DOI: 10.13009/EUCASS2022-7312
9TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)

Design of the Optimal Trajectory for Multi-Payload Multi-orbit Injection for an Upper Stage

Iñigo Alforja Ruiz⋆† *and Michèle Lavagna*⋆
⋆*Politecnico di Milano, Aerospace Science and Technology Department*
Via La Masa 34, 20156, Milano, Italy
inigo.alforja@polimi.it · michelle.lavagna@polimi.it
†Corresponding author

Abstract

This paper introduces an optimization tool that provides with the preliminary trajectory to deliver multiple payloads into differentiated orbits to keep up with the increasing number of programmed launches. This instrument is a two-level optimization algorithm which solves the sequential problem via a Population-Based Ant Colony Optimization strategy, and the transfer problem through a Multi-Objective Particle Swarm Optimization algorithm. The goal is to achieve a set of trajectories that minimize both the fuel consumption and the total mission time. The algorithm is tested with two different case scenarios showing its performance and flexibility to accommodate different kinds of injection missions.

1. Introduction

More and more, human lives depend on spaceborne systems which provide a means for communication and access to information, in addition to a source of science opportunities. This has been translated into an increasing number of satellites orbiting our Earth; a fact observable also in the increase in the volume of yearly launches, beyond the current capabilities of space transport systems. Indeed, it is predicted that the growth in the number of planned small satellite launches this decade will increase at a rate four times that of the previous one [1]. The growing interest in constellation missions, such as StarLink or OneWeb [2], as well as the higher accessibility of space for smaller users, is heading the space infrastructure into an unsustainable situation both in logistic and ecological terms. This situation demands of innovative and efficient ways to deliver spacecraft into their operating orbits.

Current strategies of delivery for multiple satellites involve piggybacking secondary payloads to a primary delivery, limiting the flexibility in terms of scheduling and orbit injection. This practice might discourage new projects, especially those from the private sector, and prevent the expansion of this developing market. A new strategy to access space is necessary, focusing in the efficient injection of multiple payloads into multiple dedicated orbits. This concept has been already tested by the use of vehicles or *tugs* which are launched and then used as intermediaries to deliver smaller satellites, staring from the orbit of a primary payload. Some examples of this type of mission are the Small Launch Orbital Maneuvering Vehicle [3], or the Sherpa-NG project of Spaceflight Inc. [4]. Therefore, it has been shown that a multiple-payload multiple-orbit delivery mission is feasible, and it could revolutionize the way in which launches are scheduled and performed.

The design of the trajectory for the multi-payload multi-injection mission requires the definition of the visitation sequence, but also the individual transfers in-between consecutive orbits. This multi-rendezvous problem has been studied before, mainly for activities related to Active Debris Removal (ADR) and On-Orbit Servicing (OOS), providing with different solutions. In fact, the combinatorial nature of the orbit visitation has been previously associated to the Traveling Salesman Problem (TSP) [5], which has led to the proposal of extensive search strategies such as in Chen et al. [6]. Daneshjou et al. [7] follow a similar approach in their work after a preliminary selection based on feasibility comparisons. However, although being able to provide with a global optimum, computing all combinatorial possibilities demands of high computational efforts, becoming unfeasible after a certain number of orbits. Alternatively, other tree-search algorithms have been studied in which different sequences are extended or cut according to different criteria. For instance, Barbee et al. [8] use a Series Method approach to construct the sequence based on the minimum necessary ΔV between two consecutive orbits, similar to a Greedy Algorithm. Nevertheless, most analyses using these strategies implement versions of the Branch-and-Bound method [9, 10, 11, 12]. However, similar to the extensive search approach, with medium to high number of orbits, the quality of the solution decreases considerably, and the

DOI: 10.13009/EUCASS2022-7312

computational times can become prohibitively expensive. As a counterpart, heuristic algorithms which provide a sub-optimal solution at remarkably lower computational times have been proposed. Among these, one of the most often used is the Ant Colony Optimization (ACO) due to its natural resemblance to the tree-search structure [13, 14]. A hybrid of an ACO with added elitism is proposed in Li et al. [15] to improve accuracy and diversity of the solution. Evolutionary algorithms, such as the Genetic Algorithm (GA), have also been used, which allow to encase the different orbit sequences as chromosomes, allowing for simple integer programming [9, 16]. Simulated annealing can also be found in some works, although they do not seem to provide any additional advantage with respect to the other algorithms [17, 18].

Nevertheless, the multi-rendezvous problem requires that the individual transfers, given a certain sequence, are also estimated. Typically, the cost of these transfers is defined by means of a pre-computed matrix which the combinatorial solver can access. This matrix is often simplified to be time-independent [19, 11] in a way that Hohmann transfers are assumed for all of the displacements between two orbits. However, the true time-dependency of the transfers can be accounted for by establishing a time grid and calculating all the possible costs between two orbits and two points in time, generally by means of impulsive Lambert targeting manoeuvres [10, 18, 12]. This process speeds up the optimization process, but requires of large data storage. A more efficient approach is found in Bang et al. [20], for which all pre-computed costs are compared looking for local minima in each transfer between two orbits, reducing the search space size. In none of these approaches the transfers are considered for optimization, usually providing with solutions of less quality in terms of minimization of objective values, such as the total ΔV. Nevertheless, some studies have already shown that by introducing them into the optimization, better solutions can be achieved. This optimization is usually done by means of heuristic algorithms such as GAs [9] or evolutionary algorithms [17, 16]. However, algorithms that can exploit the continuous nature of the search space are preferred, in particular the Particle Swarm Optimization strategy (PSO). This algorithm is found to be used in papers [21, 7, 6], providing with accurate results. Moreover, it has been noted that all these optimization strategies rely on individually optimizing the subsequent transfers to build-up the sequence total cost, instead of computing the complete trajectory optimization. This could lead to solutions of lower quality, as all transfers are subjected to the decisions and values of previous manoeuvres, limiting the search space.

The current paper focuses on the solution of a time and fuel mass constrained multi-payload multi-orbit injection mission, assuming impulsive transfers, whose aim is to visit all orbits and minimize both the consumed fuel and the time of flight. For this purpose, an optimization tool has been created structured in two levels: an outer level in which a Population-based ACO is used to solve the sequential part of the problem; and an inner level which uses a Multi-Objective PSO to solve the complete set of transfers. The mission description, as well as the mathematical formulation of the problem, are described in Section 2. Then, Section 3 shows the approach used in developing the optimization tool, as well as its structure. Finally, two specific case scenarios are solved in Section 4 to show the performance of the developed algorithm, followed by the main conclusions in Section 5.

2. Problem Statement

2.1 Mission Definition

The multi-payload multi-orbit delivery capability proposes an interesting problem scenario of multi-rendezvous in which both the sequence of visitation and the individual transfer legs need to be computed and decided upon. Such a mathematical problem is, in fact, not straightforward to solve due to its complexity.

Consider an upper stage, or "vehicle", which is used to deliver a set of N satellites, or payloads, into N distinct positions in space, whether these are differentiated orbits or specific locations within a same orbit, or a mixture of both. These orbits or positions within the orbit, as well as the masses of the payloads, are specified before-hand based on the necessities and requirements of the satellites to be delivered within the same launching. In the present study it is assumed that the trajectory starts at a specific initial orbit, which is considered to be one of the orbits of the payloads to be injected, after payload delivery. However, the cost of launching into that specific orbit is not included in the optimization of the trajectory. The decision of which orbit to start from is to be decided by the optimization process itself, assuming that the cost to reach any of these from the launching site will be relatively similar. In addition, the vehicle is to finish its trajectory at a pre-defined disposal orbit that complies with the mitigation guidelines on space debris. As the optimization of such an orbit itself (related to the parameters ensuring a certain orbital decay, security, etc.) is not within the scope of this work, a certain orbit will be arbitrarily picked that fulfils the requirements while being "close" to the target orbits.

In this way, the full mission can be summarized as follows: the vehicle deploys the first satellite and moves to the next orbit, in which it injects the following payload. It then waits in this second orbit until the next manoeuvre towards the next orbit is to be performed, ensuring that this time is at least the necessary one to perform the injection correctly.

DOI: 10.13009/EUCASS2022-7312

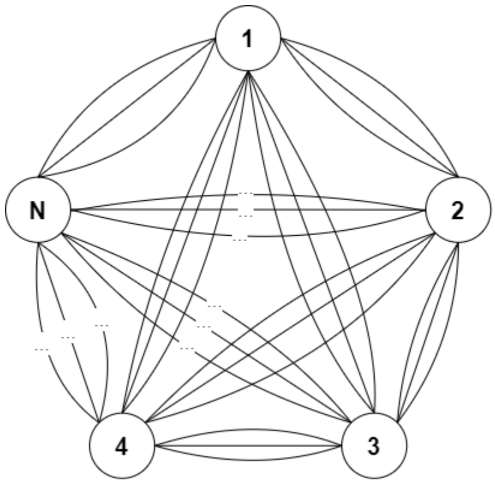

Figure 1: Schematic of the time-dependent TSP

This process is repeated until all satellites have been deployed, after which the vehicle manoeuvres itself towards the predefined disposal orbit, at which point the mission is considered to be finished. It must be noted that, in general, since the target is a specific orbit and the exact location within this orbit is not crucial (unless when dealing with constellations), the vehicle will consider all points within the orbit to be equally interesting for the transfer manoeuvre.

When dealing with a launching problem, and from the point of view of the end-users or satellite owners, the deployments are desired to be as fast and as cheap as possible, within a certain maximum value for both. It must be ensured, then, that the complete mission is fulfilled both within the specified maximum time and within the fuel mass budget given to the vehicle, while still looking for the minimal value for both. In addition, since generally upper stages are provided with high thrust propulsion systems, all firings of the main engine are approximated as impulsive manoeuvres.

2.2 Mathematical Formulation

The generation of an optimal sequence of visitation falls into the category of problems related to the Traveling Salesman Problem (TSP) [5]. This one is defined by a certain salesman or vehicle that needs to visit a set of cities (or nodes) only once, while minimizing the total cost of visitation (whether this is distance, time, fuel, etc.). Thus, the problem under consideration can be described as the graph problem $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, in which $\mathcal{V} = \{1, ..., N\}$ is the set of vertices (or orbits), and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$ is the set of arcs connecting those vertices, which in this case are the orbital transfers. Nevertheless, the TSP in space has significant differences with respect to the standard problem, among which the two most remarkable are: 1) the problem is time-dependent in a way that the instant in which the manoeuvre starts and ends will affect the cost associated to the arc, as well as being dominated by nonlinear dynamics; and 2) the problem is open-route, such that the start and end points differ.

Based on these concepts, one can define the set of satellites to be delivered as $\mathcal{S} = \{1, ..., N\}$, being the complete set of vertices or nodes to be visited $\mathcal{V} = \{0, 1, ..., N, N + 1\}$, where $i = 0$ has been included as the possibility of the vehicle being at a given initial position different than any target orbit (like a ground station or a parking orbit) and $i = N + 1$ represents the final disposal orbit. In addition, as stated before, the transfers is highly influenced by time, not only involved in the dynamics of motion but also in the total duration of the mission, as a certain transfer in terms of time will affect subsequent possible arc legs. Therefore, it is important to correctly compute the optimal transfers in between orbits given a certain sequence to ensure that the correct associated costs are being evaluated. these transfers are such that both the start and final position of the manoeuvre, as well as the time-of-flight (TOF), are variables to be decided for each one, as faster motions will imply higher fuel consumption, generally, and a balance will need to be found. The coupling of both problems becomes apparent in this way and highlights the complexity of the mathematical problem. A schematic of this type of sequential problem is shown in Figure 1 in which the possible connections are outlined, with the large range of possibilities for each one.

The discussed optimization problem can be then defined as the minimization of:

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

DOI: 10.13009/EUCASS2022-7312

with $m_{f,i}$ being the fuel mass consumed to reach orbit i, and t_{tot} the total mission time. In addition, each transfer is dominated by the following dynamic equations of motion [22] :

$$\ddot{\boldsymbol{r}} = \frac{\mu}{r^3}\boldsymbol{r} + \frac{T}{M}\boldsymbol{e_T}\delta + \boldsymbol{f}_{dist} \tag{2}$$

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

where $\boldsymbol{r}$ is the radial position of the vehicle, μ is the gravitational parameter of the central body, T is the thrust magnitude, $\boldsymbol{e_T}$ is the thrust vector, M is the total mass, δ the relay on-off function of the engine (switch), I_{sp} the specific impulse of the engine, and g_0 the gravitational acceleration at the surface of Earth. In addition $\boldsymbol{f}_{dist}$ accounts to the acceleration due to disturbance effects. For the purposes of this work, the transfers are defined to be specific manoeuvres which allow for some simplification in the calculation of these transfers, in a way that the firing of the thruster can be approximated by impulsive manoeuvres and the necessary ΔV is obtained through specific formulas. In particular, three types of motion are envisaged: 1) a general Lambert targeting manoeuvre; 2) phasing with respect to a certain orbit; and 3) exploiting the *J2* effect of the spherical harmonics model of the Earth's gravitational field. As an additional simplification, no perturbances are considered except for the *J2*-effect for this last type of manoeuvre.

The presented problem is subject to the constraints:

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

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

Equations (4), which are related to the sequence generation part of the problem, ensure that each orbit is only visited once, and that all orbits are in fact visited. On the other hand, Equations (5) ensure that the sum of the consumed fuel and time of all the k transfers stay within the limited maximum amount for each.

The mathematical formulation of the problem enables a quick identification of two clearly differentiated parts. On the one hand, on finds the combinatorial part of the problem related to the visiting sequence selection in a way that all orbits are visited but just once (Equations (4)). This objective of finding the optimal sequence falls into the category of integer programming. On the other hand, it is found the optimization of the transfers themselves (Equations (2) and (3)), which falls into the category of nonlinear continuous programming. Since both are tightly coupled, the problem to be dealt with is a mixed integer nonlinear programming (MINLP) problem. Solutions to this problem are not straightforward and need of strategies that can solve simultaneously integer and continuous optimization problems, despite their remarkably different nature.

3. Solution Approach: Bi-level Optimization

As stated previously, the problem falls into the category of MINLP optimization problems, which are of the NP-hard type and thus cannot be solved deterministically without the need of computational times that grow in a factorial number with the number of possible routing cities to be visited. As such, heuristic algorithms have been developed and used to achieve a sufficiently good (but sub-optimal) solution within feasible times. Additionally, it has been proposed to be solved by decomposing the problem into two sub-problems whose solutions can be used to achieve the optimal trajectory.

Following this same line of thought, the study proposes a bi-level optimization solution in which the transfer and the sequential problem are solved individually while keeping the inter-connections related to the problem's nature. In this way, the process is divided into: 1) an internal level optimizing the individual transfers between two consecutive orbits, assuming a certain order; and 2) an outer level dealing with the visitation sequence orbit, not explicitly accounting for the individual transfers. These two are, however, tightly connected as they depend on each other to properly solve the MINLP problem. On the one hand, the inner level requires a certain sequence of visitation forwarded by the outer level to perform the transfers between the fixed order. On the other hand, the estimated cost for the whole trajectory of transfers calculated by the inner level is used by the outer level as a measure of quality for a given sequence within its own optimization process. In this manner, both levels are structured in a nested manner and constantly interact to achieve the solution. A simple schematic of the bi-level optimization structure is shown in Fig 2.

Nevertheless, as stated before, the objectives of optimization are both the fuel mass consumed and the total mission time. These two objectives are optimized in both levels, using the concept of Pareto dominance in a way that

DOI: 10.13009/EUCASS2022-7312

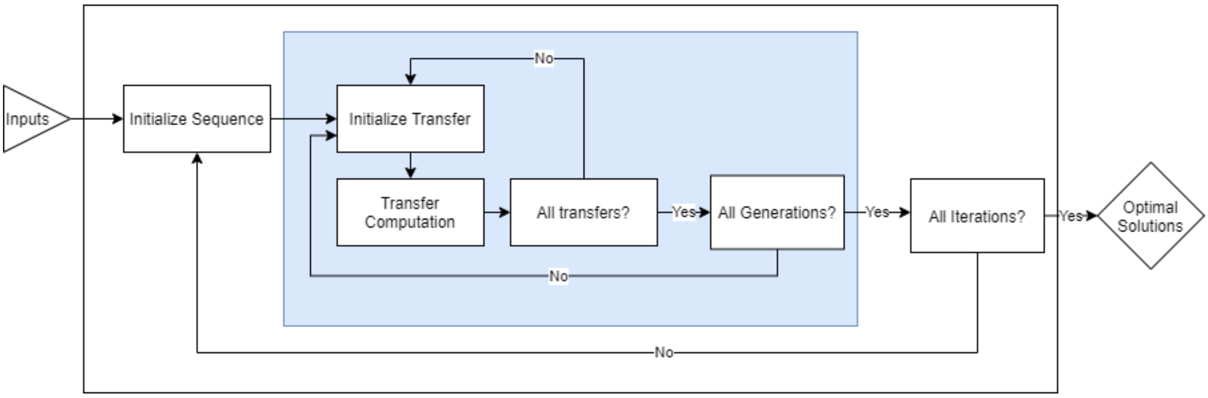

Figure 2: Schematic representation of the bi-level structure for solving the TSP in space

the output is always the set of points belonging to the Pareto front. A certain solution A1 is said to dominate a solution A2 if all of the values of the objective functions of A1 are better performing than the ones of A2. The Pareto front is composed of all non-dominated solutions. However, as the cost for a single sequence cannot be treated to be a complete set of points, but a single pair of values (fuel mass and time of flight), a single solution of the inner level is outputted for the outer level to be used as the quality reference.

3.1 Single transfer cost estimate

In this section, the methodology used to calculate the ΔV, fuel mass and TOF for the individual transfers is explained. Indeed, three types of manoeuvres between orbits are considered: 1) Lambert targeting; 2) phasing; and 3) exploiting the *J2* effect of the Earth's gravitational field to change the right-ascension of the ascending node (RAAN) of the orbit. The decision on which manoeuvre to perform at any point, between two specific orbits, is decided by the comparison of the Kepler elements defining them, as these are used as the main set of parameters defining a certain operational orbit (as expressed in the design of missions). Within this study, the Kepler elements used are: the semi-major axis (a), the eccentricity (e), the inclination (i), the RAAN (Ω), the argument of the periapsis (ω), and the true anomaly (θ).

Lambert targeting

In the most basic case of transfer, it is considered that it is done by means of an impulsive Lambert targeting strategy. The principles of this boundary-value problem can be found in Ref. [23], but can be summarized as follows: given a certain particle orbiting a central body, in a 2-body problem configuration, and given the initial and final positions of the particle as well as the time-of-flight in between those two points, the objective is to find the needed velocity at the start and end of the trajectory in a way that all conditions are met. Therefore, it is presented as one of the classic boundary-value problems in space, and its solution has been extensively studied since its proposal, leading to several different programmatic implementations that accurately solve it. Among these, one of the most famous ones is that proposed by Gooding [24] which includes some changes to the one developed by Lancaster and Blanchard [25]. This solution provides with a robust method to solve for the initial and final ΔVs to perform the desired manoeuvre. However, a more efficient solution was proposed by D. Izzo [26] by which equally accurate solutions could be found in much shorter times, at the cost of a slightly less robust algorithm. In order to deal with this issue, the used algorithm tries to solve the Lambert problem by means of Izzo's formulation and, if it is unable to converge towards a solution, it calculates it using Gooding's algorithm as a backup. In this manner, it is ensured that all Lambert computations reach a solution, and the slower algorithm is only called when its higher robustness is required. This implementation has been taken from the work of R. Oldenhuis [27].

Phasing manoeuvre

In the case of multiple injection within the same orbit (e.g. constellation missions), a phasing manoeuvre is considered. It is performed by applying two consecutive impulsive firings: a first one to move into the phasing orbit within which the spacecraft will wait until correct phasing is achieved; and a second one to re-insert itself in the original orbit once the desired phase difference is reached. The first manoeuvre is performed at the perigee or apogee of the original orbit to reach a phasing orbit with smaller or higher orbital period depending on whether the desired position is ahead or behind, respectively, of the current position of the vehicle [28]. In the case of circular orbits, the initial position is

DOI: 10.13009/EUCASS2022-7312

an optimization variable, while in elliptical orbits the possible initial position is constrained to be either the apogee or perigee to keep it as close to a Hohmann transfer as possible. The period of this intermediate of phasing orbit is calculated as:

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

where T_1 is the orbital period of the original orbit, t_{AB} the time to close the gap between both positions, and N_{rev} the number of revolutions performed in the phasing orbit. The calculated value allows to compute the semi-major axis using:

$$a_{\text{phasing}} = \sqrt[3]{\mu_E \left(T_{\text{phasing}}/2\pi\right)^2} \tag{7}$$

With this value, the rest of the orbital parameters can be obtained, allowing to calculate the necessary ΔV and the total manoeuvreing time based on the number of revolutions needed.

RAAN drift manoeuvre

Finally, when one of the parameters to be changed is the RAAN, one can exploit the *J2* effect of the Earth's spherical harmonics gravitational model to avoid the high cost in terms of ΔV associated with such a manoeuvre if the engine were to be used [29]. Instead, by considering the secular drifting effect on the RAAN of the *J2* zonal harmonic, the parameter can be changed without the need of thrusting force, saving fuel mass, but at the expense of much higher mission times. This rate of change in RAAN is given by:

$$\dot{\Omega} = -\frac{3nR_E^2 J_2 \cos i}{2a^2(1-e^2)^2} \tag{8}$$

where n is the mean angular velocity of the orbit and R_E is the radius of Earth. To reduce the drifting time, the spacecraft can move towards an intermediate orbit where it can achieve the necessary RAAN change faster, at the cost of using fuel mass to move to such orbital geometry. The selection of this orbit must be also considered in the optimization problem in terms of semi-major axis and inclination. This drifting orbit is considered circular and with an argument of perigee being half of the sum of those of the original and the final orbits, for simplification purposes. The total drifting for a spacecraft in a drifting orbit (subscript *d*) to reach the RAAN of a final orbit (with subscript *2*) is then obtained as:

$$t_{\text{drift}} = \frac{\Omega_2 - \Omega_d}{\dot{\Omega}_d - \dot{\Omega}_2} \tag{9}$$

It must be considered that the motion towards this intermediate orbit, and the transfer to the final desired orbit are modeled also as Lambert targeting problems in which the initial and final position, as well as the TOF, are design variables.

3.2 Outer Loop

As stated previously, a bi-level approach for the design of the optimization strategy is proposed, with an outer level dealing with the combinatorial problem and an inner level solving the transfer problem. To solve the visiting sequence problem, it was decided to use an heuristic that could exploit the tree-shaped structure which comes naturally with the combinatorial essence. In this sense, ACO-related algorithms were investigated, leading to the decision to use a Population-based ACO (P-ACO), as introduced in Ref. [30], and expanded towards multi-objective problems in Ref. [31]. In addition, its usage in the work of L. Simões et al. [32] for a problem of similar nature (mainly the multiple asteroid visiting of GTOC5 [33]) provides with a proof on its usefulness for the multi-rendezvous problem.

This type of algorithm follows the typical concept of ACO strategies that ants traveling a certain path (or sequence) leave a trail of pheromone concentration, denoted by τ. The higher the amount of concentration of pheromone in a certain sequence, the more likely an ant will follow it. In addition, a problem-specific heuristic measure, denoted by η, related to the problem is included as to give the ants an estimated measure of the possible cost from a certain node to subsequent nodes. In the problem under consideration, this heuristic was estimated to be the remaining ΔV fraction within the budget after performing the manoeuvres to individually change all the orbital parameters from one orbit to another, using approximate analytic expressions [29]. Then, the probability for an ant at node i to move towards node j, with j being part of the possible set of subsequent nodes S, is given by:

$$p(i,j) = \frac{\tau(i,j)^{\alpha}\eta(i,j)^{\beta}}{\sum_{z\in S} \tau(i,z)^{\alpha}\eta(i,z)^{\beta}} \tag{10}$$

DOI: 10.13009/EUCASS2022-7312

in which α and β are design factors which determine the relative contribution of both measures of interest.

Algorithm 1 P-ACO [32]

```
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
```

Algorithm 2 MOPSO algorithm [34]

```
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
```

The particularity of the P-ACO is that the pheromones are only deposited by those ants belonging to the *Population* set, composed by the best ants of each generation, which allows for a faster convergence [30]. In this case, as a multi-objective cost is being considered, a similar strategy to that proposed in Ref. [32] is used to generate the *Population*. In this way, after each generation, this set, called the *Elite* is composed by all the non-dominated solutions found until then, and then the pheromone matrix is re-calculated based on these possible solutions. However, there is a limit k on the amount of members that can be included in the population, which is implemented in a FIFO-queue fashion on each node independently. This queue is emptied at the end of each generation, to be then re-filled with the new set of non-dominated solutions. The implementation of this algorithm has been based and modified from the open source code developed by L. Simões et al. [32], with the pertinent changes. A pseudo-code is presented in Algorithm 1 as a summary of its principles. It must be noted that a number of internal functions are mentioned in the pseudo-code. These include the *pheromone* function, which provides with the pheromone matrix based on the current trails; the *heuristic* function, which provides the problem-specific heuristic value; the *roulette_wheel* function, which selects the next node based on the Roulette Wheel principle for probability-based selection; the *evaluate* function, which gives the cost associated to a certain tour; the *sort* function, which outputs the non-dominated tours based on Pareto dominance; and the *pheromone_update* function, which generates the new pheromone deposit on trails based on the new *Elite* set.

3.3 Inner Loop

At the moment of evaluating a certain tour cost, the inner loop of the optimization algorithm is called, which calculates (given the node sequence) the optimal impulsive manoeuvres needed. Similarly to the combinatorial problem, an heuristic algorithm was selected. In this case, the Multi-Objective Particle Swarm Optimization (MOPSO) algorithm was used to exploit the continuous domain of the search space. This algorithm was proposed by C. Coello [34] and the main difference with the classic PSO is that instead of a single leader, there exists a repository with all non-dominated

DOI: 10.13009/EUCASS2022-7312

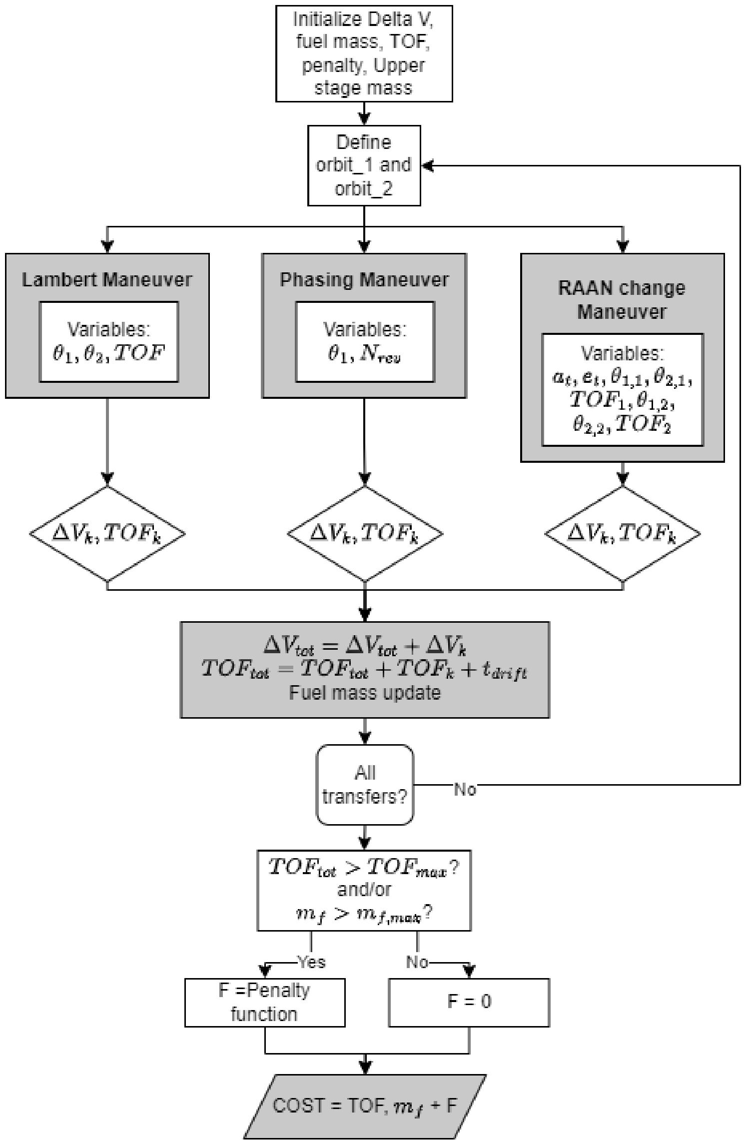

Figure 3: Flow-diagram illustrating the objective function of the MOPSO

solutions found until a certain iteration. In this way, when a particle wants to move, it randomly chooses a member of the repository as its leader. Such an algorithm was also chosen due to its proven usefulness in the multi-rendezvous problem, as shown by Daneshjou et al. [7], and the used implementation was a modification of the open source code developed by V. Martínez-Cagigal [35]. As for the previous algorithm, a pseudo-code is presented in Algorithm 2 as a summary of its principles. It must be noted that the motion equations (for velocity and position) are included, for which the following variables are used: w is the inertia of the particle, $c1$ and $c2$ are the factors providing importance of self-knowledge and leader knowledge; and $r1$ and $r2$ are random numbers between 0 and 1.

However, before calling the MOPSO algorithm, the complete cost function must be constructed and, as several types of manoeuvres are envisaged and depend on each pair of consecutive orbits, it must be generated once the tour sequence is set. In this way, once a certain ant completes the tour and calls the *evaluation* function of the problem, it triggers a process to define the full trajectory cost function before calling the optimization algorithm. This is done by first including in the tour the final disposal orbit (if required) and then checking for each transfer expected which kind of manoeuvre will be performed. In the general case, a Lambert targeting one will be selected with 3 variables for optimization: the initial true anomaly, the final true anomaly and the TOF. If the two orbits are the same, and only phasing must be corrected, a phasing manoeuvre is selected with 2 optimization variables: the location of the manoeuvre (in terms of true anomaly) and the number of revolutions. Finally, if the RAAN between the two orbits is different (and it is explicitly defined that the *J2* effect is to be exploited), a drifting strategy is picked, with 6 variables for optimization: the semi-major axis and inclination of the drifting orbit, the final position and TOF of the first Lambert

DOI: 10.13009/EUCASS2022-7312

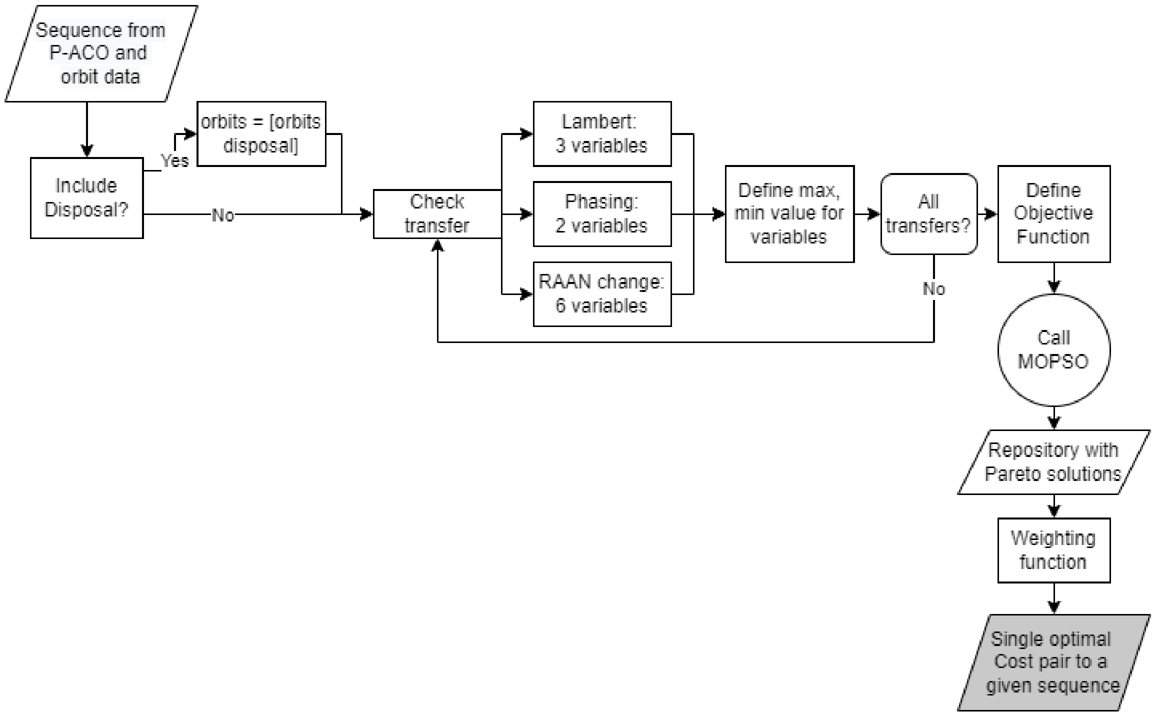

Figure 4: Flow-diagram illustrating the cost calculation for a given tour

towards the drifting orbit, and the initial position and TOF of the second Lambert towards the final orbit. The initial position at the first orbit and the final position at the final orbit are pre-selected to be the perigee and apogee respectively, as to simplify the problem and make it as close as possible to a Hohmann transfer. A maximum and minimum values are given to all these variables as to confine the search space of the MOPSO algorithm. All these variables and types of transfers are stacked in a *decision*-array that will be fed to the objective function.

Once the previous process is finished, the objective function is constructed based on the previously defined sequence of manoeuvres. The generation of this objective function is summarized in Figure 3. For every transfer, the required manoeuvre is calculated according to the expressions shown before, providing with a ΔV and a TOF for each leg. For the case of Lambert targeting, two additional checks are included: 1) the manoeuvre involves an elliptical transfer orbit (checked by ensuring that the TOF is greater than Barker's time [36]); and 2) the transfer arc does not go below a certain threshold that could endanger the upper stage. The time to wait from the arrival to an orbit to its next manoeuvre is included in the TOF, and the fuel mass used in the manoeuvre (accounting for the discrete release of satellites) is updated. After all the trasnfers are computed, a total ΔV, fuel mass and TOF are calculated. At this point, the constraints of Equation (5) are enforced by beans of a penalty function. Assuming an objective f and a maximum allowed value L, the penalty is included in the following fashion:

$$\lambda(f, L) = (\max(0, f - L))^2; \quad f = f + F \times \lambda(f, L) \tag{11}$$

where F is the scalar penalty factor, used to scale all objectives to an comparable quantity. The output of the objective function is then the cost in terms of TOF and fuel mass, with the added penalty.

The previously defined objective function can then be used to call the MOPSO algorithm, whose output is the Pareto front of all the non-dominated solutions for the optimal trajectory given a certain sequence. However, the complete set of solutions cannot be attributed to a single ant route, as a two-valued cost is expected. Therefore, a single point of the front is to be picked and attributed to the given tour in a certain ant and generation. The decision on which cost is associated to the tour is based on a weighted function such that the constraint violation in terms of maximum fuel and maximum TOF are minimized, as introduced by L. Simões et al. [32]. The way in which this function is defined is by first calculating the fraction of fuel mass and time remaining at the end of the tour with respect to the maximum values:

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

which are then used to generate the weighting reference (ranging from 0 to 2):

$$mt_{\mathrm{sum}} = m + t \tag{13}$$

DOI: 10.13009/EUCASS2022-7312

The weights can now be calculated in a way that the more an objective value complies with the requirements, the less importance is given to it. In such a way, closeness to not fulfilling constraints is punished, which also ensures the look of solutions which leave some time and fuel mass margin. The weights are then:

$$wM = 1 - m/mt_{\text{sum}}; \quad wT = 1 - t/mt_{\text{sum}} \tag{14}$$

and finally the weighted function is defined as:

$$A = wM \times m + wT \times T \tag{15}$$

The cost attributed to a certain ant (or tour) is the one maximizing the value of A, thus linking a single bi-valued cost of the MOPSO's Pareto front instead of the complete set. This value is the one being returned to the outer-level part of the optimization structure, and is the one used to be used for the selection of the new generation *elite* population. A schematic of the complete inner loop construction is given in Figure 4.

4. Analysis of a Case Scenarios

In order to demonstrate the validity of the developed optimization tool, two different cases will be solved, as to show the flexibility of the instrument. The first one is a constellation mission, while the second one is a group of independent missions "sharing the ride". In both cases, an upper stage of 1000 kg of dry mass with a maximum fuel mass of 1500 kg is considered. Chemical thrust is the propelling strategy of the main engine, as the satellites are desired to be on orbit and operational as soon as possible. Thus, a specific impulse of 450 seconds is assumed.

Case 1: Constellation mission

As a first case scenario for multi-payload multi-orbit injection, a constellation mission is proposed in which 6 satellites are to be delivered at two different altitudes. Consider, for instance, a hypothetical Earth observation mission which requires of a set of six different micro-satellites (50 kg) divided in two groups of 3 at two altitudes of 600 km and 800 km, equidistant within the orbit (with a phase difference of 120º). The observation conditions require both orbits to be Sun-synchronous (thus both orbits will be at different inclinations) and the spacecraft to be disposed at an orbit of 300 km altitude. The parameters of the orbits are summarized in Table 1, where the disposal orbit is identified with **[D]**. The mission is considered to start after the deployment of the first satellite, assuming that direct injection from the ground launch can be achieved, in a way that the subsequent transfers are optimized. However, the decision on which orbit should be first is still up for decision for the optimizer.

The output of the optimization tool is the Pareto front shown in Figure 5. In the plot, two types of solutions are differentiated: 1) *P-T-P* solutions, in which the spacecraft performs the delivery of all the satellites at a given height by phasing (*P*), then transfers to the other orbit (*T*), and delivers the remaining ones by phasing manoeuvres again, before disposing itself; and 2) *Intermittent P-T* solutions, in which the spacecraft phases and/or transfers without deploying all the satellites on one operational orbit first. As expected, it is observed that in the case in which the satellites of the two altitudes are released sequentially (one orbit and then the other) the cost in fuel mass required is lower, as the more expensive manoeuvre (the Lambert targeting) is done only once within the release sequence. However, this comes at the cost of longer times, as complete revolutions have to be completed to perform up to 4 phasing manoeuvres. On the other hand, performing the *P-T intermittent* release sequence allows for shorter deployment times, as there is no need to wait full orbital periods' orders of time to inject the payload. Instead, by performing a sequence of Lambert targettings it manages to jump from orbit to orbit as needed. This comes, of course, with the subsequent increase in the cost in terms of fuel mass due to the higher number of more expensive manoeuvres (which in addition include inclination changes). Nevertheless, it is observed how the algorithm can provide an optimal result which follows the logic of what one could expect the delivery sequence to be, providing a high-level proof of its correct performance.

Case 2: Differentiated orbits mission

After this first case, we analyze the possibility of a set of different individual missions which use the upper-stage under consideration as a ride-share but require to be deployed into their differentiated orbits. As such, the Kepler elements and satellite sizes differ among each other, giving a slightly more complex problem for the optimizer. As the previous case, the system is assumed to have deployed the first satellite and needs to be manoeuvred into a final disposal orbit at 300 km. A summary of the orbits to be visited is given in Table 2, where again the disposal orbit is identified as **[D]**.

The solution provided by the optimization for this second case is the Pareto front shown in Figure 6. It is observed how 5 different combinations are found to be optimal. The least fuel-consuming strategy is such that it tries

DOI: 10.13009/EUCASS2022-7312

Table 1: Orbits of the constellation mission (case 1)

Orbit ID	**h [km]**	**e [-]**	**i [deg]**	**Ω [deg]**	***ω* [deg]**	**Δ*θ* [deg]**
[1]	600	0	97.79	54	-	120
[2]	800	0	98.60	54	-	120
[D]]	300	0	98	54	-	-

Table 2: Orbits of the differentiated orbits mission (case 2)

Orbit ID	**h [km]**	**e [-]**	**i [deg]**	**Ω [deg]**	***ω* [deg]**	**m_{sat} [kg]**
[1]	750	0	90.0	54	-	50
[2]	650	0	88.5	54	-	60
[3]	680	0	89.2	54	-	80
[4]	720	0	90.6	54	-	70
[5]	600	0	90.1	54	-	80
[6]	700	0	91.5	54	-	50
[D]	300	0	90	54	-	-

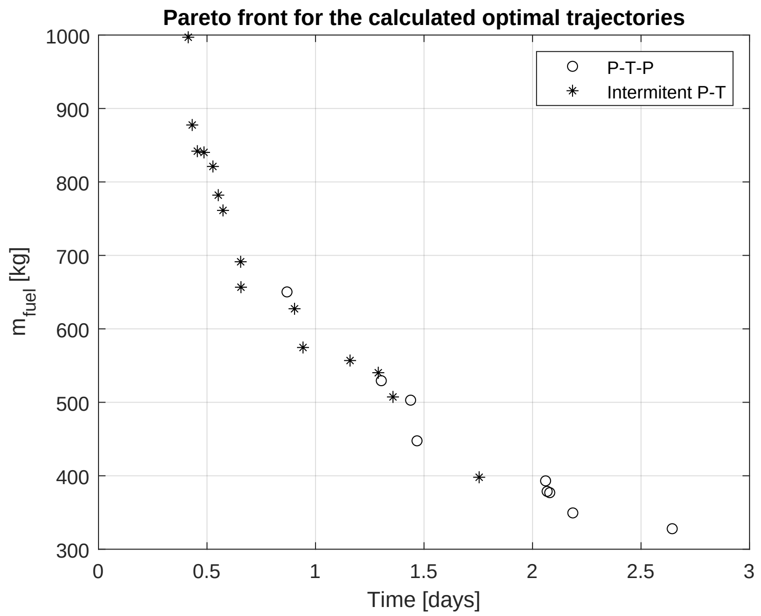

Figure 5: Pareto front of solutions for Case 1

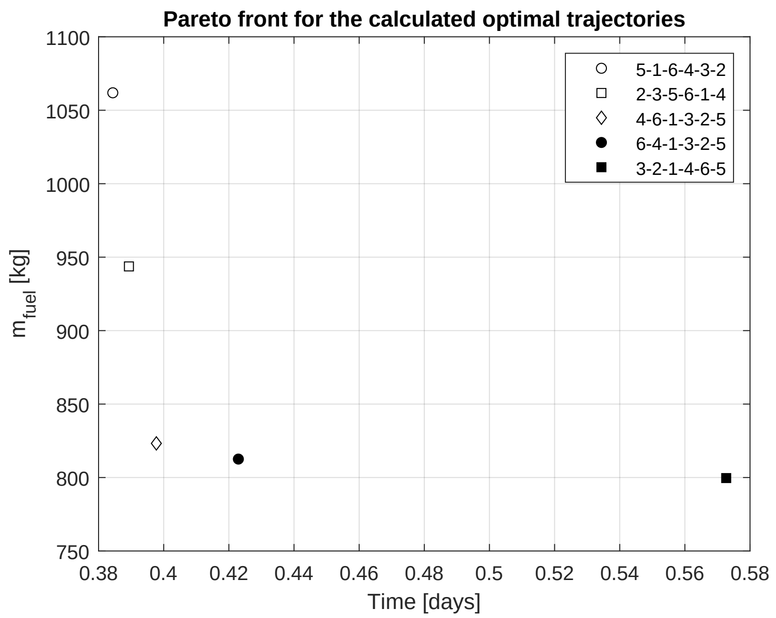

Figure 6: Pareto front of solutions for Case 2

DOI: 10.13009/EUCASS2022-7312

to follow the inclination change (while deploying first one of the bigger payloads), in a way that it first deploys the two satellites with inclinations below 90º and then those above that value. In addition, it shows how the sequence followed from orbit 1 up to the deployment is of reducing altitudes progressively, as to finish as close as possible to the disposal altitude. On the other hand, the fastest strategy follows the logic of deploying higher satellites first and then continue decreasing the height until disposal, with the only exception of starting the trajectory at orbit 5, as to get rid of a heavy satellite first. Similarly to the previous sequence, this one deploys the satellites with inclinations above 90º first, and then those with inclinations below that value. Among the intermediate-valued cases, some remarks are also to be made. There's two cases which are identical except for a swap of orbits 4 and 6 at the start. The solution starting at orbit 6 follows a similar strategy to the minimum-time one, in which the deployment is tried to be made in decreasing altitudes, starting from orbit 1, however it does not follow the previous trend in terms of inclinations, following a steady decreasing order except for orbit 5. Finally, the last case does not follow the same trend in terms of altitude, but does deliver the satellites according to the above-below 90º inclination trend. However, it is observed how it delivers faster the two heavier payloads in order to decrease the overall weight and decrease the fuel consumption. However this is proven to perform relatively worse than selecting a strategy in terms of the actual manoeuvres. It can be concluded, therefore, that while the weight of the payloads to be delivered affects the logic of delivery sequence selection, it is not as important as the geometric orbit parameters which drive the cost of the manoeuvres in terms of ΔV. However, it is possible that such small effect is due to the small differences in satellite masses, and could be aggravated if they become larger.

5. Conclusions

This paper proposes an optimization tool which allows to decide on the sequence and transfers between orbits to deliver multiple payloads into specific differentiated orbits minimizing both the fuel consumption and the total mission time. Such a tool is constructed in a bi-level manner, in a way that the integer and the nonlinear programming problems are separated for a simpler solution implementation. For the outer loop, a multi-objective P-ACO algorithm is used, which exploits both the tree-shape structure inherent to the sequential problem and the advantages in terms of computational time of heuristics. This algorithm uses as cost function the output of the inner loop, which in its turn uses a MOPSO algorithm to optimize the transfers given a sequence determined by the P-ACO. This inner loop also uses an heuristic algorithm, but this time one that can exploit the continuous nature of the transfer problem, the MOPSO. The proposed solution approach is such that the complete set of transfers are optimized at the same time, instead of one by one, to reduce the effect a certain manoeuvre will have on the subsequent ones in terms of optimization variables. In addition, three types of possible manoeuvres are envisaged, namely Lambert targeting, phasing and RAAN change through the J2 secular effect, which are used based on the Kepler elements of two consecutive orbits. Both of these conditions require the in-loop construction of the sequence in order to be able to properly generate the cost function, which also increases the flexibility of possible missions as the algorithm needs no previous knowledge of their structure. Being the inner loop also a bi-objective optimization strategy, a decision function was proposed in a way that each ant presents a unique cost in terms of fuel mass and total mission time.

Two distinct case scenarios were then solved to show the performance of the tool. Firstly, a constellation mission of 6 satellites in two Sun-synchronous orbits at different heights was studied. This analysis showed how the algorithm does not necessarily propose to deliver those of one orbit and then those of the other one. Indeed, while this type of mission sequence does provide with lower fuel consumption solutions, shorter times can be attained if the upper stage moves in between those two orbits more than once. Secondly, a mission with satellites of different masses and orbits was studied, in a case were all orbits were circular and shared the same RAAN value. It was shown how the tool tends to look for trends in the injection of the satellites either in terms of altitude or inclination in order to reduce the fuel consumption by moving progressively. It was also noted how the mass of the payloads to be delivered influenced in the selection of the first orbit or, in one case, of the complete sequence. However, the effect on the trajectory calculated of the payload masses was small compared to that of the orbit geometries. Nevertheless, by solving both types of missions, the performance of the optimization algorithm was proven, as well as its flexibility to solve different types of mission.

6. Acknowledgments

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.

DOI: 10.13009/EUCASS2022-7312

References

[1] Euroconsult. *Satellites to Be Built and Launched by 2028. A Complete Analysis and Forecast of Satellite Manufacturing and Launch Services*. Tech. rep. 2019.

[2] OneWeb. *OneWeb Website*. URL: https://oneweb.net/ (visited on 16/02/2022).

[3] Moog Inc. *Small Launch Orbital Maneuvering Vehicle*. URL: https://www.moog.com/markets/space/omv/slomv.html (visited on 16/02/2022).

[4] Spaceflight Inc. *Sherpa Program: New Orbital Transfer Vehicles Launch Smallsats to Custom Orbital Destinations*. URL: https://spaceflight.com/sherpa/ (visited on 16/02/2022).

[5] Dario Izzo et al. 'Evolving Solutions to TSP Variants for Active Space Debris Removal'. In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: Association for Computing Machinery, 2015, pp. 1207–1214. DOI: 10.1145/2739480.2754727.

[6] Yang Chen et al. 'Optimal mission planning of active space debris removal based on genetic algorithm'. In: *IOP Conference Series: Materials Science and Engineering* 715 (Jan. 2020), p. 012025. DOI: 10.1155/2021/8284022.

[7] K. Daneshjou, A.A. Mohammadi-Dehabadi and M. Bakhtiari. 'Mission planning for on-orbit servicing through multiple servicing satellites: A new approach'. In: *Advances in Space Research* 60.6 (2017), pp. 1148–1162. DOI: 10.1016/j.asr.2017.05.037.

[8] Brent W. Barbee et al. 'Design of spacecraft missions to remove multiple orbital debris objects'. In: *2011 Aerospace Conference*. 2011, pp. 1–14. DOI: 10.1109/AERO.2011.5747303.

[9] Bruce A. Conway, Christian M. Chilan and Bradley J. Wall. 'Evolutionary principles applied to mission planning problems'. In: *Celestial Mechanics and Dynamical Astronomy* 97 (Feb. 2007), pp. 73–86. DOI: 10.1007/s10569-006-9052-7.

[10] M. Cerf. 'Multiple Space Debris Collecting Mission—Debris Selection and Trajectory Optimization'. In: *Journal of Optimization Theory and Applications* 156.3 (Mar. 2013), pp. 761–796. DOI: 10.1007/s10957-012-0130-6.

[11] Nicolas Bérend and Xavier Olive. 'Bi-objective optimization of a multiple-target active debris removal mission'. In: *Acta Astronautica* 122 (2016), pp. 324–335. DOI: 10.1016/j.actaastro.2016.02.005.

[12] 'A biobjective branch and bound procedure for planning spatial missions'. In: *Aerospace Science and Technology* 73 (2018), pp. 269–277. DOI: 10.1016/j.ast.2017.11.040.

[13] Jeffrey Stuart, Kathleen Howell and Roby Wilson. 'Application of multi-agent coordination methods to the design of space debris mitigation tours'. In: *Advances in Space Research* 57.8 (2016), pp. 1680–1697. DOI: 10.1016/j.asr.2015.05.002.

[14] Tianjiao Zhang et al. 'Ant Colony Optimization based design of multiple-target active debris removal mission'. In: *2018 SpaceOps Conference*. DOI: 10.2514/6.2018-2412.

[15] Haiyang Li and Hexi Baoyin. 'Optimization of Multiple Debris Removal Missions Using an Evolving Elitist Club Algorithm'. In: *IEEE Transactions on Aerospace and Electronic Systems* 56.1 (2020), pp. 773–784. DOI: 10.1109/TAES.2019.2934373.

[16] Lorenzo Federici, Alessandro Zavoli and Guido Colasurdo. 'Evolutionary Optimization of Multirendezvous Impulsive Trajectories'. In: *International Journal of Aerospace Engineering* 2021 (May 2021), pp. 1–19. DOI: 10.1155/2021/9921555.

[17] Lorenzo Federici, Alessandro Zavoli and Guido Colasurdo. 'A Time-Dependent TSP Formulation for the Design of an Active Debris Removal Mission using Simulated Annealing'. In: *ArXiv* abs/1909.10427 (2019).

[18] M. Cerf. 'Multiple Space Debris Collecting Mission—Debris Selection and Trajectory Optimization'. In: *Journal of Optimization Theory and Applications* 156.03 (Mar. 2013), pp. 761–796. DOI: 10.1007/s10957-015-0705-0.

[19] J. M. Bourjolly, O. Gurtuna and A. Lyngvi. 'On-orbit servicing: A time-dependent, moving-target traveling salesman problem'. In: *International Transactions in Operational Research* 13 (Aug. 2006), pp. 461–481. DOI: 10.1007/s10569-006-9052-7.

[20] Jun Bang and Jaemyung Ahn. 'Two-phase framework for near-optimal multi-target Lambert rendezvous'. In: *Advances in Space Research* 61.5 (2018), pp. 1273–1285. DOI: 10.1016/j.asr.2017.12.025.

[21] Jing Yu, Xiaoqian Chen and Lihu Chen. 'Optimal planning of LEO active debris removal based on hybrid optimal control theory'. In: *Advances in Space Research* 55 (2015), pp. 2628–2640. DOI: 10.1016/j.asr.2015.02.026.

[22] AP Alpatov and Yu M Holdsht. 'On the choice of the ballistic parameters of an on-orbit service spacecraft'. In: *Technical Mechanics* (2019).

[23] Richard H Battin. *An introduction to the mathematics and methods of astrodynamics*. AIAA, 1999. DOI: 10.2514/4.861543.

[24] R. H. Gooding. 'A procedure for the solution of Lambert's orbital boundary-value problem'. In: *Celestial Mechanics and Dynamical Astronomy* 48 (June 1990), pp. 145–165. DOI: 10.1007/BF00049511.

[25] E. R. Lancaster and R. C. Blanchard. *A unified form of Lambert's theorem*. Tech. rep. NASA TN D-5368. NASA, 1969.

[26] Dario Izzo. 'Revisiting Lambert's Problem'. In: *Celestial Mechanics and Dynamical Astronomy* (Mar. 2014). DOI: 10.1007/s10569-014-9587-y.

[27] Rody Oldenhuis. *Robust solver for Lambert's orbital-boundary value problem*. https://github.com/rodyo/FEX-Lambert/releases/tag/v1.4. Github; accessed January 20, 2022. 2022.

[28] Howard D. Curtis. In: *Orbital Mechanics for Engineering Students (Third Edition)*. Ed. by Howard D. Curtis. Third Edition. Boston: Butterworth-Heinemann, 2014. ISBN: 978-0-08-097747-8. DOI: 10.1016/B978-0-08-097747-8.05001-5.

[29] W.D. McClain and D.A. Vallado. *Fundamentals of Astrodynamics and Applications*. Space Technology Library. Springer Netherlands. ISBN: 9780792369035.

[30] Michael Guntsch and Martin Middendorf. 'A Population Based Approach for ACO'. In: *Applications of Evolutionary Computing*. Ed. by Stefano Cagnoni et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 72–81. DOI: 10.1007/3-540-46004-7_8.

[31] Michael Guntsch and Martin Middendorf. 'Solving Multi-criteria Optimization Problems with Population-Based ACO'. In: *Evolutionary Multi-Criterion Optimization, Second International Conference (EMO'03)*. Ed. by C.M. Fonseca et al. LNCS. Berlin, Heidelberg: Springer, 2003, pp. 464–478. DOI: 10.1007/3-540-36970-8_33.

[32] Luís F. Simões et al. 'Multi-rendezvous Spacecraft Trajectory Optimization with Beam P-ACO'. In: *Evolutionary Computation in Combinatorial Optimization*. Lecture Notes in Computer Science. Springer International Publishing, 2017, pp. 141–156. DOI: 10.1007/978-3-319-55453-2_10.

[33] I.S. Grigoriev and M.P. Zapletin. 'GTOC5: Problem statement and notes on solution verification'. In: *Acta Futura* 8 (2014), pp. 9–19. DOI: 10.2420/AF08.2014.9.

[34] C.A.C. Coello, G.T. Pulido and M.S. Lechuga. 'Handling multiple objectives with particle swarm optimization'. In: *IEEE Transactions on Evolutionary Computation* 8.3 (2004), pp. 256–279. DOI: 10.1109/TEVC.2004.826067.

[35] Víctor Martínez-Cagigal. *Multi-Objective Particle Swarm Optimization (MOPSO)*. https://www.mathworks.com/matlabcentral/fileexchange/62074-multi-objective-particle-swarm-optimization-mopso. MATLAB Central; accessed January 25, 2022. 2022.

[36] K.F. Wakker. *Fundamentals of Astrodynamics*. TU Delft Library, 2015. ISBN: 978-94-6186-419-2.