DOI: 10.13009/EUCASS2025-338
11TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)

# Routing optimization for On-Orbit-Servicing using high-thrust propulsion

*Claudio Ciccotti*[⋆]*, Edoardo Maria Leonardi*[⋆] *and Mauro Pontani*[⋆†]
[⋆]*Sapienza University of Rome*
*Via Salaria 851, 00138 Rome, Italy*
ciccotti.1897676@studenti.uniroma1.it · edoardomaria.leonardi@uniroma1.it · mauro.pontani@uniroma1.it
[†]Corresponding author

## Abstract

This research focuses on a servicing spacecraft equipped with a high-thrust propulsion system, capable of performing multiple successive transfers while minimizing the total velocity change for a given set of satellites to service. A two-layer optimization is proposed: (i) the inner layer estimates the costs of orbit transfers, utilizing analytical models and optimization algorithms; (ii) the outer layer determines the optimal sequence and timing of serviced satellites through a dedicated encoding and simulated annealing. Mission scenarios include the effect of the Earth oblateness, and intermediate drift orbits are leveraged to pursue natural alignment of orbit planes. The methodology is successfully applied to the Starlink constellation, with horizon ranging from 1.5 through 4 years.

## 1. Introduction

On-Orbit Servicing (OOS) is an emerging solution for maintaining and supporting satellites already in orbit, offering a cost-effective alternate option to their replacement.[1] In recent years, the challenge has moved from simply launching complex spacecraft and systems, to exploiting the systems already launched to provide cost-efficient support.[2] These activities range from refueling and orbit adjustments to modular upgrades and on-orbit refurbishment, which have matured to the point where they can be applied to fairly complex satellite systems (e.g. Hubble Space Telescope and the International Space Station).[3] Notably, satellite servicing contributes to sustainable space exploration by enabling the reuse of orbit assets and addressing critical challenges such as space debris mitigation.[4] Future advancements are set to expand these capabilities to more autonomous operations, reducing dependence on direct ground control and paving the way for servicing missions in distant orbits.

This research investigates the potential of OOS missions from an orbit mechanics perspective, focusing on a servicing satellite equipped with a high-thrust propulsion system, capable of performing multiple successive transfers. In the context of the impulsive thrust approximation, the overall velocity change associated with each orbit transfer represents an excellent estimation of the propulsive cost of each leg that composes the entire servicing mission. The problem consists in minimizing the total velocity change (in a specified time horizon) for a given set of satellites to service, and can be modeled as a variant to the classic Traveling Salesman Problem (TSP) with time-dependent and moving targets.[5] In particular, Izzo et al.[6] proposed an innovative approach to optimizing space debris removal missions, with regard to two new variants to the TSP: the static and dynamic city selection TSP. However, the proposed solution through the inver-over algorithm shows effectiveness primarily in the static case, where debris orbits are not subject to precession. For the dynamic case, where debris orbits evolve due to orbit perturbations, the performance of the inver-over algorithm declines, highlighting the need for further research to adapt evolutionary algorithms to dynamic conditions. Barea et al.[7] proposed a branch-and-bound-based framework for multi-target space debris removal missions, focusing on the optimization of object selection and trajectory planning. The approach uses direct two-impulse transfers between selected target objects, leveraging an extended Hohmann-like transfer with optimal split plane changes. The framework incorporates perturbations due to the Earth oblateness, and feasibility is verified through a Mixed Integer Nonlinear Programming (MINLP) model with a Benders decomposition method. This ensures that the proposed mission sequence satisfies constraints on mission duration and fuel budget, while maximizing a criticality index on the removed objects. Medioni et al.[8] explored trajectory optimization for multi-target Active Debris Removal (ADR) missions, employing a Simulated Annealing (SA) algorithm to minimize propellant consumption and mission duration. It dynamically balances the use of natural orbit plane drift and direct maneuvers. Zona et al.[9] focused on planning ADR missions, employing evolutionary algorithms to optimize the sequence and timing of

Copyright © 2025 by Claudio Ciccotti, Edoardo Maria Leonardi and Mauro Pontani. Published by the EUCASS association with permission.

DOI: 10.13009/EUCASS2025-338



debris rendezvous while minimizing the total velocity change. Notably, this study adopts a pragmatic approach using intermediate orbits, exploiting the natural precession of orbit planes due to the Earth oblateness. Cerf[10] presented a comprehensive method for planning missions to collect multiple space debris in Low Earth Orbit (LEO). The solution technique combines combinatorial optimization for selecting and sequencing debris, with continuous optimization to define orbit maneuvers. Cost matrices are constructed by solving nonlinear programming (NLP) problems for all possible transfers between debris, considering high-thrust and low-thrust propulsion, discretizing transfer dates, and incorporating intermediate drift orbits to leverage natural precession of the orbit planes for fuel-efficient orbit changes. Then, a SA algorithm optimizes the debris sequence and rendezvous dates by interpolating costs from the precomputed matrices.

The objective of this study is to develop an innovative approach for planning and executing servicing missions toward a set of target satellites. Because it involves both real-valued and integer variables, the problem at hand is rather challenging, and a two-layer optimization methodology is proposed for its solution: (i) the inner layer estimates the costs of orbit transfers (in terms of overall velocity change), utilizing analytical models and optimization algorithms, while (ii) the outer layer determines the optimal sequence and timing for servicing multiple satellites. To this aim, different optimization algorithms are being tested within the inner layer, in order to compare their performance in terms of computational efficiency and optimality of the solution. Furthermore, a dedicated encoding and SA is proposed for the outer layer, to allow the simultaneous selection of the visiting order and timing. Since non-polar orbits are subject to differential precession due to the Earth oblateness, the strategy at hand is intended to leverage both direct transfers and intermediate drift orbits, in order to pursue the natural alignment of orbit planes. Waiting time constraints are also being addressed, to allow phasing and servicing operations between orbit transfers in a real operational scenario, namely the Starlink constellation. In short, the present research aims at integrating natural orbit dynamics and advanced optimization algorithms into servicing strategies, paving the way for more sustainable and cost-effective OOS missions.

## 2. Orbit dynamics and maneuvers

The state of the satellite is described in terms of Classical Orbit Elements (COEs) $(a, e, i, \Omega, \omega, \theta^*)$, where $a$ is the semimajor axis, $e$ the eccentricity, $i$ the inclination, $\Omega$ the Right Ascension of the Ascending Node (RAAN), $\omega$ the argument of perigee, and $\theta^*$ the true anomaly. The Earth oblateness significantly affects the orbit motion of satellites, especially at low and medium altitudes, yielding an average time variation of the RAAN,

$$\langle \dot{\Omega} \rangle = -\frac{3R_E^2 J_2 \sqrt{\mu}}{2a^{7/2}(1-e^2)^2} \cos i, \tag{1}$$

where $R_E = 6378$ km denotes the radius of the Earth, $\mu$ its gravitational parameter, while $J_2 = 1.082 \cdot 10^{-3}$ is the Earth oblateness coefficient.

### 2.1 Analytical approach to optimal noncoplanar transfers

This section investigates the optimal strategy to change the orbit plane and its size. This approach is described by Vallado[11] and allows to determine the optimal distribution of the plane change associated with each maneuver. In order to rotate the orbit plane, a partial change in the inclination ($s\Delta\theta$) can be achieved with the first impulsive maneuver, while the remaining part $(1-s)\Delta\theta$ in the second one, where $\Delta\theta$ is the angle between the two orbit planes,

$$\cos(\Delta\theta) = \sin(i_i)\sin(i_f)\cos(\Omega_f - \Omega_i) + \cos(i_i)\cos(i_f), \tag{2}$$

where $(i_i, \Omega_i)$ and $(i_f, \Omega_f)$ respectively denote the inclination and RAAN of the initial and final orbits. The optimal value of the split index $s$ must be determined in order to minimize the total velocity change,

$$\Delta v = \Delta v_1 + \Delta v_2 = \sqrt{v_i^2 + v_1^2 - 2v_i v_1 \cos(s\Delta\theta)} + \sqrt{v_f^2 + v_2^2 - 2v_f v_2 \cos((1-s)\Delta\theta)}, \tag{3}$$

where $v_i$ and $v_f$ are the velocities along the initial and final orbits, while $v_1$ and $v_2$ respectively denote the velocities after the first impulsive maneuver and before the second impulsive maneuver. By taking the partial derivative of $\Delta v$ with respect to $s$, the following expression for $\sin(s\Delta\theta)$ is retrieved:

$$\sin(s\Delta\theta) = \frac{\Delta v_1 v_f v_2 \sin((1-s)\Delta\theta)}{\Delta v_2 v_i v_1}. \tag{4}$$

However, multiple local minima may exist, and they can also be located at the boundary of the search interval. Therefore, the cost function in Equation (3) can be evaluated for different values (e.g. 50) within the interval $(0, 1)$ for $s$: if the minimum is located at the boundary, the solution is found; otherwise, the lowest value can be used as first guess of the Matlab routine $@fzero$ to retrieve the zero of Equation (4).

DOI: 10.13009/EUCASS2025-338



### 2.2 Numerical optimization of non-coplanar two-impulse orbit transfers

If two generic orbits are considered, the optimal two-impulse maneuver cannot be found through simple analytical relations and a numerical procedure is necessary.

The problem consists in finding the velocity change vectors $\Delta\underline{\boldsymbol{v}}_1$ and $\Delta\underline{\boldsymbol{v}}_2$ so that the transfer is successfully performed and the cost function $J$ is minimized,

$$J = \|\Delta\underline{\boldsymbol{v}}_1\| + \|\Delta\underline{\boldsymbol{v}}_2\| = \|\underline{\boldsymbol{v}}_1 - \underline{\boldsymbol{v}}_i\| + \|\underline{\boldsymbol{v}}_f - \underline{\boldsymbol{v}}_2\|, \tag{5}$$

where $\underline{\boldsymbol{v}}_1$ and $\underline{\boldsymbol{v}}_2$ respectively denote the velocity vectors after the first impulsive maneuver and before the second impulsive maneuver. The transfer orbit can be characterized by the true anomalies along the departure and arrival orbits $\theta_{*,i}$ and $\theta_{*,f}$ and the time of flight $\Delta t$ between the two points. The routine *@coe2rv* allows to explicitly compute the position and velocity vectors $(\underline{\boldsymbol{r}}_i, \underline{\boldsymbol{v}}_i)$ and $(\underline{\boldsymbol{r}}_f, \underline{\boldsymbol{v}}_f)$ associated with the initial and final states. The complete analytical development is described by Pontani and Conway[12] . Furthermore, to estimate the total required velocity change, an algorithm *@lambert* for the solution of the Lambert problem can be leveraged[11] .

The optimization problem is solved through the *Particle-Swarm-Optimization* algorithm, which finds the best value of the unknown parameters $[\theta_{*,i}, \theta_{*,f}, \Delta t]$ that minimize the cost function, which is computed as follows:

$$\begin{bmatrix} \theta_{*,i} \\ \theta_{*,f} \\ \Delta t \end{bmatrix} \xrightarrow{@coe2rv} \begin{bmatrix} \underline{\boldsymbol{r}}_i \\ \underline{\boldsymbol{v}}_i \\ \underline{\boldsymbol{r}}_f \\ \underline{\boldsymbol{r}}_f \\ \Delta t \end{bmatrix} \xrightarrow{@lambert} \begin{bmatrix} \underline{\boldsymbol{v}}_1 \\ \underline{\boldsymbol{v}}_2 \end{bmatrix} \rightarrow J = \|\underline{\boldsymbol{v}}_1 - \underline{\boldsymbol{v}}_i\| + \|\underline{\boldsymbol{v}}_f - \underline{\boldsymbol{v}}_2\|. \tag{6}$$

### 2.3 Phasing strategy

This section outlines a phasing strategy that also includes the $J_2$ perturbation to account for the differential drift of the RAAN along the parking orbit with respect to the reference orbit. In a spacecraft rendezvous scenario, the orbit transfer is first achieved; then, the servicer spacecraft is placed on a parking orbit, with larger (external phasing) or smaller (internal phasing) semimajor axis, until phasing is completed. Injecting the spacecraft into the parking orbit requires two velocity changes along the horizontal direction, with the same magnitude $\Delta V_\theta$, together with an out-of-plane impulsive maneuver $\Delta V_h$ to correct the differential drift in the RAAN. The complete procedure to estimate the required $\Delta V_{ph}$ is developed by Santoro and Pontani[13] and Leonardi et al.[14] . In short, the design of the phasing maneuver includes the following steps:

1) find semimajor axis and eccentricity of the parking orbit as a function of the horizontal impulsive maneuver $\Delta V_\theta$;
2) evaluate the orbit period of the parking orbit $T$;
3) compute the required out-of-plane velocity change $\Delta V_h$ to correct the differential drift in the RAAN;
4) compute the variation of the argument of latitude $\Delta\theta$ caused by the out-of-plane maneuver;
5) enforce the phasing constraint in order to find $\Delta V_\theta$.

This procedure allows estimating the cost of either internal and external phasing according to the number of periods along the drift orbit. The results of this strategy for the cases of interest are reported in Sections 3 and 4.

## 3. Routing optimization for polar-orbit satellites

The On-Orbit Servicing problem involves finding the most efficient way of visiting several satellites to perform operations and optimize the overall trajectory in terms of cost indices. This problem is referred to as the Traveling Salesman Problem. However, this leads to an increase in the complexity if applied to space transfers. Indeed, in this application targets are moving and therefore the cost of the transfer path from satellite $i$ to satellite $j$ is time-dependent. This means that the overall cost does not depend only on the target sequence, but also on the time instant when transfers are performed and on the single-leg strategy that determines the associated velocity change.

It is worth mentioning that the problem is *double* time-dependent, because of the $J_2$ effect on the orbit, which causes the orbit elements to change in time. However, this section assumes near-circular polar orbits, which are not affected by any change in the RAAN; indeed, the average variation of the RAAN due to the $J_2$ harmonic vanishes is $i = 90$ deg (cf. Equation (1)). Instead, the dynamical version of the problem for inclined orbits will be tackled in Section 4. In general, the problem can be divided into two layers:

DOI: 10.13009/EUCASS2025-338



- the **inner layer**, which optimizes the single-leg transfer;
- the **outer layer**, which finds the optimal sequence.

### 3.1 Inner layer

The inner layer of the problem consists in finding the optimal transfer from orbit $i$ to orbit $j$ assuming that favorable conditions occur to perform the orbit-to-orbit optimal maneuver. Then, phasing is performed to achieve rendezvous with the target satellite. Therefore, for each pair of orbits $(i, j)$, the optimal velocity change $\Delta v$ can be computed and this allows filling the component along row $i$ and column $j$ of the matrix $\Delta V_{mat^2}$ with the optimal transfer cost,

$$\Delta V_{mat^2}(i, j) = \Delta v_{ij}, \quad i, j = 1, ..., n \tag{7}$$

where $n$ is the number of satellites. Obviously, $\Delta V_{mat^2}(i, i) = 0$, meaning that there is no need to perform a maneuver to remain along a given orbit. The matrix $\Delta V_{mat^2}$ serves as an input for the outer layer, which selects the sequence.

### 3.2 Outer layer

The outer layer takes the generated cost matrix $\Delta V_{mat^2}$ as an input, and aims at finding the best visit order. This problem can be formulated through two main approaches, as described by Federici et al.[15] . In this study, the *PO Formulation* (Permutation Optimization) is adopted, where the permutation $\mathcal{P}$ minimizes the cost function

$$\min_{\mathcal{P}} \left\{ \sum_{i=1}^{n-1} \Delta V_{mat^2} \left( \mathcal{P}_i, \mathcal{P}_{i+1} \right) \right\}. \tag{8}$$

In this analysis, both permutation-based methods and real-valued methods are used to find the best permutation and they are collected in Table 1:

| **Permutation-based methods** | **Real-valued methods** |
|---|---|
| Brute-Force (BF) | Genetic Algorithm (GA) |
| Inver-Over[6] (IO) | Particle Swarm Optimization (PSO) |
| Ant-Colony Optimization[16] (ACO) | Differential Evolution[17] (DE) |
| Tabu Search[18] (TS) | Stochastic Fractal Search[19] (SFS) |

Table 1: List of algorithms.

However, real-valued heuristic methods require a dedicated permutation encoding, which is described in Algorithm 1.

**Algorithm 1** Map a real number $x \in [0, 1]$ to a permutation of $1 : n$

1: **Input**: A real number $x \in [0, 1]$, and an integer $n$.
2: Calculate the total number of permutations: numPerms = $n!$.
3: Divide the interval [0, 1] into equally-spaced numPerms sub-intervals.
4: Assign each sub-interval to one of the permutations of $\{1, 2, \ldots, n\}$ in lexicographic order.
5: Find the index $i$ of the sub-interval containing $x$: $i = \lfloor x \times \text{numPerms} \rfloor + 1$.
6: Retrieve the $i$-th permutation from the lexicographically ordered set of permutations.
7: **Output** the $i$-th permutation.

### 3.3 Numerical Results

In order to show the method of solution, two examples are reported, with increasing difficulty of the combinatorial problem. The orbits of the target satellites are randomly chosen in the following intervals:

- altitude of periapsis: $h_p > 400$ km;
- altitude of apoapsis: $h_a < 800$ km;
- inclination: $i = 90\,\text{deg}$;

DOI: 10.13009/EUCASS2025-338



- argument of perigee: $\omega \in (-180, 180)$ deg;
- RAAN: $\Omega \in (0, 20)$ deg;
- true anomaly: $\theta_* \in (-180, 180)$ deg.

#### 3.3.1 7-satellite mission

In the first scenario, only 7 satellites are considered. Figures 1 (a) and (b) show the orbit elements of the satellites, while (c) depicts the associated cost matrix. The primary factor contributing to the overall velocity change is associated with the RAAN of the satellites. Specifically, satellites 1 and 3, which have the highest associated cost, exhibit the greatest difference in RAAN, while satellites 5 and 6, with the lowest cost, share similar values of the RAAN.

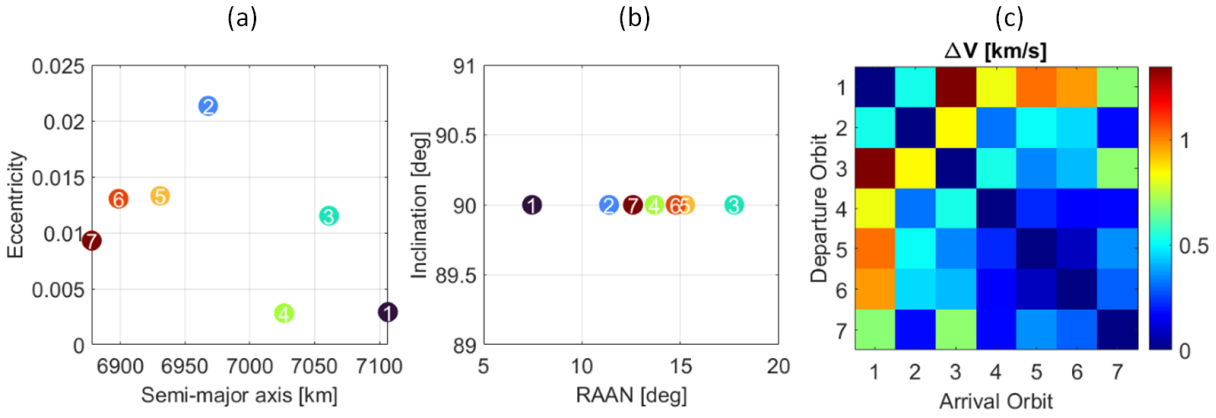



Figure 1: orbit elements (a), (b) and cost matrix of transfers (c).

**Brute-Force** In this method, each sequence is first evaluated and then the minimum-cost path is identified. The brute force method is the only approach that guarantees the optimality of the solution is guaranteed, which turns out to be

$$\mathcal{P}_{BF} = [3, 5, 6, 4, 7, 2, 1], \tag{9}$$

with an overall cost of 1.4667 km/s. Interestingly, the optimal solution requires visiting satellites with ascending or descending values of the RAAN (*RAAN-walk*[6]). For all iterations the computation time is 0.00548 s on a CPU Intel(R) Core(TM) i9-10980 @ 3.00 GHz.

**Inver-Over**[6] This method iteratively selects a segment of a tour and inverts its order. This inversion operation may be guided either through the tour of another randomly selected individual within the population or by a random mutation, in order to balance the effectiveness of local search and global exploration. This algorithm is featured with very high efficiency in the permutation space, caused by the unique scheme of segment inversions. The optimization process takes approximately 0.11525 s and the optimized permutation is

$$\mathcal{P}_{IO} = [3, 5, 6, 4, 7, 2, 1], \tag{10}$$

as in the brute force method.

**Ant-Colony Optimization**[16] This method allows exploring the permutation workspace by means of fictitious ants walking in the graph and their release of pheromone. Initially, the pheromone is spread uniformly across all paths; then the pheromone is updated based on the overall fitness function along the path followed by the ant. The path associated with a lower fitness function has a greater deposit of pheromone. At the next iteration, each ant in the colony is initialized in a random node, while the next node is chosen according to a probability function that depends on the pheromone in the arc $(i, j)$ and on its desirability (i.e. a lower $\Delta v_{ij}$ is associated with a higher desirability). The optimization process is run for 30 iterations with $10 \cdot n$ ants and yields the same solution as the brute force method,

$$\mathcal{P}_{ACO} = [3, 5, 6, 4, 7, 2, 1]. \tag{11}$$

The computation time is 0.16781 s.

DOI: 10.13009/EUCASS2025-338



**Tabu Search**[18] This method is based on a single individual which is improved each time. At the beginning of the algorithm, a list of possible *Actions* is defined, which is related to the type of action (*Insertion*, *Reversion* and *Swap*) and associated indices of the permutation vector. For each iteration, all the *Actions* are applied to the same individual and the one with the best fitness function is brought to the next iteration. The related *Action* is inserted in the *tabu list*, so that the same action cannot be performed in the next iterations. As in the previous case, the optimization process is run for 30 iterations and yields the same result,

$$\mathcal{P}_{TS} = [3, 5, 6, 4, 7, 2, 1]. \tag{12}$$

The computation time is 0.02160 s.

Details on real-valued methods are not reported as they tend to underperform, especially with an increasing number of satellites. The performance of the different algorithms is reported in Table 2, which shows that all the methods are capable of finding the minimum-cost solution, which is proven to be global since brute force is available. Moreover, it is evident that the optimal path and its reverse share the same fitness function.

| Algorithm | Sequence | Cost (km/s) | CPU time (s) |
|---|---|---|---|
| BF | [3 5 6 4 7 2 1] | 1.4667 | 0.00548 |
| IO | [3 5 6 4 7 2 1] | 1.4667 | 0.11525 |
| ACO | [3 5 6 4 7 2 1] | 1.4667 | 0.16781 |
| TS | [1 2 7 4 6 5 3] | 1.4667 | 0.02160 |
| GA | [1 2 7 4 6 5 3] | 1.4667 | 1.85839 |
| PSO | [1 2 7 4 6 5 3] | 1.4667 | 0.74945 |
| DE | [3 5 6 4 7 2 1] | 1.4667 | 4.45960 |
| SFS | [1 2 7 4 6 5 3] | 1.4667 | 3.90052 |

Table 2: Outcomes and computation times of the algorithms.

### 3.3.2 50-satellite mission

To assess the efficiency of permutation-based algorithms, a more complex scenario is addressed, which involves 50 satellites. As a preliminary step, an intermediate test case with 15 satellites is considered to select the numerical methods capable of yielding the optimal solution. Specifically, the real-valued methods GA, DE, PSO, and SFS are excluded, while BF is infeasible due to the increased computational time. Then, the IO, ACO and TS algorithms are employed for the 50-satellite scenario and the relevant parameters of each algorithm are reported:

- Inver-Over
  - population size = $10 \cdot n$;
  - maximum number of iterations = 50000;
  - chance of a random inversion[6]= 0.05 .
- Ant-Colony Optimization
  - number of ants = $10 \cdot n$;
  - maximum number of iterations = 50;
  - other parameters set to default values.[16]
- Tabu-Search
  - maximum number of iterations = 200;
  - other parameters set to default values.[18]

The results of the optimization are reported in Table 3.

| Algorithm | Sequence | Cost (km/s) | CPU Time (s) |
|---|---|---|---|
| IO | [10 22 7 29 47 ... 48 44 28 17 32] | 4.7798 | 4.43 |
| ACO | [22 10 7 29 47 ... 48 32 17 28 44] | 5.1268 | 12.31 |
| TS | [10 22 7 29 47 ... 48 44 28 17 32] | 4.7798 | 5.61 |

Table 3: Outcomes and computation times of the algorithms.

DOI: 10.13009/EUCASS2025-338



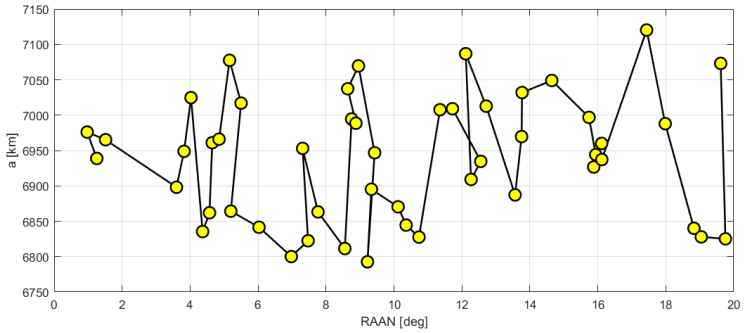



Figure 2: Best permutation found with TS

.

Apparently, both IO and TS outperformed ACO in terms of optimality of the solution and CPU time. Remarkably, Figure 2 shows that the optimal solution is not associated with a RAAN walk, which instead exhibits a higher $\Delta v$ (5.204 km/s).

### 3.4 Phasing

The theoretical procedure summarized in Section 2.3 can be applied to a real orbit to assess the order of magnitude of a phasing maneuver and its dependence on the initial phasing angle $\Delta\alpha$ between the target and the chaser. A generic circular orbit of altitude $h = 700$ km and inclination $i = 90$ deg is considered, and the results are briefly described. By increasing the number of orbits spent in the parking orbit, the cost of phasing is reduced with the drawback of increasing the maneuver time. The time needed to allow phasing can be computed considering an average orbit period of $T = 1.6$ h, yielding a total duration of the maneuver of 12 days and an associated $\Delta V_{ph}$ equal to 14 m/s.

## 4. Routing optimization for non-polar-orbit satellites

Removing the assumption of polar orbits introduces complexity, as the variation of the RAAN makes the matrix $\Delta V_{mat}$ time-dependent (cf. Equation (7)). This scenario can be classified as a Time-Dependent Traveling Salesman Problem (TDTSP), which is a variation of the classic TSP. For this reason, the computation of the optimal solution requires establishing not only the visiting order, but also the exact transition times.

The problem is still divided into two layers: the inner layer is in charge of the single-leg cost estimation, while the outer layer identifies the satellite visit order and, in this case, also the encounter times. However, in this case the matrix $\Delta V_{mat}$ is time-varying, because the cost of the optimal transfer from satellite $i$ to $j$ is not constant due to the drift of the orbit planes. This problem is addressed by creating a discrete time interval $tspan$ and evaluating the matrix at each component $k = 1, ..., n_t$. Consequently, a three-index notation will be used to identify the optimal transfer from orbit $i$ to orbit $j$ at time instant $\tau_k$,

$$\Delta V_{mat^3}(i, j, k) = \Delta v_{ijk}. \tag{13}$$

Then, the outer layer algorithm must choose the permutation and collocate each node at a specific time interval,

$$\Delta V_{mat^3}(:, :, k) = \Delta V_{mat^2}(i, j) \quad \text{at time } \tau_k. \tag{14}$$

### 4.1 Drift orbits

The use of only direct transfers is infeasible, thus intermediate orbits (*drift orbits*) can be exploited to reduce the transfer cost. Each intermediate orbit is characterized by a faster or slower drift rate with respect to the initial orbit, so that the RAAN difference between orbits $i$ and $j$ is reduced. Since the duration of the transfer legs between the initial and final orbits and the drift orbit is negligible, the time of flight from orbit $i$ to orbit $j$ is identified by the time spent along the drift orbit, denoted with $\Delta t_d$. Therefore, an additional discrete time interval *durspan* is introduced, related to the duration of the leg along the drift orbit. Then, the four-index matrix $\Delta V_{mat^4}(i, j, k, d)$ is evaluated at each component $d = 1, ..., n_d$.

Orbit transfers to and from drift orbits can be modeled as impulsive Hohmann-like transfers. Each transfer (from satellite $i$ to the drift orbit, and then from the drift orbit to satellite $j$) consists of two velocity impulses, with the inclination change split between them. The approximation is described by Cerf[10] and reported in the following. The

DOI: 10.13009/EUCASS2025-338



aim of the optimization is to find the orbit speed $V_d$ and inclination $I_d$ of the circular drift orbit so that satellite $i$ is able to reach the orbit of satellite $j$ at time $\tau_k + \Delta t_d$ with minimum cost.

If $\Omega_{i0}$ denotes the initial RAAN at the beginning of the first maneuver, and $\Omega_{jf} = \Omega_{j0} + \dot{\Omega}_j \cdot \Delta t_d$ the final RAAN, the orbit elements of the drift orbit must verify the following constraint:

$$\dot{\Omega}_{ij} = -\frac{3}{2} J_2 \sqrt{\mu} R_E^2 (R_E + H_d)^{-7/2} \cos(I_d) = \frac{\Omega_{jf} - \Omega_{i0}}{\Delta t_d} \pm 2k\pi, \quad k \in \mathbb{Z}, \tag{15}$$

where $H_d = \mu / V_d^2 - R_E$ is the altitude of the drift orbit. It is worth mentioning that the difference $(\Omega_{jf} - \Omega_{i0})$ in Equation (15) must be adjusted with $\pm 2k\pi$ in order to find the precession value of the drift orbit that is closer to $\dot{\Omega}_{ij}$. Therefore, even if the problem involves only a single unknown, both $V_d$ and $I_d$ are employed as unknown variables to enhance convergence, while Equation (15) is enforced as a constraint. The MATLAB built-in routine *fmincon* is applied in order to minimize the transfer cost while satisfying the constraint. Furthermore, an additional constraint is included to enforce a minimum altitude (i.e. 400 km) for drift orbits, which corresponds to an upper bound on the velocity along the circular orbit ($V_{max}$). Following the optimization, the optimized parameters are given as outputs together with the overall transfer cost, and the type of drift orbit (internal or external). To achieve convergence, a good initial guess must be provided. If $d = 1$, the initial guess for the drift orbit velocity is set to the average between the chaser and target orbit velocities. If the first initial guess velocity exceeds the maximum allowed value, then $V_d$ is set to $V_{max}$ and $I_d$ is re-computed through Equation (15). Then, the optimal transfer associated with indices $(i, j, k, d)$, with $d > 1$, serves as the initial guess for the optimization procedure related to indices $[i, j, k, d + 1]$.

### 4.2 Inner layer

The inner layer allows filling up the four-index matrix $\Delta V_{mat^4}(i, j, k, d)$ that serves as an input for the outer layer. To this aim, the optimal transfer from orbit $i$ to orbit $j$ at time $\tau_k$ and time of flight $\Delta t_d$ must be determined,

$$\Delta V_{mat^4}(i, j, k, d) = \Delta v_{ijkd} \quad i, j = 1, ..., n; \quad k = 1, ..., n_t; \quad d = 1, ..., n_d. \tag{16}$$

Therefore, the solver must complete $n \times (n-1) \times n_t \times n_d$ optimization procedures in order to construct the four-index matrix, thus an efficient cost estimation of the leg is crucial for both direct transfers and transfers through drift orbits. Furthermore, it is worth remarking that since the first value of the duration vector (associated with index $d = 1$) is $\Delta t_1 = 0$, each element $\Delta V_{mat^4}(i, j, k, 1)$ identifies a direct transfer. These direct transfers are generally related to higher costs, but it is useful to include them in the analysis to exploit the natural alignment of the orbit planes, which represents the optimal condition to perform a direct transfer with no intermediate orbit.

### 4.3 Outer layer

This section details the outer layer for the TDTSP. Once matrix $\Delta V_{mat^4}(i, j, k, d) = \Delta v_{ijkd}$ is correctly filled, the outer layer provides the optimal visit order and the encounter times. This problem is tackled with a permutation encoding that allows to optimize visit order and encounter times simultaneously.

#### 4.3.1 Permutation encoding

The outer layer aims at finding the optimal permutation, and for each arc the best starting transfer epoch and transfer duration. Indeed, the same permutation can be related to a huge number of missions, and the transfer epochs must be determined.

The solving procedure is based on the encoding proposed by Zona et al.[9] , which discretizes the overall time span, in order to cast the mixed-integer optimization as a pure combinatorial optimization problem. To do this, a permutation-based encoding is introduced. In particular, if $n$ and $n_t$ still denote the number of targets and discretized time epochs, the following permutation must be optimized:

$$\Pi = [1, 2, 3, ..., n, ..., n_t - 1, n_t], \quad n_t > n. \tag{17}$$

In order to obtain both the permutation and the time intervals, elements greater than $n$ are considered as blank slots, so that the optimized permutation of the $n$ satellites will appear along with the desired time intervals, as outlined in Figure 3, where $\mathcal{P}$ denotes the permutation of the subset of $n$ satellites.

DOI: 10.13009/EUCASS2025-338



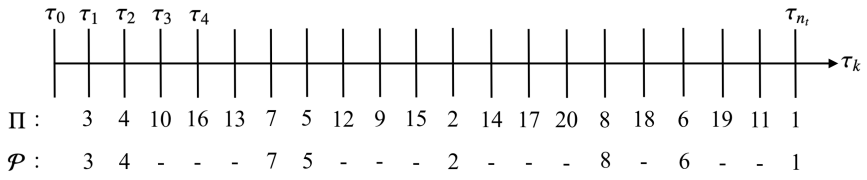



Figure 3: Permutation encoding adopted in the method (number of satellites = 8).

#### 4.3.2 Waiting time constraints

Algorithm 2 outlines the computation of the cost function capable of finding the transfer specifications and overall mission cost from the permutation Π. First, a waiting time constraint must be enforced along each leg. In particular, 12 days are allowed to perform phasing & docking, while 2 days are assumed for servicing operations, so that a total amount 14 days is allocated. In order to enforce this constraint, a variable $m$ is introduced, which is the ratio between the maximum waiting time $T_{wait}^{max}$ (i.e. 14 days) and the discretization interval $\Delta\tau$, rounded up to the smallest integer,

$$m = \inf\left\{ z \in \mathbb{Z} \ : \ z \geq \frac{T_{wait}^{max}}{\Delta\tau} \right\}. \tag{18}$$

Once the meaningful numbers in the permutation (i.e. $\leq n$) are identified, the visit order $\mathcal{P}$ and the departure times are retrieved, storing the related indices in the *indices* vector. Then, for each transfer, the integer quantity $\tilde{d}$ is computed, i.e. the largest duration index allowed for that transfer including the wait time constraint, and the one related to the minimum cost is stored.

**Algorithm 2** Pseudocode for the computation of the cost function.

**Input:** Π, $\Delta V_{mat^4}$
**Output:** $\Delta v$ (total transfer cost)

*indices* ← indices where Π ≤ $n$
$\mathcal{P}$ ← Π(*indices*)
$\Delta v = 0$
**for** $k = 1 : length(\mathcal{P}) - 1$ **do**
    $\tilde{d} \leftarrow indices(k+1) - indices(k) - m + 1$
    $\Delta v \leftarrow \Delta v + \min\{\ \Delta V_{mat^4}(\mathcal{P}_k, \mathcal{P}_{k+1}, indices(k), 1 : \tilde{d})\ \}$
**end for**

**return** $\Delta v$

As an illustrative example, let us consider the permutation Π in Figure 3. If the discretization interval is 2 weeks for both *tspan* and *durspan*, then the waiting time parameter is $m = 1$. The first is executed at time $\tau_1$ (*tspan*(2)) starting from satellite 3, so that $\tilde{d} = (3 - 2 - 1 + 1) = 1$. Since the first index in the duration vector is related to $\Delta t_{d=1} = 0$, a direct transfer to satellite 4 is performed. The second leg, instead, connects satellite 4 to satellite 7. In this case, the maximum duration index is $\tilde{d} = (7 - 3 - 1 + 1) = 4$, so that a duration of $\Delta t_{d=4} = (4 - 1) \cdot 2$ weeks = 6 weeks is allowed.

#### 4.3.3 Method of solution

A modified version of the SA algorithm[20] is used to solve the problem at hand. SA is a probabilistic optimization technique, inspired by the annealing process in metallurgy, where metals are first heated and then gradually cooled down until they achieve some stable crystalline structure. The aim of SA is to locate the global minimum or maximum of a cost function by exploring the solution space so that exploration and exploitation are balanced.

Starting from a random permutation Π, at each iteration a set of *Neighbor* permutations of the current solution is generated according to the following different choices and associated probabilities: insert (20%), reverse (40%), swap (20%), scramble (10%) or change date (10%). If the new solution is better, it is accepted, otherwise it is ruled out or accepted based on a probability function that decreases with the temperature. However, the main drawback of the formulation described in the previous section is that the cost function is heavily non-injective. In fact, since all the numbers $\tilde{n}$ such that $n < \tilde{n} < n_t$ are seen as blank spots, every swap in those numbers will result in the same solution.

DOI: 10.13009/EUCASS2025-338



Consequently, if the selected indices are related to an unmeaningful action, these are re-chosen randomly. Note that the last action (i.e. change date) represents a novelty with respect to the implementation by Heris[20] and involves a single-index action. This takes an element of the permutation and swaps randomly with its previous or next one, in order to allow a local search of the encounter epochs.

### 4.4 Test case from literature

To test the proposed methodology, a well-known test case presented by Cerf[10] is considered. This test case involves a series of orbit transfers targeting a specific set of debris objects, which is reported in Table 4.

| Debris number | Altitude (km) | Inclination (deg) | Initial RAAN (deg) |
|---|---|---|---|
| 5 | 740 | 98.2 | 18 |
| 16 | 850 | 97.5 | 324 |
| 17 | 860 | 97.8 | 72 |
| 20 | 890 | 98.7 | 342 |
| 21 | 900 | 99.0 | 360 |

Table 4: Set of debris objects[10] .

In this case, the discretization interval is set to $\Delta\tau = 7$ days, and the waiting time constraint is set to $T_{wait}^{max} = 7$ days, yielding $m = 1$. Consequently, direct transfers are evaluated with starting times increased by 7 days. When drift orbits are considered, it can be reasonably assumed that the phasing condition is met at zero additional cost,[10] and this assumption is more consistent as the transfer time increases. Instead, for direct transfers, phasing must be accounted for. In this case, the reference orbit has altitude $h = 850$ km and inclination $i = 98$ deg, yielding an average orbit period of $T = 1.7$ hrs. Since the waiting time is 7 days, 90 orbits can be completed, related to a worst case $\Delta V_{ph} \approx 40$ m/s. The results of the optimization process, including the effect of phasing, are compared to the test case from literature.

| **Literature** | | | **Solution without phasing** | | | **Solution with phasing** | | |
|---|---|---|---|---|---|---|---|---|
| Debris number | Date (days) | $\Delta V$ (m/s) | Debris number | Date (days) | $\Delta V$ (m/s) | Debris number | Date (days) | $\Delta V$ (m/s) |
| 16 | 3.1 | 287.1 | 16 | 0.0 | 276.7 | 16 | 0.0 | 294.0 |
| 20 | 183.1 | 210.8 | 20 | 189.0 | 206.4 | 20 | 168.0 | 202.0 |
| 21 | 389.3 | 202.2 | 21 | 392.0 | 181.8 | 21 | 371.0 | 189.0 |
| 5 | 513.6 | 111.0 | 5 | 539.0 | 102.6 | 5 | 511.0 | 108.2 |
| 17 | 545.3 | - | 17 | 546.0 | - | 17 | 546.0 | - |
| Total | | 811.1 | Total | | 767.5 | Total | | 793.2 |

Table 5: Comparison between the test case from literature and the proposed solution.

Table 5 collects the debris ID, the starting epochs for the transfers, and the cost of each leg. By comparing the solutions without phasing, it is evident that the debris order, the departure times and associated $\Delta V$s are accurately estimated with respect to the literature solution. However, slight differences emerge in the last two legs, depicted in blue.

**$\Delta\Omega$-plot.** The $\Delta\Omega$-plot is introduced in Figure 4 to allow deeper understanding of the behavior of the transfer. The lines correspond to the evolution of all possible pairwise differences in RAAN (in absolute values). In particular, since the RAAN difference strongly affects the $\Delta v$, direct transfers can be conveniently performed when two orbit planes are close, ideally coincident. The dots in Figure 4 mark the time instants when a transfer begins. The color and the label of each dot specify the transfer type and the satellite number. In this case, internal drift orbits are chosen (as in the reference solution). Indeed, if $\Delta\Omega = \Omega_j - \Omega_i > 0$, the chaser spacecraft must travel a lower-altitude retrograde orbit to increase the precession rate. Instead, if $\Delta\Omega < 0$, an external drift orbit is necessary to reduce the precession rate.

Figure 4 allows a better understanding of the difference between the test case and the proposed solution with respect to legs $21 - 5$ and $5 - 17$. Indeed, since close to the final time the $\Delta\Omega$ between orbits 5 and 17 is approaching zero, the solver chooses to perform a direct transfer, and this results in a slightly lower cost with respect to the reference solution, where only drift orbit transfers are considered. This allows delaying the starting epoch of the last leg, so that the latter leg can benefit from a longer drift time, consequently lowering the overall $\Delta V$.

DOI: 10.13009/EUCASS2025-338



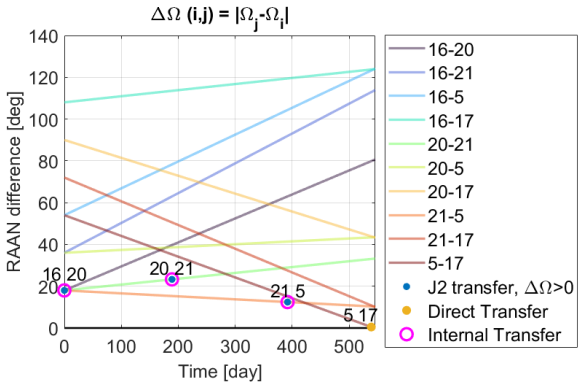



Figure 4: ΔΩ-plot without phasing.

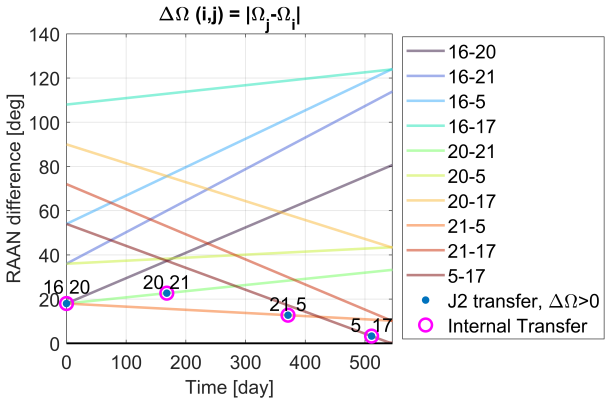



Figure 5: ΔΩ-plot with phasing.

From comparison between Figures 4 and 5, it is evident that direct transfers are avoided when phasing is included, resulting in an overall increase of about 26 m/s in the mission cost. Remarkably, the new solution differs from the one found in the literature, still offering a slightly lower cost. Tables 6 and 7 collect the overall mission details; in particular, the field *type* specifies if the solution is achieved through direct transfer (type 0), internal drift orbit (type= $-1$) or external drift orbit (type= $+1$).

| # dep | # arr | $t_{dep}$ [day] | $\Delta t$ [day] | $h$ [km] | $i$ [deg] | type | $\Delta v$ [m/s] |
|---|---|---|---|---|---|---|---|
| 16 | 20 | 0 | 182 | 850 → 728 → 890 | 97.5 → 98.9 → 98.7 | -1 | 276.7 |
| 20 | 21 | 189 | 196 | 890 → 749 → 900 | 98.7 → 99.4 → 99.0 | -1 | 206.4 |
| 21 | 5 | 392 | 140 | 900 → 738 → 740 | 99.0 → 98.9 → 98.2 | -1 | 181.8 |
| 5 | 17 | 539 | 0 | 740 → 860 | 98.2 → 97.8 | 0 | 102.6 |
| | | | | | | | 767.5 |

Table 6: Results of the optimization without phasing.

| # dep | # arr | $t_{dep}$ [day] | $\Delta t$ [day] | $h$ [km] | $i$ [deg] | type | $\Delta v$ [m/s] |
|---|---|---|---|---|---|---|---|
| 16 | 20 | 0 | 161 | 850 → 712 → 890 | 97.5 → 98.9 → 98.9 | -1 | 294.0 |
| 20 | 21 | 168 | 196 | 890 → 752 → 900 | 99.4 → 99.4 → 99.0 | -1 | 202.0 |
| 21 | 5 | 371 | 133 | 900 → 730 → 740 | 99.0 → 99.0 → 98.2 | -1 | 189.0 |
| 5 | 17 | 511 | 28 | 740 → 723 → 860 | 98.2 → 98.3 → 97.8 | -1 | 108.2 |
| | | | | | | | 793.2 |

Table 7: Results of the optimization with phasing.

### 4.5 Starlink constellation

Starlink is a satellite mega-constellation developed by SpaceX with the primary goal of providing global high-speed, low-latency internet coverage. It represents one of the most ambitious and innovative endeavors in the field of satellite communications, aiming to bridge the digital divide by delivering internet access to remote and underserved regions, where traditional infrastructure is impractical or unavailable. The orbit elements of the constellation can be obtained from the CeleStrak website[21] . Since the orbits have an average eccentricity of $1.6885 \cdot 10^{-4}$, with a maximum value of $3.1889 \cdot 10^{-3}$, they are reasonably assumed as circular. Orbits with altitude lower than 450 km are discarded since internal drift orbits cannot be exploited without accounting for atmospheric drag. Furthermore, since the inclinations are divided in groups, the set associated with $i \in (50, 60)$ deg is selected. Among them, a random subset of 20 target satellites is chosen, and the related orbit elements are collected in Table 8. The selected satellites span a range of altitudes and orientations, ensuring a wide representation of operational scenarios. For better understanding and visualization, the spatial arrangement of these satellites in their respective orbits is depicted in Figure 6.

DOI: 10.13009/EUCASS2025-338



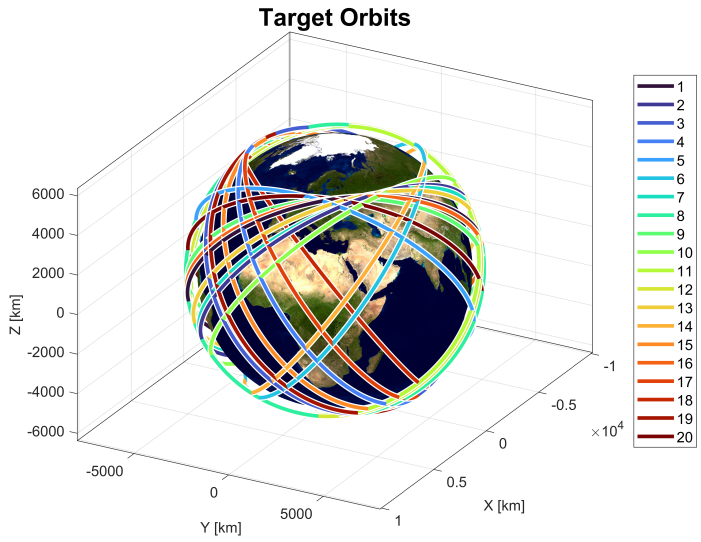



Figure 6: Target set from Starlink constellation.

| # satellite | $h$ [km] | $i$ [deg] | $\Omega_0$ [deg] | $\dot{\Omega}$ [$10^{-6}$ rad/s] |
|---|---|---|---|---|
| 1 | 539.71 | 53.218 | -48.92 | -0.9073 |
| 2 | 539.76 | 53.217 | -27.91 | -0.9073 |
| 3 | 539.74 | 53.216 | 157.93 | -0.9073 |
| 4 | 547.31 | 53.055 | -168.21 | -0.9072 |
| 5 | 539.74 | 53.217 | -88.50 | -0.9073 |
| 6 | 539.77 | 53.218 | 32.04 | -0.9072 |
| 7 | 492.31 | 53.050 | -30.78 | -0.9330 |
| 8 | 547.23 | 53.053 | 113.45 | -0.9073 |
| 9 | 463.25 | 53.161 | -57.19 | -0.9445 |
| 10 | 547.14 | 53.055 | -16.75 | -0.9073 |
| 11 | 475.23 | 53.161 | 101.85 | -0.9338 |
| 12 | 547.23 | 53.054 | 103.52 | -0.9073 |
| 13 | 539.74 | 53.219 | -37.25 | -0.9072 |
| 14 | 539.79 | 53.219 | 23.42 | -0.9072 |
| 15 | 475.21 | 53.160 | 174.56 | -0.9338 |
| 16 | 539.74 | 53.216 | -51.77 | -0.9073 |
| 17 | 539.76 | 53.218 | -155.78 | -0.9073 |
| 18 | 463.17 | 53.161 | -138.26 | -0.9446 |
| 19 | 539.74 | 53.215 | 162.81 | -0.9073 |
| 20 | 529.88 | 53.053 | -73.43 | -0.9153 |

Table 8: orbit elements of the target set @ 31-Oct-2024 06:01:21 UTC.

### 4.5.1 Simulated Annealing parameters

The SA algorithm must be properly tuned with its hyperparameters[20] in order to obtain satisfactory results. Due to the high complexity of the combination problem, the optimization process is repeated 50 times for each case, to enhance the exploration of the entire solution space. After trial and error, the initial temperature is set to $T_0 = 0.150$ (which is related to an initial acceptance probability of 90%), the maximum number of iterations to $MaxIt = 6000$, the maximum number of sub-iterations (i.e. at the same temperature) to $MaxSubIt = 500$, and the temperature rate to $\alpha = 0.999$. These hyperparameters are shown to yield satisfactory results across all scenarios.

### 4.5.2 Implementation details

The following practical implementation remarks are reported:

- the cost index of the mission is the total $\Delta v$, which is directly related to the mass ratio. In order to have an estimation of the maximum allowed $\Delta v$, the Tsiolkovsky equation can be exploited, with effective exhaust velocity of the high-thrust propulsion system set to 3 km/s and minimum mass ratio equal to 0.2, yielding $\Delta v = 4.828$ km/s;
- different mission scenarios are considered, with total duration ranging from 1.5 to 4 years, with the inclusion of the phasing maneuver;
- due to the discretization of the time span, a perfect match between two orbit planes $i$ and $j$ can never happen. This could lead to an overestimation of the cost associated with direct transfers. Therefore, when computing direct transfers, if $|\Omega_j - \Omega_i| < 1$ deg, the RAAN difference is set to zero, since there always exists a time at which the two planes coincide. The resulting maneuver requires a simple inclination and semi-major axis change;
- the time vector $tspan$ and the drift duration vector $durspan$ are discretized assuming 1-week intervals. A maximum duration of 5 months is allowed for the drift phase.

Several scenarios with different time windows are investigated and three limiting cases are reported: the first one considers a time interval of 1.5 years, while the other two scenarios assume a time window of 4 years, either without and with phasing penalization. The computation of matrix $\Delta V_{mat^4}$ in the inner layer requires 4 hours.

### 4.5.3 1.5-year mission

The results of the first test case are reported in Figure 7 and Table 9. The $\Delta\Omega$-plot looks messy at a first sight with respect to the 5 satellite-case from literature in Figure 4, since the number of lines in the plot increases up to $20 \cdot (20-1)/2 = 190$, but it allows a full understanding of the problem at hand. Remarkably, several orbits share the same precession rate, resulting in an almost horizontal line. Instead, orbits with different precession rates allow the RAAN to match naturally.

DOI: 10.13009/EUCASS2025-338



The overall cost is 7.979 km/s, meaning that this mission is infeasible. Both direct transfers and drift orbit transfers are selected; however, it is evident from Figure 7 that direct transfers are not only performed when there is a plane crossing, but also when the RAAN difference is close to zero, due to the very short time span. A closer inspection of Table 9 reveals that intermediate orbits share high altitudes in order to comply with the limited available time, yielding an increase in the overall $\Delta v$. The minimum-fuel solution is found in 13/50 iterations and the overall procedure takes 3.8 hours. It is worth remarking that another local minimum is found in 2/50 iterations, corresponding to a solution that differs from the previous one by only $\approx 1.63$ m/s.

#### 4.5.4 4-year mission

The results of the second test case are reported in Figure 8 and Table 10. The mission requires an overall cost of 1.954 km/s. The optimization takes 4.1 hours, and the minimum-fuel solution is found 5/50 times. In Table 10, the duration associated with leg $8-3$ is colored in orange because it reaches the maximum duration allowed by the algorithm.

Phasing is also addressed within this scenario. Across a sufficiently long time interval, set to 12 days, 180 orbits can be traveled, yielding a total velocity change of $\Delta v_{ph} = 50$ m/s. Therefore, this quantity is added to any direct transfer as the worst case, and the results of the optimization are reported in Table 11. Even though the visiting order $\mathcal{P}$ remains, the whole permutation $\Pi$ is different. The results in Table 11 highlight that direct transfers are no longer chosen, and drift orbits are used instead. In these cases, the characteristic orbit elements of the drift legs are reasonably very similar to the orbit elements of either the target or the chaser.

## 5. Concluding Remarks

This work presents a thorough study on On-Orbit Servicing (OOS), focusing on dynamical modeling and routing optimization. The mission scenario involves a spacecraft equipped with a high-thrust propulsion system, capable of servicing a given set of satellites, while minimizing the total velocity change. A two-layer strategy is proposed and tested to tackle the problem: (i) the inner layer estimates the costs of orbit transfers, through analytical models or optimization algorithms, while (ii) the outer layer finds the optimal encounter sequence. Two different scenarios are explored: (a) in the first scenario, polar orbits, not affected by the gravitational harmonic associated with the Earth oblateness, are addressed; (b) in the second scenario, non-polar orbits, subject to precession of their orbit planes (due to the Earth oblateness) are considered. With regard to scenario (a), several heuristic optimization methods, including tabu search, inver-over, and ant colony optimization, are incorporated in the outer layer, and their performance is compared. Different study cases, with increasing number of satellites, are investigated, and this leads to identifying both advantages and limitations of each approach. Specifically, the inver-over and tabu-search algorithms prove to outperform all other optimization methods in terms of computational efficiency and effectiveness in detecting the globally optimal solution. Scenario (b) involves orbits that are subject to differential precession related to the Earth oblateness. The orbits traveled by the satellites to service are visited by leveraging either direct transfers or intermediate drift orbits. The latter option is very useful for the purpose of minimizing the transfer cost, because natural alignment of different orbit planes is pursued through temporary injection into proper intermediate orbits. Instead, direct transfers are preferred when two orbit planes (nearly) coincide. The outer layer optimization combines a dedicated encoding and simulated annealing, which allow the simultaneous selection of the order of satellites to visit and the respective epochs. The methodology at hand is successfully applied to the Starlink constellation, including multiple mission scenarios, with horizon ranging from 1.5 to 4 years. The numerical simulations showcase the scalability and effectiveness of the optimization technique developed in this research. Compared to previous studies, this research features the following major contributions: (i) a systematic analysis of different optimization algorithms allows to compare their performance in scenario (a); (ii) a modified version of simulated annealing is proposed to enhance local exploration in scenario (b); (ii) suitable waiting time requirements are included to enable phasing and servicing operations; (iii) the combined use of either drift orbits or direct transfers is exploited, to reduce propellant usage and improve mission efficiency; (iv) a novel intuitive representation (i.e. the $\Delta\Omega$-plot) is introduced to visualize the structure and characteristic properties of the solution. In particular, drift-assisted transfers demonstrate significant advantages in scenarios where extended mission durations are allowed, thus testifying to the capability of incorporating practical operational requirements and dynamical constraints.

DOI: 10.13009/EUCASS2025-338



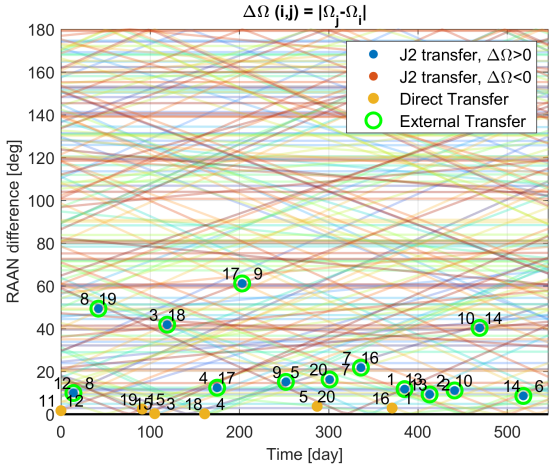



Figure 7: ΔΩ-plot for the 1.5-year mission.

| # dep | # arr | $t_{dep}$ [day] | $\Delta t$ [day] | $H_d$ [km] | $I_d$ [deg] | type | $\Delta V$ [m/s] |
|---|---|---|---|---|---|---|---|
| 11 | 12 | 0 | 0 | / | / | 0 | 181.7 |
| 12 | 8 | 14 | 14 | 883.26 | 53.310 | 1 | 361.5 |
| 8 | 19 | 42 | 35 | 1301.20 | 53.697 | 1 | 781.6 |
| 19 | 15 | 91 | 0 | / | / | 0 | 260.0 |
| 15 | 3 | 105 | 0 | / | / | 0 | 36.4 |
| 3 | 18 | 119 | 28 | 1229.80 | 53.735 | 1 | 760.0 |
| 18 | 4 | 161 | 0 | / | / | 0 | 48.6 |
| 4 | 17 | 175 | 14 | 976.09 | 53.466 | 1 | 461.6 |
| 17 | 9 | 203 | 35 | 1404.80 | 53.859 | 1 | 926.1 |
| 9 | 5 | 252 | 21 | 881.26 | 53.481 | 1 | 410.9 |
| 5 | 20 | 287 | 0 | / | / | 0 | 391.8 |
| 20 | 7 | 301 | 21 | 848.90 | 53.310 | 1 | 366.2 |
| 7 | 16 | 336 | 21 | 1062.20 | 53.550 | 1 | 579.2 |
| 16 | 1 | 371 | 0 | / | / | 0 | 307.4 |
| 1 | 13 | 385 | 14 | 945.61 | 53.529 | 1 | 434.3 |
| 13 | 2 | 413 | 14 | 850.44 | 53.458 | 1 | 335.7 |
| 2 | 10 | 441 | 14 | 924.50 | 53.428 | 1 | 409.0 |
| 10 | 14 | 469 | 35 | 1133.70 | 53.582 | 1 | 619.4 |
| 14 | 6 | 518 | 14 | 824.11 | 53.439 | 1 | 308.1 |
| MR: | 0.07 | | | | | | 7979.5 |

Table 9: Results for the 1.5-year mission.

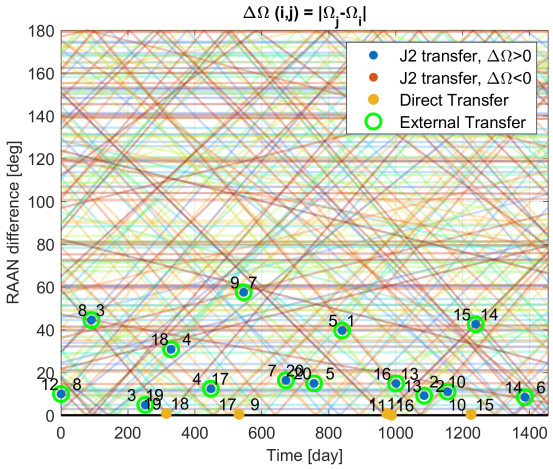



Figure 8: ΔΩ-plot for the 4-year mission.

| # dep | # arr | $t_{dep}$ [day] | $\Delta t$ [day] | $H_D$ [km] | $I_D$ [deg] | type | $\Delta V$ [m/s] |
|---|---|---|---|---|---|---|---|
| 12 | 8 | 0 | 77 | 603.07 | 53.097 | 1 | 61.9 |
| 8 | 3 | 91 | 147 | 677.90 | 53.239 | 1 | 149.5 |
| 3 | 19 | 252 | 49 | 582.32 | 53.249 | 1 | 47.4 |
| 19 | 18 | 315 | 0 | / | / | 0 | 43.0 |
| 18 | 4 | 329 | 105 | 674.70 | 53.221 | 1 | 187.8 |
| 4 | 17 | 448 | 70 | 620.63 | 53.197 | 1 | 88.1 |
| 17 | 9 | 532 | 0 | / | / | 0 | 43.0 |
| 9 | 7 | 546 | 112 | 716.15 | 53.288 | 1 | 263.4 |
| 7 | 20 | 672 | 70 | 630.72 | 53.143 | 1 | 132.7 |
| 20 | 5 | 756 | 70 | 636.22 | 53.221 | 1 | 114.3 |
| 5 | 1 | 840 | 119 | 688.17 | 53.333 | 1 | 163.2 |
| 1 | 11 | 973 | 0 | / | / | 0 | 36.4 |
| 11 | 16 | 987 | 0 | / | / | 0 | 36.4 |
| 16 | 13 | 1001 | 70 | 632.68 | 53.290 | 1 | 102.8 |
| 13 | 2 | 1085 | 56 | 610.73 | 53.274 | 1 | 78.7 |
| 2 | 10 | 1155 | 56 | 629.48 | 53.203 | 1 | 97.6 |
| 10 | 15 | 1225 | 0 | / | / | 0 | 42.1 |
| 15 | 14 | 1239 | 133 | 682.13 | 53.327 | 1 | 192.9 |
| 14 | 6 | 1386 | 56 | 605.12 | 53.270 | 1 | 72.5 |
| MR: | 0.521 | | | | | | 1953.5 |

Table 10: Results for the 4-year mission.

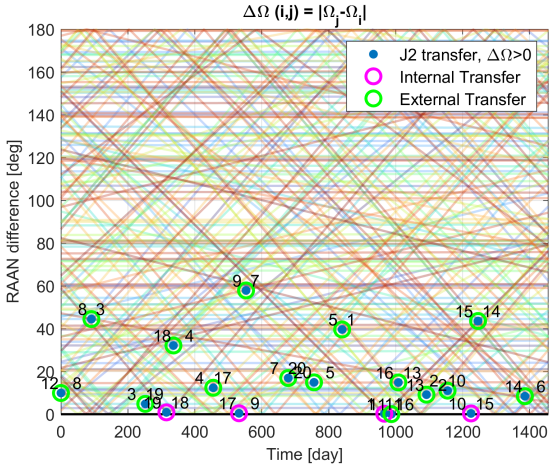



Figure 9: ΔΩ-plot for the 4-year mission with phasing penalization.

| # dep | # arr | $t_{dep}$ [day] | $\Delta t$ [day] | $H_d$ [km] | $I_d$ [deg] | type | $\Delta V$ [m/s] |
|---|---|---|---|---|---|---|---|
| 12 | 8 | 0 | 77 | 603.07 | 53.097 | 1 | 61.89 |
| 8 | 3 | 91 | 147 | 677.90 | 53.239 | 1 | 149.47 |
| 3 | 19 | 252 | 49 | 582.32 | 53.249 | 1 | 47.35 |
| 19 | 18 | 315 | 7 | 514.57 | 53.197 | -1 | 42.96 |
| 18 | 4 | 336 | 105 | 680.39 | 53.226 | 1 | 193.96 |
| 4 | 17 | 455 | 63 | 629.45 | 53.204 | 1 | 97.50 |
| 17 | 9 | 532 | 7 | 489.06 | 53.180 | -1 | 43.00 |
| 9 | 7 | 553 | 112 | 717.85 | 53.289 | 1 | 265.25 |
| 7 | 20 | 679 | 63 | 646.82 | 53.156 | 1 | 150.30 |
| 20 | 5 | 756 | 70 | 636.22 | 53.221 | 1 | 114.27 |
| 5 | 1 | 840 | 112 | 697.95 | 53.341 | 1 | 173.76 |
| 1 | 11 | 966 | 7 | 489.25 | 53.173 | -1 | 36.42 |
| 11 | 16 | 987 | 7 | 543.23 | 53.218 | 1 | 40.29 |
| 16 | 13 | 1008 | 70 | 632.69 | 53.290 | 1 | 102.80 |
| 13 | 2 | 1092 | 49 | 621.14 | 53.282 | 1 | 90.13 |
| 2 | 10 | 1155 | 56 | 629.48 | 53.203 | 1 | 97.58 |
| 10 | 15 | 1225 | 7 | 499.67 | 53.124 | -1 | 42.06 |
| 15 | 14 | 1246 | 126 | 694.52 | 53.336 | 1 | 206.33 |
| 14 | 6 | 1386 | 56 | 605.12 | 53.270 | 1 | 72.47 |
| MR: | 0.509 | | | | | | 2027.80 |

Table 11: Results for the 4-year mission with phasing penalization.

DOI: 10.13009/EUCASS2025-338



## References


[1] Isabella Hatty. Viability of on-orbit servicing spacecraft to prolong the operational life of satellites. *Journal of Space Safety Engineering*, 9(2):263–268, 2022.

[2] National Aeronautics and Space Administration. On-orbit satellite servicing study project report. Technical report, Goddard Space Flight Center, October 2010. Accessed on October 16, 2024.

[3] Roberto Opromolla, Dmitriy Grishko, John Auburn, Riccardo Bevilacqua, Luisa Buinhas, Joseph Cassady, Markus Jäger, Marko Jankovic, Javier Rodriguez, Maria Antonietta Perino, et al. Future in-orbit servicing operations in the space traffic management context. *Acta Astronautica*, 220:469–477, 2024.

[4] Wenfu Xu, Bin Liang, Bing Li, and Yangsheng Xu. A universal on-orbit servicing system used in the geostationary orbit. *Advances in Space Research*, 48(1):95–119, 2011.

[5] Jean-Marie Bourjolly, Ozgur Gurtuna, and Aleksander Lyngvi. On-orbit servicing: a time-dependent, moving-target traveling salesman problem. *International Transactions in Operational Research*, 13(5):461–481, 2006.

[6] Dario Izzo, Ingmar Getzner, Daniel Hennes, and Luís Felismino Simões. Evolving solutions to tsp variants for active space debris removal. In *Proceedings of the 2015 annual conference on genetic and evolutionary computation*, pages 1207–1214, 2015.

[7] Adrian Barea, Hodei Urrutxua, and Luis Cadarso. Large-scale object selection and trajectory planning for multi-target space debris removal missions. *Acta Astronautica*, 170:289–301, 2020.

[8] Laura Medioni, Yvan Gary, Myrtille Monclin, Côme Oosterhof, Gaetan Pierre, Tom Semblanet, Perrine Comte, and Kévin Nocentini. Trajectory optimization for multi-target active debris removal missions. *Advances in Space Research*, 72(7):2801–2823, 2023.

[9] Danilo Zona, Alessandro Zavoli, Lorenzo Federici, and Giulio Avanzini. Evolutionary optimization for active debris removal mission planning. *IEEE Access*, 11:41019–41033, 2023.

[10] Max Cerf. Multiple space debris collecting mission: optimal mission planning. *Journal of Optimization Theory and Applications*, 167(1):195–218, 2015.

[11] David A. Vallado. *Fundamentals of Astrodynamics and Applications*. Microcosm Press, 4th edition, 2013.

[12] Mauro Pontani and Bruce A. Conway. Particle swarm optimization applied to impulsive orbital transfers. *Acta Astronautica*, 74:141–155, 2012.

[13] Riccardo Santoro and Mauro Pontani. Orbit acquisition, rendezvous, and docking with a noncooperative capsule in a mars sample return mission. *Acta Astronautica*, 211:950–962, 2023.

[14] Edoardo Maria Leonardi, Giulio De Angelis, Mauro Pontani, et al. Guidance strategies to deploy a lunar satellite constellation from gateway. In *Proceedings of the 75th international astronautical congress*, pages 1–15. International Astronautical Federation (IAF), 2024.

[15] Lorenzo Federici, Alessandro Zavoli, and Guido Colasurdo. A time-dependent tsp formulation for the design of an active debris removal mission using simulated annealing. *arXiv preprint arXiv:1909.10427*, 2019.

[16] Seyedali Mirjalili. Ant colony optimization (aco). https://www.mathworks.com/matlabcentral/fileexchange/69028-ant-colony-optimiztion-aco, 2024. MATLAB Central File Exchange. Accessed on October 7, 2024.

[17] Markus Buehren. Differential evolution. https://www.mathworks.com/matlabcentral/fileexchange/18593-differential-evolution, 2024. MATLAB Central File Exchange. Accessed on October 7, 2024.

[18] Mostapha Heris. Tabu search (ts). https://www.mathworks.com/matlabcentral/fileexchange/52902-tabu-search-ts, 2024. MATLAB Central File Exchange. Accessed on October 7, 2024.

[19] Hamid Salimi. Stochastic fractal search: a powerful metaheuristic algorithm. *Knowledge-based systems*, 75:1–18, 2015.

[20] Mostapha Heris. Simulated annealing (sa). https://www.mathworks.com/matlabcentral/fileexchange/52896-simulated-annealing-sa, 2024. MATLAB Central File Exchange. Accessed on October 7, 2024.

[21] CelesTrak. Norad two-line element sets. https://celestrak.org/NORAD/elements/, 2024. Accessed on October 31, 2024.