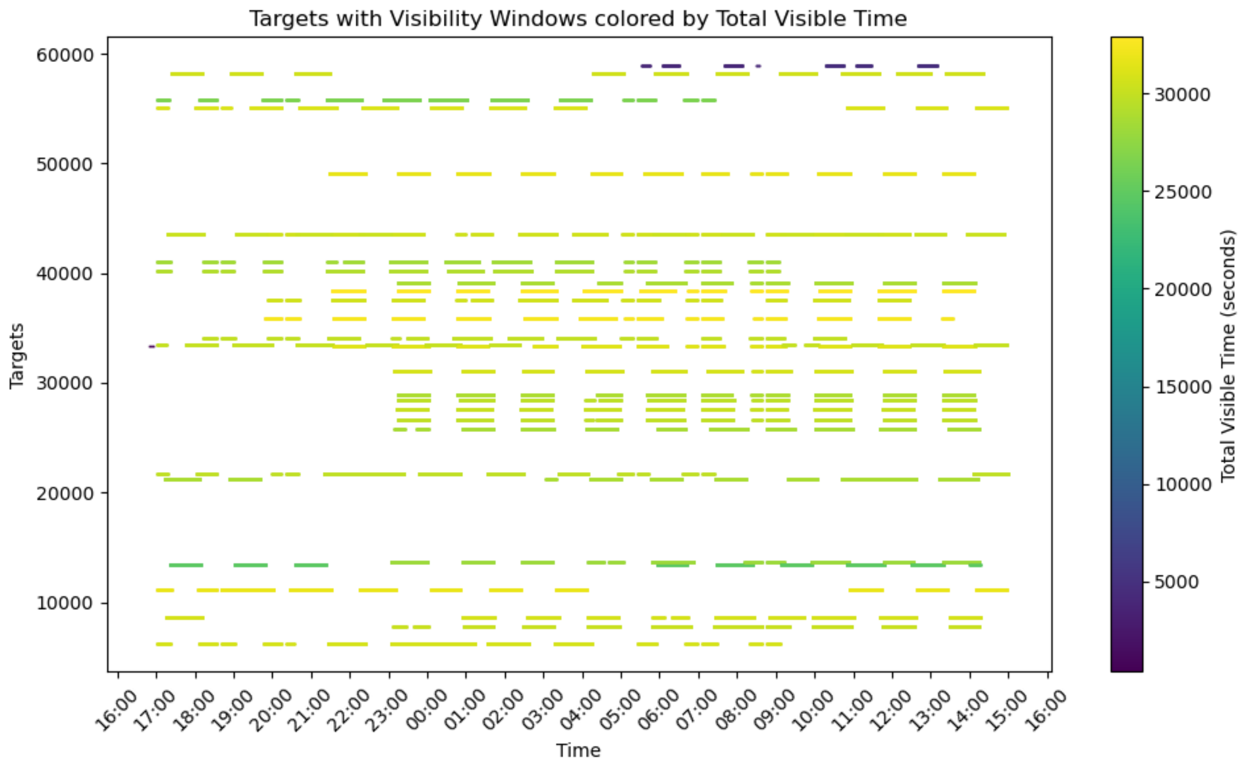

Figure 1 - Scenario 1 Visibility windows for each target across a 24hr period

1.1 Literature Review

The scheduling of a single space-based SSA sensor is as demanding if not more than that of multi-satellite EOS or agile earth observation satellite (AEOS). While scheduling of EOS and AEOS may also face overlapped visibility windows for multiple tasks, space-based SSA sensor is challenged with significantly more overlapped tasks with RSOs in the congested LEO environment and GEO belt. Additionally, a space-based sensor may be tasked to track and image hundreds of targets on a given day, which amounts to thousands of visibility windows in total. The satellite's on-board optical telescope is limited with relatively low narrow field of views. Therefore, the combinatory scheduling problem needs to take slew, setup and imaging times between consecutive imaging tasks into considerations.

In [2] a proposed priority-based inter linear programming (ILP) for multi-AEOSs scheduling solved by CPLEX, which failed to produce a feasible solution for observations of ground targets for more than 200 tasks. Machine learning techniques were compared with classical optimization methods in [3], for SSA sensor tasking with a ground-based sensor for catalogue maintenance tasks for geosynchronous RSOs. Pointing directions are discretized into grid fields which simplify the search space. Good computation speed was observed for the proposed Machine Learning methods. However, these approaches require significant training data and careful tuning to avoid overfitting, and their decision-making processes can be less transparent than classical methods.

A deep reinforcement learning (DRL) was applied to space-based sensor tasking in [4], demonstrating that DRL can learn effective policies for observation scheduling of geosynchronous RSOs in dynamic environments without explicit programming of the task. However, the use of discretized Field of View (FOV) grids to simplify the action space for modeling is a limitation, as it does not fully leverage the enhanced agility of modern SSA satellites. In addition, the approach requires extensive computational resources and careful hyperparameter tuning, and its training process can be unstable or sample inefficient. DRL was also leveraged in [5] to optimize the tasking of ground-based sensors, showcasing the algorithm's ability to make near-optimal decisions in near-real time under uncertainty. However, the study simplifies the problem to a small number of observations across non-overlapping patches of the GEO belt. While this method shows promise in simulation, its performance in real-world applications may be limited by practical constraints and the complexity of space-based observations over overlapping visibility windows from RSOs in both LEO and the GEO belt. Also, some studies [6] discretize the field of regard of SSA sensor into a grid of viewing directions covering potentially multiple targets across a strip of the sky. In this paper, we assume a point-based imaging model where the satellite observes a single target at a given time.

Table 1 Acronyms

Acronyms	*Definition*
i, j	target index, i,j=1,2,……,J
a	Ad-hoc task a = 1,2,……,A
T_j	visibility time window of target j
st	starting time of a task
et	end time of a task
t	scheduled moment, t∈{st,st+1,...,et}
d	setup duration between observations
Δ_{ij}	transition time from target j to target i
c_t	cost of schedule at scheduled moment t
n_j	total number of tracks required for target j, n=1,2,…,N
α, β, γ	Weights for transition time, gap time and target availability, respectively

2. Material and methods

Unlike EOS, SSA observations for the same target can often take place consecutively until the required number of tracks have been achieved. During SSA missions, the scheduler must also be able to block off a period of time for user-defined activities including but not limited to imaging, ground station and intersatellite communications (both uplinks and downlinks) without sacrificing performance. Therefore, the reason for using greedy iterative search is threefold 1) dynamically updating the total availability of targets 2) ensuring smooth transition to ad-hoc activities at a specific time period determined by a start and end time, figuratively defined by a “goal post” as no other tasks can be scheduled during this period 3) efficiently handling a mixed observations of both LEO and GEO objects by searching through each discrete second. Several assumptions are made in this paper. For instance, the slew speed for manoeuvring the camera is fixed at 1.5 m/s. The buffer time between consecutive observations is assumed to be 2 minutes, with 1 minute allocated for imaging and 1 minute for stabilization. However, the proposed algorithms can iteratively adapt to varying transition speeds and buffer times without pre-computation, which can become computationally intensive as the observation space expands.

2.1 Cost Function

The exact cost of observing a target depends on both its position and the time required for the SSA sensor to slew from its current location. Consequently, the transition cost, Δ_{ij}, is highly time-dependent. From a global optimization perspective, the total availability or visibility time for each target varies based on their relative positions to the SSA sensor. The gap time between consecutive observations is also crucial, as it prioritizes tasks that can be scheduled sooner and penalizes those that can only be viewed after a long interval. The gap time is defined by the subtraction of end observation time of task i and the starting observation time of task j. An indicator to reflect target’s total availability is defined by the total visibility time and the remaining number of requested tracks. The visibility time per track is dynamically updated after each scheduled action. The cost function c_{ij} between the current and the next positions i and j is defined as follows:

$$c_t = \alpha\Delta_{ij} \cdot \beta(et_i - st_j) \cdot \gamma \frac{\sum_j v_j}{n_j} \qquad (1)$$

Route selection decisions are made by a cost function, which is formulated as the minimization of a set of weighted indicators in (1). Let y_{jt} be the binary decision variable, if starting time t is assigned for the observation of target j, and zero otherwise, with discrete starting time $t \in \{st_j, st_j + 1, \ldots, et_j\}$ of target $j = 1,2, \ldots, J$. Additionally, each possible job is associated with a cost c_j.

$$y_{jt} \begin{cases} 1, \ if\ starting\ time\ t\ is\ assigned\ for\ the\ observation\ of\ target\ j \\ 0, \ otherwise \end{cases} \tag{2}$$

2.2 Objectives and Constraints

We introduce the objectives and constraints of the unique scheduling problem for a space-based SSA telescope.

$$Maximize \sum_{j \in J} \sum_{t=st_j}^{et_j} \frac{1}{c_t} y_{jt} + \sum_{a \in A} y_{at} \tag{3}$$

subject to:

$$st_i > st_j + d_{jt} + \Delta_{ij}, \forall i, j \in J \tag{4}$$

$$\sum_j \sum_k y_{jk} = n_j \tag{5}$$

$$st_a = T_a^{start}, st_a + d_a = T_a^{end}, y_a = 1 \quad \forall a \in A \tag{6}$$

The aim of the objective function in (3) is to maximize the number of observations for both normal surveillance and ad-hoc tasks by minimizing the total cost of the scheduled tasks. Equation (4) imposes the transition time constraints; suppose current scheduled job is j (with observation finishing at $st_j + d_{jt}$. For any candidate job j to follow, its start time much account for both the observation duration and the slew (or transition) time between j and i. (5) hard constraint that ensures that each target is visited for exactly the required number of times. (6) is another hard constraint so that ad-hoc tasks are strictly scheduled at their fixed times.

2.3 Graph Representation

The scheduling problem formulated above is represented using weighted graph. Each possible observation starting point can be defined by a vertex. Vertices are connected by edges with associated costs to facilitate search. The graph is acyclic as no closed circles are possible as the search moves forward in time. The graph is also dynamically updated as equation (5) is a hard constraint so when the requested number of tracks of an RSO are satisfied, all its relating vertices are dropped.

2.4 Dynamic greedy search

We first propose a dynamic greedy search scheduler that can adapt to the mixed space-based observations of both the LEO and the GEO belts. Detailed implementation can be found in Algorithm 1. Unlike many scheduling methods that limit the search problem to selecting a period of time or observation windows, this approach aims to identify the precise starting point to the second, regardless of the observation window it resides in. The first iteration globally searches for the most optimal starting point, determined by the cost function. Subsequently, vertices with a starting time before this point are eliminated from the graph, refining and reducing the search space for the next starting point.

2.5 Rolling horizon greedy search

A rolling horizon greedy search is also proposed to facilitate more efficient search. The implementation of this approach is similar to the dynamic greedy search depicted in Algorithm 1 except that the search space is divided into smaller windows defined by the setup duration. Therefore, the graph is formulated iteratively with a smaller number of vertices. Besides computation speed, one main advantage of this is its adaptability to ad-hoc tasks with fixed time intervals thanks to its more localized search space compared to the dynamic search algorithm. Both approaches can be well represented using acyclic graphs and dynamically update target availability at each iteration. The search across smaller time intervals results in a smoother transition between ad-hoc and normal tasks.

Algorithm 1: Dynamic greedy search with weighted heuristic

Input:

- Graph with nodes representing RSOs and edges representing possible transitions
- st_i, et_i start and end time
- setup time for ad-hoc task or between i and j: d_a or d_{ij}
- Weights: α, β, γ
- Current telescope position p_j

Output:

- Sequence of selected nodes with updated observation times

---

while $st_i < et_J$ **do**
  if Graph.nodes == $\emptyset$ **then**
    break
  for each edge (u_i, v_j) **in** Graph **do**
    Compute $\Delta_{ij} = haversine(p_i, p_j)$
    Assign weight $c_t(u_i, v_j)$ to edges (u_i, v_j) using Equation (1)
  end for
  Select next node such that:
    $et_j = argmin\{v \in Graph\} c_t(u_i, v_j)$
    Check for ad-hoc tasks
    if $st_a < et_j + d_a$ then
      $et_j = st_a$
      $st_i = et_a$
      $p_i = p_a$
    else if no ad-hoc task not found
      update $st_i = et_j$
      Set et_j += d_{ij}
      Set $p_i = p_j$ as telescope's new position has changed
    end if
  Update number of tracks: $n_j -= 1$
  Update target availability $\frac{\sum_j V_j}{n_j}$
  Remove nodes from Graph that no longer satisfy timing or observation constraints
end while

3. Experiments

Both proposed methods are evaluated against three different scenarios that properly reflect a day-in-the-life of a modern space-based SSA satellite. All TLEs are propagated using SGP4 while Redwing's orbit is modelled as sun-synchronous at 575 km altitude. For each scenario, all valid access windows between Redwing and each target are pre-computed using various constraints: line of sight, solar elongation above 45 degrees, Earth grazing angle above 10 degrees, and a maximum relative angular velocity of 225 arc-seconds per second. The simulations are run in the Intel Xeon Gold 5118 processor. Table 1 below addresses the performance of the two different methods under different scenarios. For simplicity, we use an averaged slew time of 1.5 degrees per second, with plans to model slew timing dynamics once Redwing's performance is measured on orbit. We also define 1 minute for both the set-up time (to match each RSO's angular rate) and the observation time for each task.

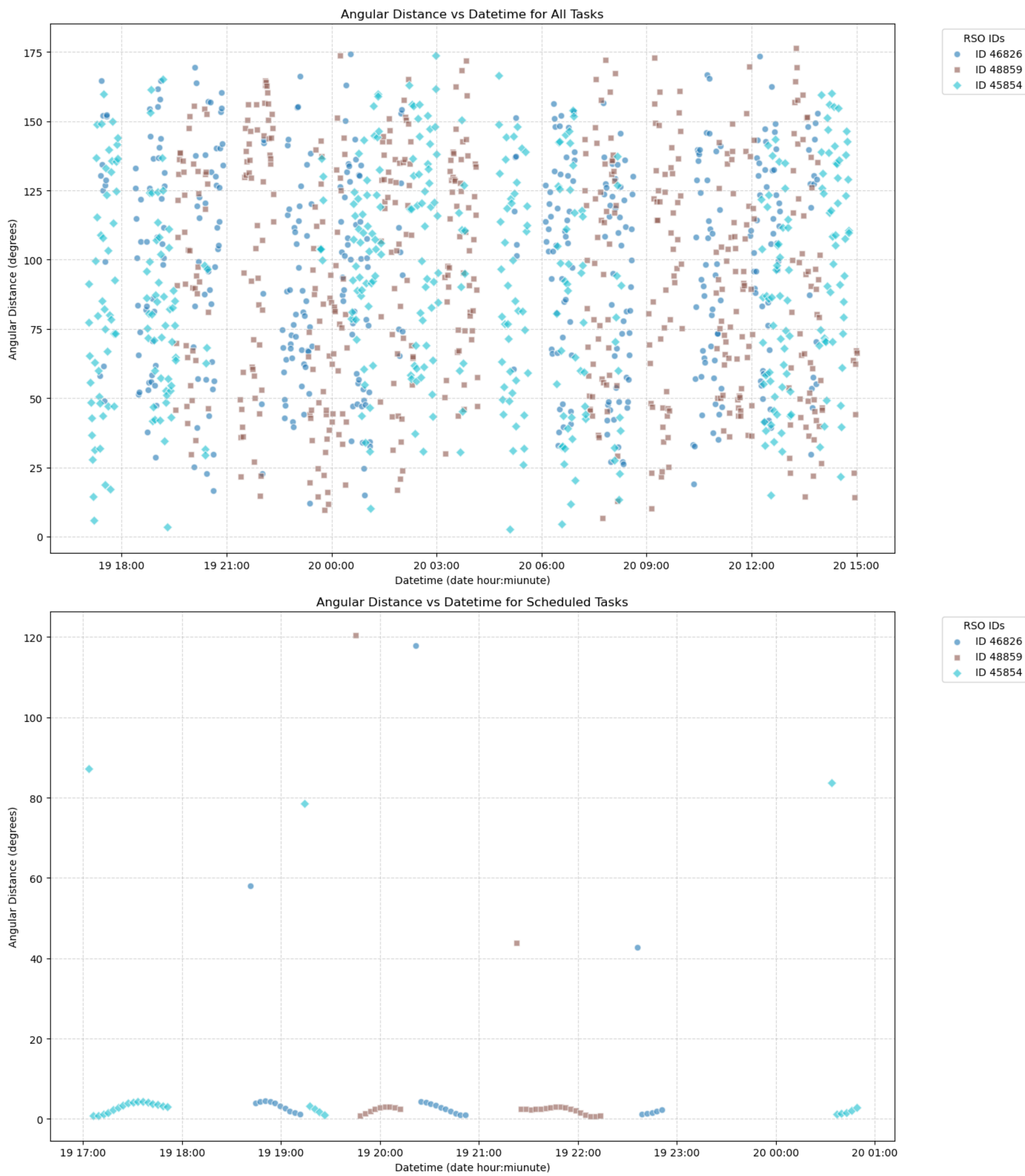

Figure 2 - Angular distance vs time for all targets (top) and angular distance vs scheduled tasks (bottom)

For Scenario 1, a TLE catalogue of the most recent 300 geosynchronous RSOs has been used. This represents the traditional space catalogue maintenance mode that optical SSA sensors have been utilized in the past. Figure 1 depicts the crowded observation space with the visibility window of each target highlighted with their total visibility time. Depending on the specific interests of the objects, they require a minimum of 1 track for catalogue maintenance tasks and a maximum of 7 tracks for RSO monitoring within a 24-hour window.

Scenario 2 aims to test the scheduler's ability of tracking GPS satellites consistently for persistent monitoring of a given RSO as well as sensor metric calibration. Four GPS satellites were selected with a maximal amount of 84 tracks per satellite, it is expected that the scheduler maintain tracking on a single object as long as possible before breaking to the next available GPS satellite. This is the only scenario with more tracks requested than the total number of visibility windows. The selection of mid Earth orbit GPS satellites is commensurate with acquiring sensor calibration data. In this scenario it is desirable to have the schedule optimized to stay on a visible window as long as possible before switching to the next available RSO. Two goal posts with a total of 208mins were introduced to facilitate user-defined tasks that must be executed at a fixed time. Figure 2 shows that the greedy solver successfully maintained its focus on the target for at least 4 consecutive observations for each satellite.

For Scenario 3, 100 LEO and 289 GEO RSOs were requested for 1 track each across a 24hr window. This scenario stresses the needs for a scheduler to meet the SSA sensor's capabilities imaging long duration, high availability GEO RSOs mixed with very short duration LEO RSO observing windows. In this scenario 389 tracks must be selected from 4951 highly overlapped visibility windows. The sheer number of visibility windows resulted in prolonged computation times for the dynamic greedy solver, as it had to iterate through a large number of them during the initial iterations before the search space reduced to a manageable size. The rolling-horizon greedy solver significantly improved the computation time by segmentizing the search space into multiple manageable pieces.

Table 1 - performance comparison

Scenario	Algorithm	Ad-hoc task duration (minutes)	Total tasks requested	Scheduled tasks	Total slew time (seconds)	Total gap time (hours)	Computation time (seconds)
3	Dynamic greedy	248	389	327	3068	0.55	1687
3	Rolling-horizon	248	389	389	4187	4.5	5.9
2	Dynamic greedy	7	84	84	547	4	3
2	Rolling-horizon	7	84	84	512	6	2
1	Dynamic greedy	7	76	76	824	0.7	26
1	Rolling-horizon	7	76	76	842	0.8	2

4. Conclusions

A dynamic greedy algorithm and a roll-horizon approach are introduced in this paper to solve a combinatorial optimization problem for the scheduling of modern space-based SSA such as the Redwing satellite. Three drastically different scenarios were used to represent a day-in-the-life space-based surveillance of both GEO and LEO RSOs. The iterative nature of the algorithms allows for fast computation time and are not demanding on the hardware. Overall, both the dynamic greedy solver and the roll-horizon variation adapt well to the changing priority of targets as their total availability can be dynamically updated. The rolling-horizon excels in more demanding scenario with thousands of visibility windows and hours of fixed ad-hoc tasks, however, the dynamic greedy approach has access to a larger search space that result in a route with less gap time and slew time in most scenarios.

References

[1] R.L. Scott, S. Thorsteinson, Key Findings from the NEOSSat Space-Based SSA Microsatellite Mission, AMOS Technical Conference 2018, Maui Economic Development Board, Kihei, HI
[2] R. Xu, H. Chen, X. Liang, and H. Wang, "Priority-based constructive algorithms for scheduling agile earth observation satellites with total priority maximization," Expert Systems with Applications, vol. 51, pp. 195–206, Jun. 2016, doi: 10.1016/j.eswa.2015.12.039.
[3] B. D. Little and C. Frueh, "SSA Sensor Tasking: Comparison of Machine Learning with Classical Optimization Methods," Advanced Maui Optical and Space Surveillance Technologies (AMOS), 2018.

[4] T. G. Roberts, P. M. Siew, D. Jang, and R. Linares, "A Deep Reinforcement Learning Application to Space-based Sensor Tasking for Space Situational Awareness," Advanced Maui Optical and Space Surveillance Technologies (AMOS), 2021
[5] P.M. Siew; R. Linares Optimal Tasking of Ground-Based Sensors for Space Situational Awareness Using Deep Reinforcement Learning. Sensors 2022, 22, 7847. https://doi.org/10.3390/s22207847
[6] M.A. Stephenson, H Schaub, Optimal Target Sequencing In The Agile Earth-Observing Satellite Scheduling Problem Using Learned Dynamics, Journal of Spacecraft and Rockets, 2024