IAC-22-C1,5,10,x71577

Analysis of Multi-Orbit Multi-Payload Injection Scenarios for an Upper Stage

Iñigo Alforja Ruiz[a]*, Michèle Lavagna[b], Juan Carlos Bastante[c]

[a] *Politecnico di Milano, Aerospace, Science & Technology Dept., via La Masa 34, 20156, Milano, Italy*, inigo.alforja@polimi.it
[b] *Politecnico di Milano, Aerospace, Science & Technology Dept., via La Masa 34, 20156, Milano, Italy*, michelle.lavagna@polimi.it
[c] *OHB System AG, Mission Analysis and Flight Dynamics Team, Universitätsallee 27, 28359, Bremen, Germany*, juan_carlos.bastante@ohb.de
* Corresponding Author

Abstract

In the recent years, the number of launches has increased dramatically, showing a tendency beyond the current space transport systems. Such a problem, partially provoked by the increased number of space users due to the industry inclusion, as well as by the shift of interest towards smaller satellites and constellation missions, demands of innovative and economical solutions. One of these is the capability of an upper stage to directly inject multiple satellites into their respective differentiated orbits, reducing this way the number of necessary launches while allowing the growth of the space environment usage.

Such multi-payload multi-orbit injection trajectory requires of a control law that can provide the manoeuvres while minimising both the fuel consumption and the overall mission time. Its definition is not straight forward and requires solving a complex optimisation problem composed by the visiting sequence and the individual transfers. The current paper proposes a strategy to define such a trajectory by diving the problem into two: a preliminary bi-level bi-objective optimisation algorithm that determines the ideal orbit order and the approximate necessary Lambert transfers.

The result is a flexible algorithm that can provide for the full set of transfers given a set of orbits and satellites to be delivered, regardless of the final injection orbits and mass properties of the payloads. This algorithm is then used to analyse a certain payload injection scenario, given realistic mission data from the space industry. The performance of the upper stage in terms of total time and consumed fuel is examined, providing an assessment on the feasibility of this type of mission. The same case scenario is then studied under different conditions in terms of randomness, decision criteria and routing constraints to achieve further understanding on the possible real case scenarios. It is shown that in all cases the algorithm is able to converge towards feasible solutions, and that the nature of the resulting trajectories depends highly on the scenarios themselves, but also on the logical operation of the bi-level optimisation loop.

Keywords: Lambert, Optimisation, Traveling Salesman Problem, Multi-Rendezvous

Nomenclature

- a = semi-major axis [m]
- e = eccentricity [-]
- F = penalty factor [-]
- g_0 = gravitational acceleration at Earth's surface [m/s^2]
- i = inclination [deg]
- i, j = generic numbering of nodes [-]
- I_{sp} = specific impulse [s]
- m = mass [kg]
- p = probability [-]
- $\boldsymbol{r}$ = position vector [m]
- r = distance magnitude [m]
- R_E = radius of the Earth at the equator [m]
- t = time [s]
- T = orbital period [s]
- ΔV = impulsive manoeuvre magnitude [m/s]
- η = heuristic measure [-]
- θ = true anomaly [deg]
- λ = penalty function [variable unit]
- μ = gravitational parameter of the central body [m^3/s^2]
- τ = pheromone concentration [-]
- ω = argument of perigee [deg]
- Ω = right-ascension of ascending node [deg]

Acronyms/Abbreviations

Ant Colony Optimisation (ACO)
Active Debris Removal (ADR)
First-in First-out (FIFO)
Genetic Algorithm (GA)
On-Orbit Servicing (OOS)
Mixed Integer Nonlinear Optimisation Problem (MINLP)
Multi-Objective Particle Swarm Optimisation (MOPSO)
Population-based Ant Colony Optimisation (P-ACO)
Particle Swarm Optimisation (PSO)
Time of Flight (TOF)
Traveling Salesman Problem (TSP)

1. Introduction

The development of society in the last years has been tightly coupled with that of technologies related to communication and access to information. This has made the dependence on spaceborne systems devoted to such activities crucial for today's way of living, entailing an increasing need of satellites orbiting the planet. In fact, current predictions estimate the growth rate in the number planned small satellite launches in this decade to be four times that of the previous decade [1]. The increasing interest in constellation missions, in addition to the higher accessibility to space for modest users, is pushing further the aforementioned growth, leading the space infrastructure towards a logistically and ecologically unsustainable position. New and efficient ways to deliver payloads into orbit are clearly necessary.

Current methods for multiple satellite delivery involve piggybacking strategies, which limit the flexibility of launch and injection for non-primary loads. Such practice could discourage new projects, particularly those coming from the private sector, preventing the expansion of this market. The new way of accessing space efficiently should then focus on the injection of multiple payloads into dedicated and different orbits. A similar strategy has been tested using kick-stages which behave as intermediary vehicles between the primary orbit and those of the secondary small satellites (for instance the Small Launch Orbital Manoeuvring Vehicle [2], or the Sherpa-NG project of Spaceflight Inc. [3]). Thus, multiple-payload multiple-orbit delivery has been proven feasible, becoming a concept that could drastically change the way in which launches are planned and performed.

To define the trajectory necessary to deliver multiple payloads into different orbits, both the order of injection and the individual transfers must be specified. This multi-rendezvous problem has been studied previously, mainly for missions related to Active Debris Removal (ADR) and On-Orbit Servicing (OOS), in which the distinction between both mentioned problems is clearly distinguished. Generally, these problems are separated, and different strategies are used to solve them individually, coupling them at a certain point in the optimisation process. Regarding the visitation sequence problem, its combinatorial nature has led to identify it as a specific case of the Traveling Salesman Problem (TSP) [4]. As such, typical solutions to this problem have been used. On the one hand, one can find the use of extensive search algorithms such as in in Chen et al. [5] or Daneshjou et al. [6]. These methods, while providing for a global optimum to the sequence selection, require of high computational efforts to calculate of possible permutations, reaching unfeasible levels when the number of orbits to be visited is medium-to-high. As an alternative, tree-search algorithms have been proposed in a way that the different sequences or "branches" are extended or cut based on some criteria. Two interesting examples are the Series Method [7] or the Branch-and-Bound strategy [8-11]. However, similar to the previous methods, after a certain number of orbits, the computational effort becomes prohibitive. To overcome this issue, heuristic algorithms have been proposed providing with a sub-optimal solution but at lower computational times. Among these, the Ant Colony Optimisation (ACO) is frequently used as its structure resembles the tree-shape structure characteristic of the combinatorial problem [12-14]. Other heuristic methods proposed include evolutionary algorithms such as the Genetic Algorithm (GA) [8,15] or Simulated Annealing [16,17].

As stated before, the multi-rendezvous problem entails a second part related to the definition of the individual transfers given a certain sequence. Typically, the cost or ΔV associated to a certain transfer is estimated or pre-computed and stored in a matrix to be accessed by the combinatorial solver. The cost is normally simplified to be time-independent, such that moving from an orbit to another one has a certain constant cost, normally that of a Hohmann transfer [10,18]. However, a more realistic computation requires the time to be taken account. This can be done by generating a time grid and computing all possible costs between orbits given an initial and final time using Lambert manoeuvres [9,11,17], and then storing them in higher-dimension arrays. This method allows for a fast computation of the costs but requires large data storage. Among the different strategies, Bang et al. [19] shows a more efficient method of pre-computation by looking for the local minima in each of the transfers, which are the only ones stored and considered in the overall optimisation, reducing the search space. None of these strategies consider the optimisation of the transfers themselves, thus reaching solutions of generally less quality. In fact, several studies have shown that optimising the transfers within the complete problem can improve the overall solution. This complete optimisation is normally done through heuristic algorithms such as GAs [15] or evolutionary algorithms [8,16]. These methods, however, require of the discretisation of time, therefore not considering its continuous nature. To overcome this, other algorithms are considered, among which the most interesting is the Particle Swarm Optimisation (PSO) [5,6,20].

Nevertheless, these strategies rely on building up the trajectory by adding the individual optimisations of the transfers one by one, in a similar manner to the (pseudo-)exact methods of the combinatorial problem, instead of considering the complete visitation sequence and set of transfers as a whole. Therefore, the optimisation of each transfer is influenced by the decisions and values of the previous ones, limiting the search space. In addition, it behaves similar to a greedy algorithm which considers the best immediate solution at each step, disregarding those that (even if they have a higher cost) would lead to overall smaller values of the cost function.

The paper focuses on the solution of a time and fuel mass constrained multi-rendezvous mission, assuming impulsive manoeuvres, with the objective of visiting all required orbits minimising both fuel and time. An optimisation tool has been developed as a bi-level solver conformed of a Population-based ACO (P-ACO) for the combinatorial problem, and a Multi-Objective PSO (MOSPO) for the set of transfers. The algorithm is then used to study a typical multi-injection case scenario under different considerations for its analysis. To do so, first the mission and the mathematical problem are stated in Section 2. Then, the tool is presented in Section 3. Section 4 will show the solution to the case scenario and the results of the different analyses performed. Finally, Section 5 will present the main conclusions and steps forward.

2. Problem Definition

Before presenting the optimisation tool for the multi-rendezvous problem, it is necessary to define the specific mission under consideration and its subsequent mathematical formulation and modelling.

2.1 Mission Definition

Let us consider an upper stage which is used as means to deliver a set of N payloads into N distinct orbits or positions in space. These orbits, as well as the masses of the payloads, are specified before-hand based on the requirements of the missions related to their deployment. In this study, it is assumed that the trajectory definition starts at the orbit of the first deployed satellite, once it has been released, in a way that the launch cost is not included as part of the cost function as it is assumed that the differences in ΔV from Earth to any orbit are negligible. However, the selection of the first orbit (the initial one) is given to the optimiser as part of the sequence definition. Additionally, it is considered that, in order to comply with the mitigation guidelines on space debris, the upper stage is to finish its trajectory at a previously designated disposal orbit. The optimisation of this disposal orbit (related to the parameters ensuring a certain orbital decay, security, etc.) is not within the scope of this work. Therefore, an arbitrarily chosen orbit that fulfils these requirements, and which is “close” to the set of target orbits, will be used.

Thus, the complete mission is stated as follows: the vehicle, after deploying the first payload, moves into the following target orbit where it releases the next satellite. It then waits until the next manoeuvre to reach the following orbit, making sure that this waiting time is enough to perform the injection activity. This sequence is the repeated until all satellites are injected, after which the vehicle transfers to the final disposal orbit, at which the mission is over. Generally, the target is the orbit itself and not a specific point within it (unless several satellites within the same orbit and with a certain phase difference among them is required). Therefore, the point at which the transfer arrival is performed is not crucial, and all points are considered to be equally interesting when performing the optimisation.

When dealing with the deployment of satellites, and considering the point of view of possible customers, the former are desired to be operational as fast and cheap as possible, within a certain threshold. As such, the solution should minimise both the total mission time and the fuel mass consumption, ensuring that they are always within the allowed margins given by the requirements. Finally, it must be noted that, since upper stages usually operate with high thrust chemical propulsion systems, all firings of the main engine are modelled as impulsive manoeuvres.

2.2 Mathematical Formulation

As stated in Section 1, the problem of defining an optimal visitation sequence falls into those categorised as TSP. In this problem, a certain salesperson needs to visit a pre-defined set of cities only once, while minimising the total cost of the trip (either time, distance, fuel consumption, etc.) and finishing at the starting city. Such a problem is translated into the graph problem $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with $\mathcal{V} = \{1, \dots, N\}$ being the set of vertices or nodes; and $\mathcal{A} = \{(i,j) \in \mathcal{V}, i \neq j\}$ being the set of links between those nodes. However, the multi-rendezvous problem present significant differences with respect to the typical TSP, some of which have been already lightly mentioned before:

1) The problem is time-dependent due to the nonlinearity of the dynamics involved in the motion of the vehicle, in a way that the tarting and ending points in time for a certain transfer will affect the cost of such link.
2) The problem is open-route, meaning that the vehicle will start and finish at different orbits.

By defining then the set of payloads to be delivered as with $\mathcal{S} = \{1, \dots, N\}$, the set of vertices to be visited becomes $\mathcal{V} = \{0, \mathcal{S}, N+1\}$, where $i = 0$ has been included for the sake of completeness in the case that the initial vertex is the launching site or a parking orbit. The last node included is, as expected, the final disposal orbit. As stated by point 1, time influences the problem, not only within the equations of motion, but as a direct involvement in the possible arcs between two nodes and their associated cost. This characteristic of the problem can be understood with the schematic shown in Fig. 1, in which some connections are represented, among the (infinite) range of possibilities for each one. Properly including the time within the problem is crucial to accurately solve the optimal transfers in between two consecutive orbits. To do so, the initial and final position, as well as the time-of-flight (TOF), are optimisation variables for each one of the transfers. Nevertheless, the order in which the transfers are performed will also affect the solution, as time will change but also the set of possible nodes to be visited will decrease, enabling or not certain paths. In fact, by solving the transfers sequentially, all subsequent manoeuvres will be constrained to the time and fuel mass left by the previous one. To counteract this problem, it is necessary to optimise the complete sequence in terms of both variables. This characteristic highlights the complex coupling of both mathematical problems being discussed.

The presented optimisation problem, involving both the sequential and the transfer parts, can be mathematically stated as:

$$min\left\{\sum_{i=1}^{N+1} m_{f,i}\,, t_{tot}\right\} \tag{1}$$

In a way that each transfer is dominated by the dynamic equations of motion [21]:

$$\dot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r} + \frac{T}{m}\boldsymbol{e_T}\delta + \boldsymbol{f_{dist}} \tag{2}$$

$$\dot{m} = -\frac{T\delta}{I_{sp}g_0} \tag{3}$$

Where T is the thrust magnitude, $\boldsymbol{e_T}$ the thrust direction vector, δ the main engine relay on-off function, and $\boldsymbol{f_{dist}}$ the acceleration due to environmental disturbances. However, for the purposes of the current study, some simplifications are considered. On the one hand, all engine firings are modelled as impulsive manoeuvres for which the ΔV is estimated accordingly. Two kinds of motion are considered in this paper: Lamber targeting, and phasing within a certain orbit. On the other hand, all disturbances are neglected, and ideal Kepler orbits are assumed. In addition, the problem is subject to the constraints:

$$\sum_{i=0}^{N+1} s_{i,j} = 0\,;\ \sum_{j=0}^{N+1} s_{i,j} = 0 \tag{4}$$

$$\sum_{k=1}^{N+1} m_{f,k} \leq m_{f,max}\,;\ \sum_{k=1}^{N+1} t_k \leq t_{max} \tag{5}$$

Equations (4) are related to the combinatorial problem and ensure that each orbit is visited only once. Equations (5) establish the constraint in terms of maximum fuel mass consumed and mission time given for the specific trajectory. The complete mathematical formulation of the problem enables to easily spot the two differentiated sub-problems already mentioned. On the one hand, one has the optimisation of the transfers given by Equations (2) and (3), which falls into the category on Nonlinear Continuous Programming. On the other hand, there is the combinatorial problem related to the visitation sequence with the single-time visiting constraint given by Equations (4), which falls into the category of Integer Programming. Being both tightly coupled, the complete mathematical problem can be categorised as a mixed integer nonlinear programming (MINLP) case. This problem does not have straightforward solutions and other strategies need to be considered to solve both the continuous and the integer sub-problems.

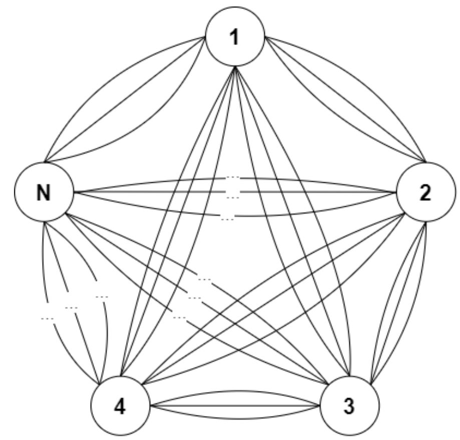

Fig. 1. Schematic of the TSP time-dependent links

3. Bi-level Optimisation

The MINLP optimisation problem can be classified among the NP-hard type, meaning that it needs computational times which grow in a factorial manner with the number of orbits to be visited to solve it in a deterministic way. Therefore, heuristic algorithms that reach good sub-optimal solution at reasonable times are of interest for the problem under consideration. In addition, the complexity of the overall problem can be reduced by separating the two sub-problems mentioned in Section 2 and using different strategies for each of them according to their nature, and hose solutions can be put together to achieve the final optimal trajectory.

This paper follows such a line of thought, so that a bi-level optimisation algorithm is proposed, in a way that each level solves one of the sub-problems while their coupling is kept. The structure of this approach is as follows: an internal layer solving the transfer optimisation problem among all orbits, given a specific sequence; and an outer layer dealing exclusively with the combinatorial visitation order problem. As stated before, these are connected, as they rely on each other to achieve a proper solution to the MINLP problem. In fact, the inner layer needs a certain sequence given by the outer level to calculate the transfers among orbits; while the outer level uses the time and total fuel mass consumption associated to a certain sequence calculated by the inner level as cost function. To achieve this interconnection, the nested structure allows for easy information transfer.

However, an issue arises regarding this transfer of information as, as stated previously, there are two objectives to be minimised. The strategy followed in the algorithm is such that both the inner and the outer loop make use of Pareto dominance to establish the possible Pareto front optimal solutions for their respective problems. Thus, for a given sequence, one would get an array of possible solutions instead of a pair of values, which would make the comparison with other sequences difficult. A decision function is used to pick the cost pair solution for a given sequence from the inner loop to be forwarded to the outer loop, which will be further detailed in Section 3.3.

3.1 Cost of a Single Transfer

Before diving into the algorithm structure, it is important to show how the single transfers between two consecutive orbits are computed in terms of ΔV, fuel mass and TOF. Two types of manoeuvres are envisaged and will be presented in this subsection: Lambert targeting, and phasing. The selected manoeuvre at any transfer is picked based on a comparison of the Kepler elements among the two consecutive orbits. This set of variables are used as state representation as they are the ones used to define operational orbits for the satellites when designing a mission.

In the most general transfer, an impulsive Lamber manoeuvre is used. This boundary-value problem looks to find the initial and final velocity of a particle in a 2-body system needed to go from a specific initial position to a final one given a certain TOF [22]. Being a classic problem in space, several solutions have been presented along history. Among these, the most famous one is the one proposed by Gooding [23], which is based on that of Lancaster and Blanchard [24] with some variations, due to its robustness. A more efficient algorithm was proposed by D. Izzo [25], which trades a faster solution for less robustness. The present study uses a mixture of both algorithms in a way that, as a general rule, Izzo's strategy is used, unless it is unable to converge until a number of iterations, after which Gooding's algorithm is used. Therefore, all calls to the Lambert solver will reach a solution, and the slower strategy is only called when the faster one fails, speeding up the process. This implementation has been based on that of R. Oldenhuis [26].

In the case of constellation missions where several satellites are to be injected within the same orbit, a phasing manoeuvre is used. This strategy consists of a first impulsive firing to reach the phasing orbit, in which the upper stage waits until correct phasing is achieved, after which it performs a second firing. These are performed at the apogee or perigee of the original orbit to reach a phasing orbit with smaller or higher orbital period depending on whether the final position is ahead or behind, respectively [27]. Considering t_{AB} to be the time needed to close the gap, the semi-major axis of this intermediate orbit is obtained by first calculating its necessary orbital period:

$$T_{ph} = T_1 - \frac{t_{AB}}{N_{rev}} \tag{6}$$

$$a_{ph} = \sqrt[3]{\mu (T_{ph}/2\pi)^2} \tag{7}$$

This value allows to obtain the remaining orbital parameters, allowing to calculate the required ΔV and the TOF based on the number of revolutions N_{rev}.

3.2 Outer Layer

As stated at the start of this Section, the outer layer is in charge of solving the combinatorial problem, using the inner level as the cost function for each sequence. A Population-based ACO (P-ACO) was used for this purpose due to its tree-shape structure (typical also of combinatorial problems). Such a strategy was introduced in Ref. [28] and expanded towards multi-objective problems in Ref. [29]. As an advantage of its use, it has been already proven to work in the domain of multi-rendezvous problems by L. Simões et al. [30].

The algorithm is based on the basic concept of ACO strategies, that is an ant travels a certain path given by a sequence, leaving a trail of pheromones on it, denoted by τ. The probability of another ant to take a certain path will be directly related to the amount of pheromones left by the previous generation of ants on the different paths. In addition, the problem considers another measure of "desirability" for an ant to pick a specific track, which is given by a heuristic value η related to the problem. For this case, the heuristic picked is an estimate of the theoretical ΔV to change individually each of the Kepler elements among two orbits, using approximate analytic equations [29]. Both the pheromone concentration and heuristic are scaled in relative importance at decision-making by means of exponential factors. With these two values, the probability for an ant at node i to move towards node j, which can be any subset S of nodes still unvisited is:

$$p(i,j) = \frac{\tau(i,j)^{\alpha} \eta(i,j)^{\beta}}{\sum_{z \in S} \tau(i,z)^{\alpha} \eta(i,z)^{\beta}} \tag{8}$$

The difference of P-ACO with respect to the basic ACO is that pheromones are not deposited by all ants, but rather by a subset of ants, or *Population*, composed of the beast ants of each generation. These ants enter the population in a FIFO-queue manner. In the specific case of a bi-objective optimisation problem, the population is composed by the members of the Pareto Front of the previous generation, as proposed in Ref. [30]. This set of solutions, or *Elite*, is emptied and updated at each

Algorithm 1 P-ACO

```
Initialize colony parameters
Initialize pheromone matrix τ with τ_min
for i=1:N_generations do
    generation = [ ]
    for j=1:N_ants do
        if start_node != [ ] then
            tour(1) = start_node
        else
            tour(1) = random
        end if
        stop = 0
        while stop==0 do
            ph ← pheromone(tour)
            heur ← heuristic(tour)
            p = ph **alpha * heur ** beta
            next_node ← roulette_wheel(tour,p)
            tour = [tour next_node]
            if length(tour)==n_cities then
                stop=1
            end if
        end while
        cost ← evaluate(tour)
        generation(j) = [cost tour]
    end for
    elite ← sort(elite,generation)
    best = elite
    elite_pheromone ← pheromone_update(elite)
end for
OUTPUT : elite
```

Fig. 2. Pseudo-code of P-ACO

generation, such that it is conformed of all non-dominated solutions used to recalculate the pheromone matrix. The algorithm implementation has been done as a modification of the one provided as open-source code developed by L. Simões et al. [30]. A pseudo-code is shown in Fig. 2 as a summary of the main steps.

3.3 Inner layer

The evaluation of the fitness of each of the ants is made by calling the inner layer of the algorithm, which computes the optimal set of impulsive manoeuvres given the sequence that determines the ant's path. To solve this problem, a heuristic algorithm is used, being the one selected the Multi-Objective PSO (MOPSO). This strategy was first proposed by C. Coello [31], and the main difference with respect to the single objective PSO is the definition of the *leader* particle. In this case, it is not a single leader, but a set of non-dominated particles kept in a repository which lead the remaining particles. At each iteration, a random particle is selected from this repository which serves as leader for the new generation. The algorithm has been proven to work in the multi-rendezvous problem Daneshjou et al. [6] and was selected due to its ability to exploit the continuous time domain, as opposed to the typical time-grid-strategies. The implementation was modified from the open-source code of V. Martínez-Cagigal [32]. A pseudo-code is presented in Fig. 3. The equation of motion for the particles is included, being each of the contributions weighted by a certain importance factor and a random number.

Algorithm 2 MOPSO algorithm

```
Initialize swarm parameters
Initialize swarm position, velocity, fitness
Establish first repository of leaders
for i=1:N_generations do
    h ← select_leader
    VEL = w*VEL+c1*r1*(PBEST-POS)+
    +c2*r2*(h-POS)
    POS = POS + VEL
    POS ← mutation
    POS_FIT ← evaluate(POS)
    REP ← update_repository
    PBEST ← update personal best
end for
OUTPUT : BEST_POS
```

Fig. 3. Pseudo-code of MOPSO

The MOPSO is called to optimise a certain function, which is to be constructed based on the different orbits to be visited and their order. In fact, the optimisation variables depend on which manoeuvre is taking place, Thus, once an ant has selected a tour, the inner function is called which constructs the different manoeuvres to be followed and specifying the set of variables to be used, as well as their limits (to restrict the search space to reasonable bounds). All these are stacked into an array of decision variables, which is the one to be optimised. Once this is done, the objective function is fully constructed based on these on-line defined manoeuvres. For each transfer, both the ΔV and TOF are calculated based on the type of manoeuvre, checking that 1) the transfer orbit is elliptical (ensuring that the transfer time is greater than Barker's time [33]), and 2) the vehicle does not crash into the atmosphere. The fuel consumed is then computed using Tsiolkovsky's equation:

$$m_f = m_0\left(1 - exp\left(\frac{-\Delta V}{I_{sp} g_0}\right)\right) \tag{9}$$

It must be noted that at each new leg, the initial mass m_0 is that of the upper stage after the previous leg, minus the mass of the payload deployed at the last visited orbit. This means that the mass evolution of the upper stage is discrete at some points, at it must be accounted for in the generation of the cost function. After all transfers are completed, constraints in maximum TOF and fuel mass are included via penalty functions of the type:

$$\lambda(f, L) = (max\{0, f - L\})^2 \tag{10}$$

$$f = f + F \cdot \lambda(f, L) \tag{11}$$

The cost output by the cost function is then the pair of TOF and fuel mass, plus the value of the penalty function, ensuring that the particles move by their own means towards feasible ranges. This on-line construction of this cost function allows for increased flexibility on the range of possible missions and does not require of previous knowledge among the possible sequences to construct the cost of any trajectory. In addition, by limiting the generation of the sequence of transfers to this part, it allows to include any other possible manoeuvre without needing to modify any other part of the optimisation process.

However, as stated at the introduction of this Section 3, the output of the MOPSO algorithm is a set of non-dominated particles instead of one. Since a single cost pair is to be attributed to a single ant, a decision has to be taken on which particle from the Pareto Front is given to the tour as its fitness. Different logic strategies are implemented that can be followed:

1) The minimum fuel solution
2) The minimum time solution
3) Random solution
4) Solution that maximises the compliance of the trajectory both in terms of fuel mass and TOF.

This last option is done by establishing a weighted function which considers the consumed fuel and time with respect to the maximum allowable and outputs a value from 0 to 1 depending on how much of both is remaining at the end of the mission. This function is as introduced by L. Simões et al. [30], and it is generated by firstly computing the remaining fraction of fuel and time (Equations (12)). With these values, the weights (or importance) given to each one are calculated in a way that the variable which has consumed the most fraction of its maximum allowed value is considered more important than the other one (Equations (13)). This logic follows by an effort to punish solutions close to not fulfilling the constraints and rewarding those which leave safe margins in terms of both variables.

$$m = \frac{m_{f,max} - m_f}{m_{f,max}}; \ t = \frac{t_{max} - t}{t_{max}} \tag{12}$$

$$w_m = 1 - \frac{m}{m+t}; \ w_t = 1 - \frac{t}{m+t} \tag{13}$$

$$A = w_m \cdot m + w_t \cdot t \tag{14}$$

The weighted function (Equation (14)) is then constructed with both variables, and the outputted fitness is the non-dominated solution which maximises A.

4. Analysis of Case Scenarios

The tool is then to be used in solving a complex case of multi-orbit visitation to prove its validity and analyse the different possibilities that it proposes depending on different criteria. For this purpose, a mixture of part of a constellation mission and several nanosatellites is considered, based on historical data. For the constellation satellites, it was decided to include 3 satellites similar to the Starlink constellation, for which a mass of 200 kg was assumed. For their operational orbit, the elements of the 4th shell were used [34], which are included in Table 1. Regarding their injection, an approximate 6-degree true anomaly phase difference among the satellites is required, based on the population of each orbit. On the other hand, for the nanosatellites, the history of already successfully launched university- and institution-owned payloads was studied. The distribution of the semi-major axis altitude (subtracting the Earth radius), eccentricity, inclination, and mass of these payloads is shown in Fig. 4. It is observed how their masses are generally below 5 kg, with a few exceptions; and their orbits tend to be quasi-circular. A wider variety is observed when it comes to their altitudes, being two sectors the more interesting ones: in the 400–500-kilometre range; and in the 600-700-kilometre range. Similarly, two inclination magnitudes are mainly used: around 60 degrees; and near 100 degrees. In fact, a correlation is observed among both characteristics, as lower inclination orbits tend to correspond to lower altitude orbits, while higher inclinations relate to higher altitudes. This is due to the purpose of these orbits, which is easily seen by, for instance, the Sun-Synchronous regime. For the case under consideration, it was decided that 7 random satellites in the near-100-degree area would be picked. Their main orbital and payload characteristics are given in Table 1.

Table 1. Target orbits and associated payloads

Index	a-R_E [km]	e [-]	i [deg]	m [kg]
1-3	557.1	0	97.70	200
4	720.55	0.0095	98.10	1
5	659.1	0.0060	97.93	26
6	409.6	0.0008	97.44	3.2
7	678.6	0.0039	98.20	10
8	723.6	0.0006	98.28	1
9	726.6	0.0104	98.10	1
10	631.6	0.0014	97.90	1

The study is then focused on solving the given multi-rendezvous problem, analysing the effect of random operations within the optimisation algorithm, the influence of the criteria used to attribute a certain cost to

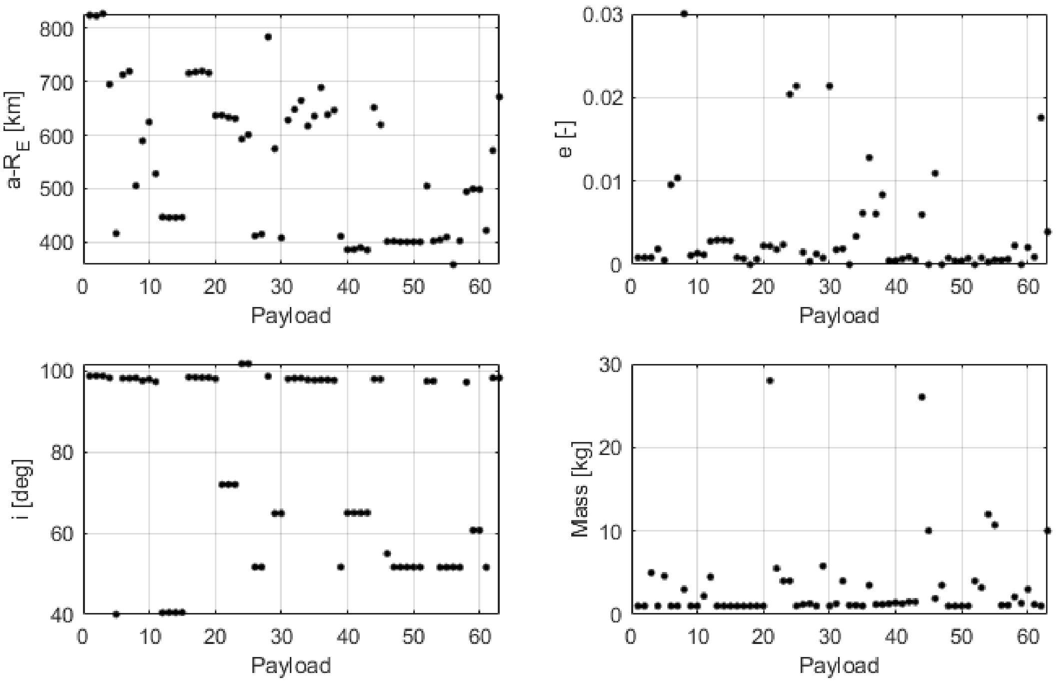

Fig. 4. Historical distribution of the a, e, i, and payload mass of considered nanosatellites

an ant, and finally the impact of "forcing" the algorithm to start the sequence in a specific orbit. The results of the subsequent studies are given in the following subsections.

In all cases, the upper stage considered is inspired on the design of AVUM, with a dry mass of 660 kg and a specific impulse of 320 s [35]. However, due to the higher manoeuvring load, higher fuel masses are considered. Also, the disposal orbit used in the current study has been arbitrarily picked to be circular with a 300 km altitude, and with an inclination of 1.71 radians (which is about 97.976 degrees), such that it is geometrically located near all the target orbits. It must be considered, also, that for the sake of simplicity of this problem, all RAANs and arguments of perigee are set to zero.

4.1 Solution to the base problem

The single solution of the problem is obtained for two cases: one in which the maximum fuel mass is 2500 kg and another one with a maximum fuel mass of 2000 kg. This is done to observe the compliance of the algorithm to respect boundaries and study the effect of reduced fuel mass and its impact on the total weight of the system. The results can be observed in Fig. 5.

As a first remark, it can be easily seen how the overall fuel masses needed in the second test are lower (due to the more stringent maximum fuel mass), while showing a larger spread in terms of total TOF of the mission. In fact, better results can be obtained with a lower amount of fuel mass, even though it could be expected that, having a greater amount of fuel, the upper stage could "allow" itself for faster and more expensive manoeuvres. This behaviour can be attributed to the fact that the extra 500 kg in fuel mass that must be carried by the upper stage generate a detrimental effect on the performance of the impulsive manoeuvres, as higher mass fuel is needed to perform the same change in ΔV (Equation (9)). A break-even point must there exist in which the mounted fuel is the exact needed (plus some safety margin) to perform all the manoeuvres without excessive remaining unused mass to achieve an optimal solution. Nevertheless, it can be seen how the optimiser converges towards feasible solutions.

Regarding the generated solutions, certain trends were observed (which had been previously discussed in Ref. [36]). As such, in both cases the lowest fuel mass consumption trajectories involve delivering the three heavier constellation satellites, which are in the same orbit, as soon as possible (sometimes the first ones, sometimes after delivering a previous nanosatellite). In this way, the total mass of the upper stage and its payloads is reduced, and the subsequent manoeuvres require less propellant to perform. Nevertheless, as these phasing manoeuvres are generally more time consuming, the algorithm proposes solutions of intermediate transfers and/or phasing (without the three of the constellation satellites being delivered sequentially) to achieve faster injections, at the expense of a higher fuel cost.

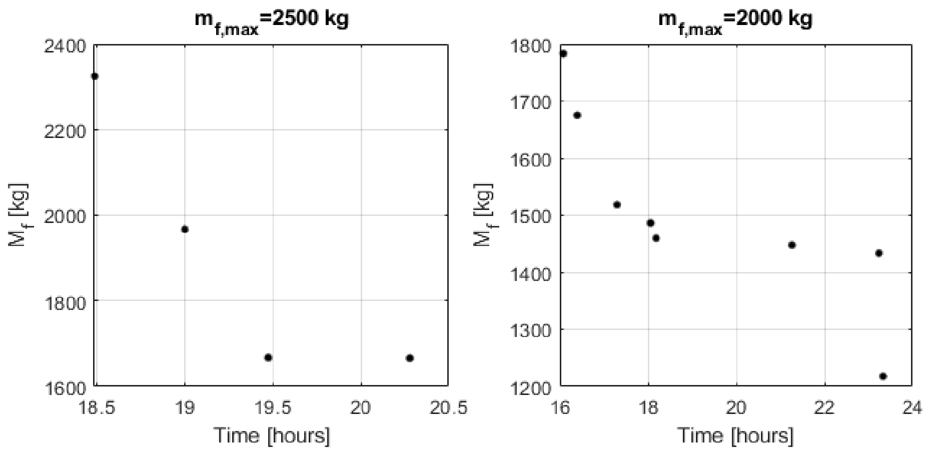

Fig. 5. Solutions for the problem for maximum fuel masses of 2500 kg and 2000 kg

In addition, it is noted that the effect of the orbit eccentricity on the decision of the sequence is almost non-existent, and the main factors are both the semi-major axis and the inclination of the orbits. In fact, as noted previously in [36], the optimiser looks for trends based on the first orbit and the location of the disposal orbit. As such, it generates a balanced inclination sequence, completing all the injections at inclinations below the one of the disposal orbit, then those above and finally it manoeuvres itself towards the final one (or the opposite motion). Regarding altitudes, it either proposes lowering sequentially the orbital altitudes until reaching the lowest one (the disposal orbit), or a growing change from the constellation satellites' operational orbit, through the rest, until the disposal orbit.

However, it can be seen how the optimiser is able to provide with several trajectory options, each one with its own geometrical sequence logic and cost, which allows the user to decide depending on their own interests.

4.2 Analysis of random operations

Once the main problem solution has been studied, it is interesting to understand how the different random operations within the program can affect the final solution. To do so, it was run 10 times for both the 2500 kg and the 2000 kg mass fuels (for a total of 20 full solving processes) changing at each time the seed of the random number generator. The results for the 2500 kg fuel case are shown in Fig. 6; while the results for the 2000 kg fuel case are shown in Fig. 7. It must be noted that the final Pareto Front of the 10 runs is highlighted with respect to the other solutions.

In the case of the 2500 kg fuel mass scenario, results of similar magnitude to the base solution are reached, with most of the solutions demanding fuel masses between 1500 and 2000 kg, and times close to a full day. The best solutions after the 10 runs, however, show how it can reach trajectories requiring less than a day, but beyond the 1500 kg of fuel mass required. An interesting

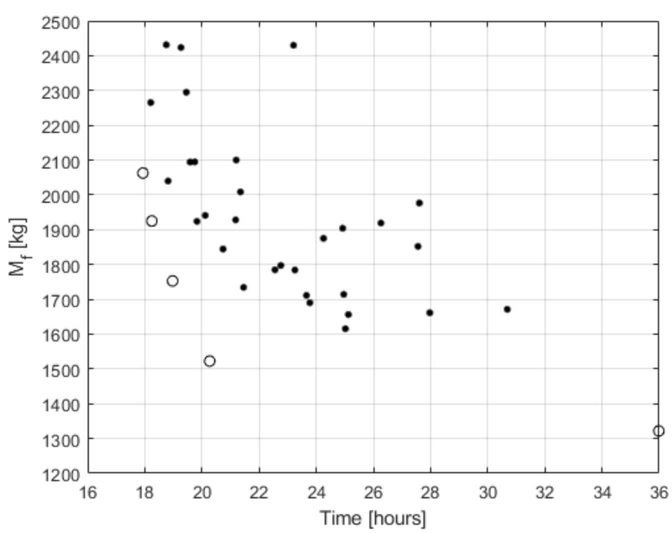

Fig. 6. Results of the problem for 10 runs with 2500 kg of fuel.

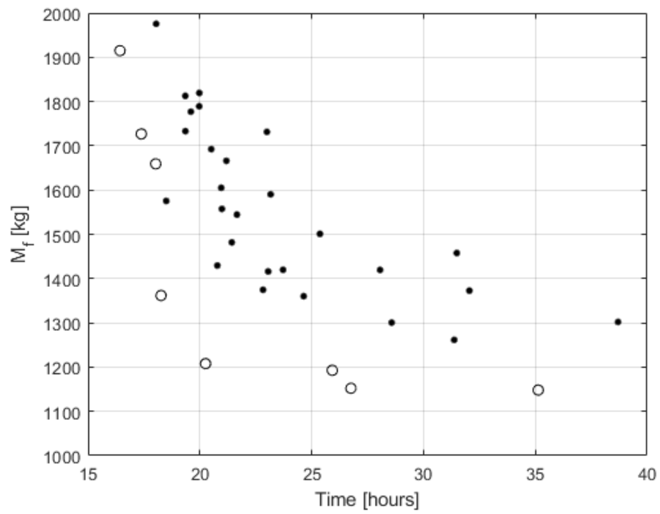

Fig. 7. Results of the problem for 10 runs with 2000 kg of fuel.

sequence is found which needs of a day and a half but is able to achieve all target orbits and the disposal one within 1300 kg, saving 200 kg of fuel both for the activity itself but also for the launcher. Longer times and cheaper mass consumption is related to the delivery of the constellation satellites first through phasing manoeuvres, to then follow into the injection of the nanosatellites, whereas faster deliveries might not necessarily include the full set of constellation payloads first or in sequence.

A similar analysis can be done from the results of the case in which 2000 kg of fuel are available, although the general trend is to (obviously) require less mass fuel, generally between 1300 kg and 1800 kg, and slightly higher times. These solutions, similar to the previous scenario, deliver first all the constellation satellites through phasing, and afterwards moves towards the delivery of the nanosatellites. The lower fuel mass consumption with lower (or similar times) is attributed to the excess weight given with higher usable fuel mass (500 kg) that must be dragged at all times during subsequent manoeuvres as long as it is not consumed. Therefore, it is necessary to investigate what would be the break-even mass fuel point in which just the necessary fuel is mounted on board.

For both cases, however, not all the solutions from all the runs lie within the overall Pareto Front. This is a normal behaviour but highlights the high effect of randomness within heuristic algorithms, which dominate the behaviour of the solver. In fact, while all solutions converged to similar results, either more generations of ants are required to achieve trajectories lying closer to the non-dominated solutions, or more runs are to be performed to get an overall Pareto Front (as done in this case). Nevertheless, it proves how heuristic algorithms provide with a solution which, while being time-efficient might not necessarily provide the global optimum.

4.3 Analysis of ant cost decision

As stated in Section 3.3, each ant can only carry a bi-objective cost from the internal MOMPSO, and different criteria can be used to select which one of the non-dominated solutions is to be attributed to a certain ant. Four different possibilities are envisioned: 1) picking the lower fuel consumption solution; 2) picking the lower TOF solution; 3) picking a random solution; and 4) picking the solution that maximises the weight function of Equation (15). The problem is solved again four times, each time with a different criterion to understand the effect of this decision on the final outputted solution. The results are plotted in Fig. 8. In this case, a maximum of 2500 kg is considered as maximum fuel mass.

Similar to the previous study, it is observed how not all the solutions lie within the Pareto Front. This behaviour was expected based on the previous results. However, more interestingly, these results display a certain trend based on their own nature. Within the overall distribution, it is seen how the minimum time decision criterion provides solutions which lie within the left-most side of the plot giving faster but more fuel-consuming solutions (except for one case), whereas the minimum fuel criterion generates solutions within the more right-hand side, where longer times but cheaper solutions are found. Of course, this was something to be expected, as it follows the own nature of the criteria. On the other hand, using the weighted function shows a nice distribution across both sides, giving solutions which are, in comparison, closer to the origin of the axes, a desired condition of bi-objective optimality. Finally, as expected, by picking a random cost, less evenly distributed results are obtained, as well as less points in the plot, as such criterion does not contribute to converge towards a certain part of the Pareto Front, requiring of more computation to achieve a shape of non-dominated

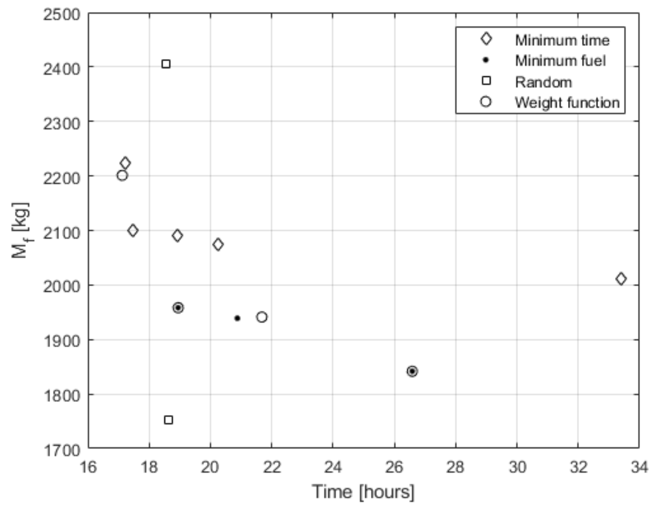

Fig. 8. Results of the problem under different ant cost selection criteria.

solutions distribution. However, the benefit of using this random decision can be observed as one of the outputted solutions is such that the fuel consumption is the lowest, within a time similar to the fastest TOF.

Therefore, it can be stated that the criterion to be used depends on the requirements and the desired outputs, whether it is a lower fuel interest or lower TOF interest, or a balanced one among both.

4.4 Effect of starting at a certain orbit

As a final analysis, the effect of restricting the algorithm to start at a certain node is to be studied. This would be the case of, for instance, a certain payload being high priority with respect to the other ones in terms of timely injection for quick operational status. For this study, a maximum of 2500 kg of fuel mass is assumed, and three different target orbits are considered: orbit 3, orbit 5 and orbit 7. The resulting costs are shown in Fig. 9.

As a first note, it is observed how the results tend to a certain Pareto Front, without necessarily lying on it, as it would be expected. It is interesting however, that even if a certain orbit is specified, the overall results tend to be similar both in terms of time and consumed fuel. For the case of the mission scenario in which the upper stage is forced to start at orbit 3, it is seen how in all provided orders, the constellation satellites are delivered first through phasing, and then all the remaining nanosatellites are injected. These provide with overall lower fuel consumption than for the other two cases, while the lower TOF results show poorer behaviour with respect to them. For the other two cases, all results show how after delivering the first required satellite, they immediately move towards orbit 3 to perform the phasing, through

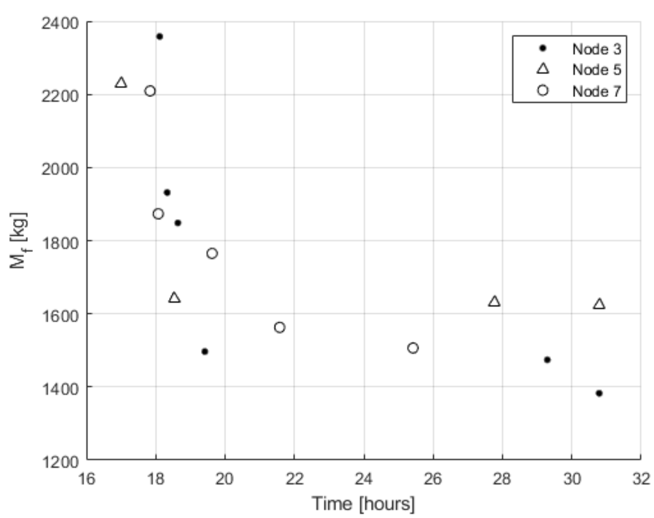

Fig. 9. Results of the problem when forced to start at certain nodes.

orbits 2 and 1, and then the vehicle moves towards the remaining target orbits. This shows how, in fact, even if a certain mandatory initial orbit is given, the optimiser finds delivering the heaviest and same-orbit satellites as soon as possible to minimise the fuel consumption. Therefore, it can be stated that it efficiently accommodates for additional routing requirements depending on the necessities of possible customers.

As a final note, it is observed how the closeness to the supposed Pareto Front across all case scenarios is higher than in the previous analyses. This is in fact attributed to the fact that, by fixing the first orbit, the optimisation problem magnitude has been reduced from 10! to 9! (a complete order of magnitude) and thus convergence is expected to be much faster in terms of iterations, and the effect of randomness is then slightly reduced.

5. Conclusions

This work has proposed an optimisation algorithm that can decide both the sequence and the specific transfers for the multi-rendezvous problem of multiple payload injection within a single launch, minimising the fuel consumption and the total mission time. This is achieved by separating the integer combinatorial problem (solved with a P-ACO strategy) and the continuous manoeuvring problem (solved with MOPSO) and nesting them as to generate a link between both. A strategy that optimises the full set of transfers at once, instead of the typical block building method, is implemented as to achieve a better global optimum.

The algorithm was studied for a case mixing constellation satellites deployment and nanosatellites to study both the flexibility and the performance of the algorithm under realistic industrial case scenarios. It was shown that in all cases, the tool converges towards feasible solutions proposing several possible sequences with the respective impulsive manoeuvres. The results were also influenced by the maximum allowable fuel mass, which showed how having access to more propellant does not necessarily translate to faster manoeuvres, as the excess mass affects the consumption for a given transfer. In addition, due to the combinatorial nature of the sequence problem and the effect of randomness in heuristics, it is suggested to use the tool several times or with more iterations to achieve a better overall set of non-dominated solutions.

All the solutions, however, were shown to be affected by different characteristics of the case scenario to be solved. In fact, depending on the desires of the user in terms of solutions of interest, the attribution of a certain cost to an ant linking both the MOPSO and the P-ACO algorithms, can be selected among the different options. In addition, if any payload is to be delivered first due to mission requirements, it can be accommodated, and will solve the remaining visitation trajectory successfully.

The results allow to conclude that the tool can correctly achieve solutions for the multi-rendezvous problem within the required time and fuel limits, in a timely manner. In addition, the different case scenario characteristics that could impact the performance and results of the algorithm can be easily accommodated, proving the flexibility of the proposed strategy, and its usefulness towards real industrial case scenarios. Finally, the tool provides with feasible solutions that can be realistically considered towards the multi-orbit multi-injection activity of an upper stage, bringing closer this innovative and more efficient way of satellite injection in the quickly growing space access sector.

Acknowledgements

The work presented here has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement N◦860956.

References

[1] Euroconsult, Satellites to be built and launched by 2028. A complete analysis and forecast of satellite manufacturing and launch services, Tech. report (2019).

[2] Moog Inc. Small Launch Orbital Maneuvering Vehicle. https://www.moog.com/markets/space/omv/slomv.html (accessed 16.02.22).

[3] Spaceflight Inc. Sherpa Program: New Orbital Transfer Vehicles Launch Smallsats to Custom Orbital Destinations. https://spaceflight.com/sherpa/ (accessed 16.02.22).

[4] D. Izzo et al., Evolving Solutions to TSP Variants for Active Space Debris Removal, **2015 Annual Conference on Genetic and Evolutionary Computation**, **New York, USA, 2015**.

[5] Y. Chen et al., Optimal mission planning of active space debris removal based on genetic algorithm, IOP Conference Series: Materials Science and Engineering 715 (2020).

[6] K. Daneshjou, A.A. Mohammadi-Dehabadi and M. Bakhtiari, Mission planning for on-orbit servicing through multiple servicing satellites: A new approach, Adv. Space Res. 60.6 (2017) 1148–1162.

[7] B. W. Barbee et al., Design of spacecraft missions to remove multiple orbital debris objects, **2011 Aerospace Conference, 2011**.

[8] B. A. Conway, C. M. Chilan and B. J. Wall, Evolutionary principles applied to mission planning problems, Celest. Mech. Dyn. Astron. 97 (2007) 73–86.

[9] M. Cerf., Multiple Space Debris Collecting Mission—Debris Selection and Trajectory Optimization, J Optim. Theory Appl. 156.3 (2013), 761–796.

[10] N. Bérend and X. Olive, Bi-objective optimization of a multiple-target active debris removal mission, Acta Astronaut. 122 (2016), 324–33.
[11] D. Madakat, J. Morio, D. Vanderpooten, A biobjective branch and bound procedure for planning spatial missions, Aerospace Science and Technology 73 (2018), 269–277.
[12] J. Stuart, K. Howell and R. Wilson, Application of multi-agent coordination methods to the design of space debris mitigation tours, Adv. Space Res. 57.8 (2016), 1680–1697.
[13] T. Zhang et al., Ant Colony Optimization based design of multiple-target active debris removal mission, **2018 SpaceOps Conference, 2018.**
[14] H. Li and H. Baoyin, Optimization of Multiple Debris Removal Missions Using an Evolving Elitist Club Algorithm, IEEE Transactions on Aerospace and Electronic Systems 56.1 (2020), 773–784.
[15] L. Federici, A. Zavoli and G. Colasurdo, Evolutionary Optimization of Multirendezvous Impulsive Trajectories, Int. J. Aerosp. Eng. 2021 (2021), 1–19.
[16] L. Federici, A. Zavoli and G. Colasurdo, A Time-Dependent TSP Formulation for the Design of an Active Debris Removal Mission using Simulated Annealing, ArXiv abs/1909.10427 (2019).
[17] M. Cerf, Multiple Space Debris Collecting Mission—Debris Selection and Trajectory Optimization, J Optim. Theory Appl. 156.03 (2013), 761–796.
[18] J. M. Bourjolly, O. Gurtuna and A. Lyngvi, On-orbit servicing: A time-dependent, moving-target traveling salesman problem, International Transactions in Operational Research 13 (2006), 461–481.
[19] J. Bang and J. Ahn, Two-phase framework for near-optimal multi-target Lambert rendezvous, Adv. Space Res. 61.5 (2018), 1273–1285.
[20] J. Yu, X. Chen and L. Chen, Optimal planning of LEO active debris removal based on hybrid optimal control theory, Adv. Space Res. 55 (2015), 2628–2640.
[21] A.P. Alpatov and Y. M. Holdsht, On the choice of the ballistic parameters of an on-orbit service spacecraft, Technical Mechanics (2019).
[22] R. H. Battin, An introduction to the mathematics and methods of astrodynamics, AIAA, 1999.
[23] R. H. Gooding, A procedure for the solution of Lambert's orbital boundary-value problem, Celest. Mech. Dyn. Astron.48 (1990), 145–165.
[24] E. R. Lancaster and R. C. Blanchard, A unified form of Lambert's theorem, Tech. rep. NASA TN D-5368 (1969).
[25] D. Izzo, Revisiting Lambert's Problem, Celest. Mech. Dyn. Astron.(2014).
[26] R. Oldenhuis. Robust solver for Lambert's orbital-BVP. https://github.com/rodyo/FEX-Lambert/releases/tag/v1.4, Github (accessed 20.01. 2022).
[27] H. D. Curtis, Orbital Mechanics for Engineering Students, third ed., Butterworth-Heinemann, Boston, 2014.
[28] M. Guntsch and M. Middendorf, A Population Based Approach for ACO, in: Applications of Evolutionary Computing, Springer Berlin Heidelberg, Berlin, 2002, pp. 72–81.
[29] M. Guntsch and M. Middendorf, Solving Multi-criteria Optimization Problems with Population-Based ACO, **Evolutionary Multi-Criterion Optimization, Second International Conference (EMO'03), 2003**.
[30] L. F. Simões et al. 'Multi-rendezvous Spacecraft Trajectory Optimization with Beam P-ACO, Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science, Springer International Publishing, 2017, pp. 141–156.
[31] C.A.C. Coello, G.T. Pulido and M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation 8.3 (2004), 256–279.
[32] V. Martínez-Cagigal, Multi-Objective Particle Swarm Optimization (MOPSO). https://www.mathworks.com/matlabcentral/fileexchange/62074-multi-objective-particle-swarm-optimization-mopso MATLAB Central (accessed 25.01.2022).
[33] K.F. Wakker, Fundamentals of Astrodynamics, TU Delft Library, 2015.
[34] A. Alcantarilla Romera, Fourth shell of Starlink constellation reaches 1,000 satellites. 17 July 2022, https://www.nasaspaceflight.com/2022/07/starlink-4-22-launch/ (accessed 21.08.2022).
[35] S. Gallucci, R. Mancini, E. Scardecchia, and F. Spaziani, The AVUM orbital module for the space rider system, **European Conference for Aeronautics and Space Science (EUCASS), Madrid, Spain, 2019.**
[36] I. Alforja Ruiz, M. Lavagna, Design of the Optimal Trajectory for Multi-Payload Multi-Orbit Injection of an Upper Stage, **European Conference for Aeronautics and Space Science (EUCASS), Lille, France, 2022.**