DOI: 10.13009/EUCASS2019-469

DOI: ADD DOINUMBER HERE

Impulsive Multi-Rendezvous Trajectory Design and Optimization

Lorenzo Federici⋆*, Alessandro Zavoli*⋆† *and Guido Colasurdo*⋆
⋆*Department of Mechanical and Aerospace Engineering, Sapienza University of Rome*
Via Eudossiana 18 - 00184, Rome, Italy
lorenzo.federici@uniroma1.it · alessandro.zavoli@uniroma1.it · guido.colasurdo@uniroma1.it
†Corresponding author

Abstract

This paper investigates methods for optimizing an impulsive multi-rendezvous trajectory aimed at performing a complete tour of a prescribed set of targets. A simple heuristic is adopted for estimating the cost of each transfer leg. A genetic algorithm is used for optimizing both target sequence and rendezvous epochs. This approach is tested firstly by assuming that all transfers have the same duration. Then, the time domain is discretized over a finer grid, allowing a more appropriate sizing of the time-window allocated for each leg. Numerical results for a set comprising up to 15 targets are presented.

1. Introduction

A multi-rendezvous (MRR) trajectory concerns the motion of an active spacecraft (chaser) which executes a sequence of maneuvers with the aim of performing a complete tour of a prescribed set of targets (e.g., space debris or asteroids). MRR trajectories are gradually increasing in popularity within the aerospace community, as missions based on a chaser spacecraft "hopping" among multiple bodies are becoming of practical interest. Typical operational scenarios involve on-orbit servicing/refueling of geostationary satellites, and Active Debris Removal (ADR) missions. ADR missions are particularly significant as they would allow to restore the functionality of the Low Earth Orbit (LEO) environment, close to being compromised by the huge amount of orbiting wreckage.[1] Each ADR mission may involve the removal of several dozens of fragments, in order to reduce the cumulative cost of the operation. As a result, the best use of the chaser propellant becomes mandatory, and consequently, a well-designed trajectory must be searched for.

A number of authors dealt with long term or time-free ADR missions aimed at removing a small number of debris from Sun synchronous orbits (at a rate of three to ten per year). These missions heavily rely on J_2 orbital perturbation for the alignment of the orbital planes of consecutive targets before starting the rendezvous maneuver, in order to reduce the mission cost.[2] However, such an operational scenario becomes impractical in presence of strict time-constraints or long debris sequences. In this respect, the present paper investigates the design of an ADR mission involving a tighter time-constraint and possibly longer target chains, assuming an impulsive thrust model for the chaser. The aim is to minimize the overall mission ΔV, while performing a complete tour of a prescribed set of targets that move on the same orbital plane at slightly different altitudes.

The single-target time-fixed rendezvous is a well-known problem in spaceflight mechanics. Several optimization methods have been proposed assuming either finite[3,4] or impulsive thrust.[5,6] In the latter case, four burns permit the achievement of the optimal solution, if close coplanar orbits are considered.[7] Extension to a series of consecutive rendezvous is straightforward if the sequence of targets is assigned *a priori*.[8] However, the combinatorial aspect introduced by allowing permutations of the target sequence changes radically the problem nature and further increases its complexity. In this respect, the MRR problem presents several analogies with the Traveling Salesman Problem,[9] a well-known problem in Operational Research, where the goal is finding the shortest tour which allows a salesman to visit a prescribed set of cities.

While similar, the problem here investigated is more complex than a standard TSP, as the cost associated with traveling between any two targets changes with time, due to the orbital motion. With this analogy in mind, several attempts have been made to find the optimal solution of the MRR problem by using algorithms developed in the context of Operational Research. Exhaustive, brute-force, approaches[10,11] and branch and bound search[12] have been attempted first, for the design of optimal ADR missions in LEO. However, the effectiveness of those methods is limited to small sets of targets, due to the *course of dimensionality*.[13] Meta-heuristic approaches have thus rapidly gained in

DOI: 10.13009/EUCASS2019-469

popularity, as they allow to find a sub-optimal solution in a reasonable, limited amount of time. Beam Search[14] and Ant Colony Optimization[2] have been widely exploited for both ADR and asteroid exploration mission planning. An hybridization of the two has also been discussed.[15] These methods attempt at chaining one target after the other, into a sequence, evaluating many (if not all) possible branches departing from a given starting node. In order to reduce the overall computational effort, simplistic transfer models are used for a fast evaluation of any target-to-target transfer cost; moreover, pre-determined encounter epochs are commonly employed.[16] Despite being rather flexible, this formulation usually underperforms in case a complete tour is searched for. Other methods, instead, rely on an iterative refinement of a set of (possibly random) initial solutions and guarantee the tour completeness at any point in the optimization process. Prominent examples are Tabu Search (TS),[17] Simulated Annealing (SA),[18] and Genetic Algorithm (GA).[19–21]

Among the aforementioned optimization methods, Genetic Algorithm is the most flexible one, as, in principle, it may handle any type of decision variable. GA performs a global optimization and, thanks to its stochastic selection and mutation operators, it has greater chances to evade from local optima than greedy methods. In addition, it may adopt an encoding that guarantees the intrinsic completeness of the tour. GA is a population-based method, hence its execution time can be significantly reduced by an efficient parallel implementation, that is not possible for single-solution methods like TS and SA. All these features make it a solid candidate for solving the complete tour problem here studied and motivate its choice.

The paper is organized as follows. Section 2 presents a comprehensive mathematical formulation of the impulsive multi-rendezvous problem, resulting in a Mixed-Integer Nonlinear Programming (MINLP) problem. Section 3 proposes a bi-level optimization approach aimed at solving it. A simple heuristic is proposed in Section 3.1, based on a sub-optimal four-impulse analytic solution of the single-target rendezvous problem, for estimating the cost of each leg connecting two assigned consecutive targets once initial time and maximum transfer duration are prescribed. The effectiveness of this sub-optimal solution is verified in a few relevant cases by comparing it with the optimal solution. On the basis of this heuristic, a combinatorial optimization problem, or touring problem, is defined, which shares some features with the TSP. Section 3.2 presents three formulations of the combinatorial problem with increasing complexity (which, in turn, provide increasing possibility of delivering the optimal solution of the original problem). Section 4 describes the Generic Algorithm here adopted for the simultaneous optimization of the target sequence and rough estimation of the rendezvous epochs. Once the best sequence and the approximate encounter times has been determined, the whole transfer is optimized through a multi-population self-adaptive Differential Evolution algorithm. Numerical results for a set comprising up to 15 targets are presented in Section 5. A conclusion section ends the paper.

2. Problem Statement

Let us consider a set of N prescribed targets that move on circular coplanar orbits at slightly different altitudes under a keplerian dynamical model. For each target body A_i, orbital radius r_{A_i} and right ascension at starting time $\theta_{A_i}(t_0)$ are assigned. The velocity is constant and equals to $v_{A_i} = \sqrt{\mu/r_{A_i}}$, while the angular position at any time is given by $\theta_{A_i}(t) = \theta_{A_i}(t_0) + \sqrt{\mu/r_{A_i}^3}\,(t - t_0)$, where μ is the gravitational parameter of the central body. The chaser is also assumed to be initially on a circular orbit of radius r_0 at the right ascension $\theta_0(t_0) = 0$, on the same orbital plane as all the targets. The problem is thus planar. The goal is to design an impulsive multi-rendezvous transfer trajectory that allows the chaser to perform a complete tour of a prescribed set of targets within a specified maximum time-length of the entire mission T_M, minimizing the overall mission ΔV.

Let us assume that a sequence $S_A = \{A_1, A_2, \ldots, A_N\}$ of N non-repeating bodies to encounter and the corresponding set of (monotonically increasing) encounter times $\boldsymbol{t} = \{t_1, t_2, \ldots, t_N\}$, with $t_N = T_M$, have been assigned, so that the integer $A_j \in [1, N]$ identifies the target met at time t_j. The overall trajectory of the chaser can be decomposed into a series of target-to-target body legs. The k-th leg departs from body A_k (with $A_0 = 0$) at time t_k and arrives at the body A_{k+1} at time t_{k+1}, for $k = 0, \ldots, N-1$.

The rendezvous condition requires that, at the ending point of the leg, position and velocity of the chaser are the same as the target:

$$\boldsymbol{r}(t_k) = \boldsymbol{r}_{A_k}(t_k) = \boldsymbol{r}_k \qquad \forall\, k \in \{0, 1, \ldots, N\} \tag{1}$$

$$\boldsymbol{v}(t_k) = \boldsymbol{v}_{A_k}(t_k) = \boldsymbol{v}_k \qquad \forall\, k \in \{0, 1, \ldots, N\} \tag{2}$$

Being four the maximum number of impulses for an optimal time-constraint planar rendezvous,[7] each body-to-body transfer leg is made up of a sequence of three ballistic *arcs*, named "*a*", "*b*", and "*c*", joined by impulsive maneuvers located at the departure, at the two internal points labeled with subscripts "$k + 1/3$" and "$k + 2/3$", and at the arrival point, respectively.

A position formulation is here considered, that is, the trajectory is parameterized with respect to radii $r_{k+1/3}$, $r_{k+2/3}$ and anomalies $\theta_{k+1/3}$, $\theta_{k+2/3}$ at the internal maneuvering points. Spacecraft velocities immediately before $\boldsymbol{v}^-_{k+1/3}$,

DOI: 10.13009/EUCASS2019-469

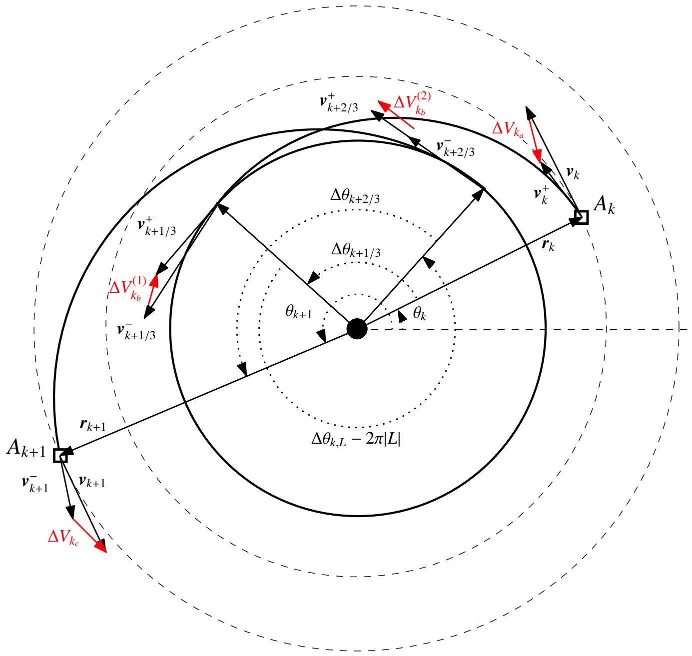
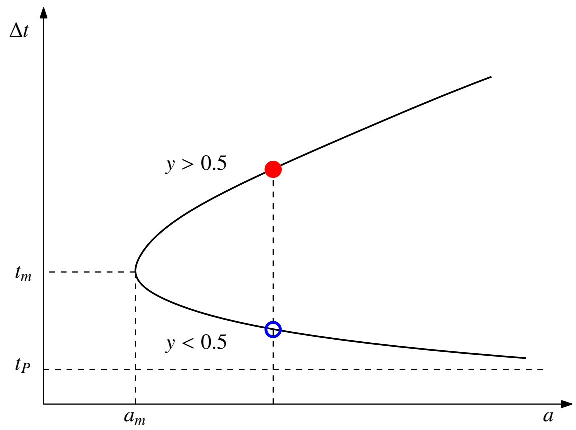


Figure 1: Trajectory sketch for the k-th leg.

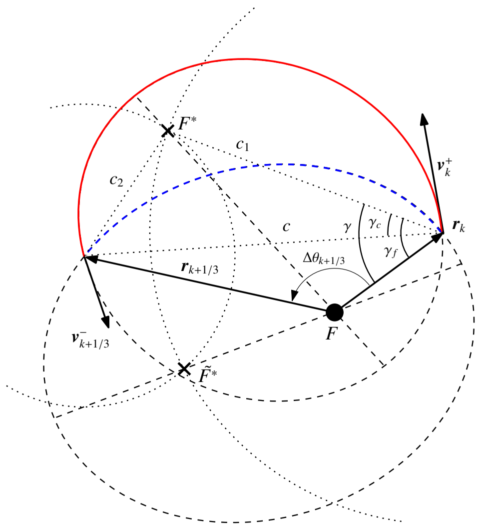
$\boldsymbol{v}^-_{k+2/3}$ or after $\boldsymbol{v}^+_{k+1/3}$, $\boldsymbol{v}^+_{k+2/3}$ the maneuvers are found by solving either a geometrical problem or a Lambert problem.

2.1 Geometrical Problem

Let us first consider an arc "a" connecting the points "k" and "$k + 1/3$". Two families of ellipses that connects $\boldsymbol{r}_k$ and $\boldsymbol{r}_{k+1/3}$ exist. They might be parameterized as a function of the semi-major axis and labeled as *fast* and *slow* families.[22]



(a) Transfer ellipse flight time and semi-major axis as a function of the y parameter.



(b) Geometrical construction of the transfer ellipse.

Figure 2: Geometrical Problem. Filled marker refers to the slow solution, corresponding to the solid-line transfer arc; empty marker refers to the fast, dashed-line solution.

Let us introduce a non-dimensional parameter $y \in [0, 1]$ so that:

$$a = \frac{a^{min}}{4y\,(1 - y)} \tag{3}$$

DOI: 10.13009/EUCASS2019-469

where $a^{min} = (r_k + r_{k+1/3} + c)/4$ is the smallest semi-major axis which connects the ending points, and $c = \left\| \boldsymbol{r}_k - \boldsymbol{r}_{k+1/3} \right\|$ is the chord distance between the arc endpoints. Pairing the fast solutions with values $y < 1/2$, and the slow solutions with values $y > 1/2$, the elliptic arc connecting the two assigned endpoints is uniquely identified for any given choice of y. Figure 2(b) depicts the geometric construction that allows an unambiguous definition of the transfer ellipse, that is, semi-major axis a, eccentricity e and argument of pericenter ω, as soon as y and $\Delta\theta$ are known. Chord lengths $c_1 = 2a - r_k$, and $c_2 = 2a - r_{k+1/3}$ are evaluated first. Angle γ follows from:

$$\gamma = \operatorname{acos}\left(\frac{r_k^2 - r_{k+1/3}^2 + c^2}{2 r_k \, c} \right) \tag{4}$$

hence $\gamma_f = \gamma - \gamma_c$, where:

$$\gamma_c = \begin{cases} \operatorname{acos}\left(\dfrac{c_1^2 - c_2^2 + c^2}{2 c_1 c} \right) & \text{if } y \le 0.5 \\ -\operatorname{acos}\left(\dfrac{c_1^2 - c_2^2 + c^2}{2 c_1 c} \right) & \text{if } y > 0.5 \end{cases} \tag{5}$$

Eventually, the eccentricity is found as

$$e = \frac{\sqrt{c_1^2 + r_k^2 - 2 c_1 \, r_k \cos \gamma_f}}{2a} \tag{6}$$

Velocities at both endpoints ($\boldsymbol{v}_k^+$ and $\boldsymbol{v}_{k+1/3}^-$) follow from standard equations of the two-body problem. Transfer time Δt_a can also be evaluated and, consequently, the epoch at the intermediate maneuver $t_{k+1/3} = t_k + \Delta t_a$ is obtained. The same procedure holds for the arc connecting points "$k + 2/3$" and "$k + 1$", resulting in a similar geometrical definition of the third arc "c". As a result, one has

$$\left[\boldsymbol{v}_k^+, \, \boldsymbol{v}_{k+1/3}^- \right] \leftarrow yArc\left(\boldsymbol{r}_k, \boldsymbol{r}_{k+1/3}, y_{k,a} \right) \tag{7}$$

$$\left[\boldsymbol{v}_{k+2/3}^+, \, \boldsymbol{v}_{k+1}^- \right] \leftarrow yArc\left(\boldsymbol{r}_{k+2/3}, \boldsymbol{r}_{k+1}, y_{k,c} \right) \tag{8}$$

The cost of the maneuvers at the departure and arrival points are thus evaluated as:

$$\Delta V_{k_a} = \left\| \boldsymbol{v}_k^+ - \boldsymbol{v}_k \right\| \tag{9}$$

$$\Delta V_{k_c} = \left\| \boldsymbol{v}_{k+1}^- - \boldsymbol{v}_{k+1} \right\| \tag{10}$$

2.2 Multi-revolution Lambert problem

The central arc "b", which connects points "$k + 1/3$" and "$k + 2/3$", cannot be dealt with in the same fashion. In fact, for a given choice of the parameters $y_{k,a}$ and $y_{k,c}$ the maneuvering epochs $t_{k+1/3}$ and $t_{k+2/3}$, and, consequently, the travel time, are assigned. A multi-revolution Lambert problem can be formulated, being the position vectors $\boldsymbol{r}_{k+1/3}$, $\boldsymbol{r}_{k+2/3}$ and the travel time $\Delta t = t_{k+2/3} - t_{k+1/3}$ known. This problem admits $1 + 2n_{max}$ solutions, where n_{max} is the maximum allowed number of revolutions; one solution for the 0-revolution transfer arc and two additional solutions, namely left and right branch, for each n-revolution transfer orbit. Let us introduce an integer parameter $L \in [-n_{max}, n_{max}]$ indicating the solution corresponding to the $|L|$-revolution transfer orbit and side $sign(L)$, positive for the right branch, negative for the left one. For the L-th solution, the velocity vectors immediately after the second impulse $\boldsymbol{v}_{k+1/3}^+$ and just before the third one $\boldsymbol{v}_{k+2/3}^-$ can be evaluated as according to the algorithm by Izzo,[23] that is:

$$\left[\boldsymbol{v}_{k+1/3}^+, \boldsymbol{v}_{k+2/3}^- \right] \leftarrow \text{Lambert}\left(\boldsymbol{r}_{k+1/3}, \boldsymbol{r}_{k+2/3}, t_{k+2/3} - t_{k+1/3}; L \right) \tag{11}$$

Hence, the total cost of the two internal maneuvers is:

$$\Delta V_{k_b} = \left\| \boldsymbol{v}_{k+1/3}^+ - \boldsymbol{v}_{k+1/3}^- \right\| + \left\| \boldsymbol{v}_{k+2/3}^+ - \boldsymbol{v}_{k+2/3}^- \right\| \tag{12}$$

Instead of treating L as an optimization variable, an enumeration approach could be used, that is, all the $2n_{max} + 1$ possible solutions are computed and the one with the lowest total ΔV is chosen. However, as we are considering transfers between close orbits, we can make an educated guess and safely restrict the analysis to just three scenarios, that is the chaser performs the same number of revolution as the target, one more, or one less, corresponding to six solutions (three right and three left branches).

DOI: 10.13009/EUCASS2019-469

2.3 MINLP formulation

According to the proposed formulation, the overall trajectory can be parameterized by using a set of $8\,N$ parameters, that is:

$$\boldsymbol{x} = \bigcup_{k=1}^{N} \boldsymbol{x}_k \tag{13}$$

where:

$$\boldsymbol{x}_k = \{A_k,\ t_k\} \cup \{r_{k+1/3},\ \Delta\theta_{k+1/3},\ y_{k,a}\} \cup \{r_{k+2/3},\ \Delta\theta_{k+2/3},\ y_{k,c}\} \tag{14}$$

with $S_A = \{A_1, A_2, \dots, A_N\}$ a permutation of N, not-repeated, targets. Eventually, the impulsive time-constrained MRR optimization problem can be formulated as:

$$P \quad = \begin{cases} \min\limits_{\boldsymbol{x}} \Delta V_{tot}(\boldsymbol{x}) \\ \text{s.t.} \quad \boldsymbol{x}_L \le \boldsymbol{x} \le \boldsymbol{x}_U \end{cases} \tag{15}$$

where the overall cost of the MRR trajectory is:

$$\Delta V_{tot} = \sum_{k=0}^{N-1} (\Delta V_{k_a} + \Delta V_{k_b} + \Delta V_{k_c}) \tag{16}$$

and $\boldsymbol{x}_L, \boldsymbol{x}_U$ are the lower and upper bounds of the design variables, respectively. This problem involves the simultaneous optimization of both integer variables (defining the encounter sequence) and real-value decision variables (such as, radius and anomaly at the maneuvers) and it is thus labeled as a Mixed-Integer Nonlinear Programming (MINLP) problem.

3. Bi-Level Optimization Approach

The MINLP problem in Eq. (15) belongs to the class of NP-hard problems, hence no deterministic algorithm exists for finding the optimal solution in polynomial-time. A variety of stochastic meta-heuristic techniques have been developed over the last decades aiming at attaining a (often sub-optimal) good-quality solution in a reasonable, limited amount of time. However, as the problem dimension increases, the required computational time may become prohibitive.

Instead of solving the problem *as a whole*, one might attempt to decompose the problem into simpler sub-problems, that could be (more or less easily) solved separately, and eventually their solutions can be recomposed into the original problem solution. For the problem at hand, a bi-level approach can be pursued, by isolating i) an outer level that concerns the definition of the encounter sequence and a (possibly rough) evaluation of the epochs at each encounter, while details of each body-to-body transfer leg are neglected; ii) an inner level which deals with the optimization of each body-to-body transfer with full details, assuming that departure and arrival bodies are assigned; encounter epochs may or may not be fixed.

The two layers are interconnected: the outer layer requires a measure of the cost associated to each transfer leg for "weighting" the quality of a certain encounter sequence, even though the actual ΔV of each leg can be evaluated only by solving the full-transfer optimization problem, that is, the inner-layer problem. On the other hand, the inner layer requires the definition of the encounter sequence and rendezvous epochs, which, in turn, is the output of the combinatorial, outer-layer problem. In practice, the two problems might be solved sequentially provided that a way, that is, an heuristic, exists for attaining a reasonable estimate of the transfer cost without solving the full optimization problem. Once the heuristic has been established, the outer-level combinatorial problem is isolated and solved first; its solution is then used as initial guess for the inner-level problem.

3.1 Cost Estimate for a Single Rendezvous Leg

This section presents an analytical, sub-optimal, four-impulse strategy to assess the ΔV of a trajectory leg, for any assigned pair of departure and arrival bodies that fly on circular orbits, which fairly approximates the behavior of the time-fixed optimal solution when the allowed travel time is sufficiently large.[24]

Assuming that departure and arrival orbits are not too far apart, the minimum-ΔV solution is represented by a Hohmann transfer, possibly preceded and/or followed by coasting arcs on departure/arrival orbits which allow the correct phasing required by this kind of maneuver. Let $\theta_{1,0}$ (respectively, $\theta_{2,0}$), be the true anomaly at time $t = 0$ of the departure (respectively, arrival) body, flying on circular orbits of radius r_1 (respectively, r_2). The departure coasting arc

DOI: 10.13009/EUCASS2019-469

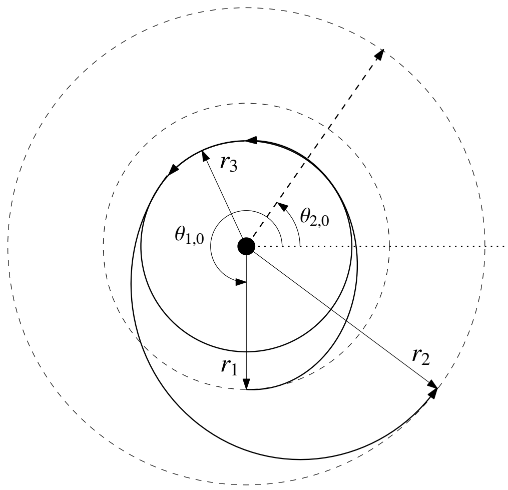


Figure 3: Chaser trajectory according to the sub-optimal rendezvous strategy adopted as heuristic.

duration, that is, the time T_{wait} required to attain the correct phase $\gamma^\star = \pi - \omega_2 T_{H_{12}}$ between departure and arrival body can be evaluated as:

$$T_{wait} = \frac{\Delta\gamma}{-\dot{\gamma}} = \frac{\theta_{2,0} - \theta_{1,0} - \pi + \omega_2 T_{H_{12}}}{\omega_1 - \omega_2} \qquad \text{if} \quad r_1 < r_2 \tag{17}$$

$$T_{wait} = \frac{2\pi - \Delta\gamma}{\dot{\gamma}} = \frac{\theta_{2,0} - \theta_{1,0} - \pi + \omega_2 T_{H_{12}} - 2\pi}{\omega_1 - \omega_2} \qquad \text{if} \quad r_2 < r_1 \tag{18}$$

By comparing the available maximum transfer time T_{max} with the sum of the waiting time T_{wait} plus the time spent on the Hohmann transfer $T_{H_{12}} = \pi\sqrt{a_{12}^3/\mu}$, one obtains a condition for the availability of the Hohmann transfer:

$$T_{wait} + T_{H_{12}} \leq T_{max} \tag{19}$$

Whenever Eq. (19) holds, the cost of the transfer leg is easily evaluated as ΔV_h of the Hohmann transfer. If the Hohmann transfer is not possible, the mission scheme depicted in Figure 3 is adopted: the maneuvering spacecraft is injected into a circular (either internal or external) waiting orbit of radius r_3 with an Hohmann transfer "1-3", of semi-major axis $a_{13} = (r_1 + r_3)/2$ and duration $T_{H_{13}} = \pi\sqrt{a_{13}^3/\mu}$, in order to adjust its phase with respect to the target body. A second Hohmann transfer "3-2", of semi-major axis $a_{23} = (r_2 + r_3)/2$ and duration $T_{H_{23}} = \pi\sqrt{a_{23}^3/\mu}$ is then used to close the rendezvous. The rendezvous equation, which imposes the equality of chaser and target position at the end of the maneuver, is enforced:

$$\Delta\theta_0 = (T_{max} - T_{H_{13}} - T_{H_{23}})\,\omega_3 - T_{max}\omega_2 + 2k_{rev}\pi \tag{20}$$

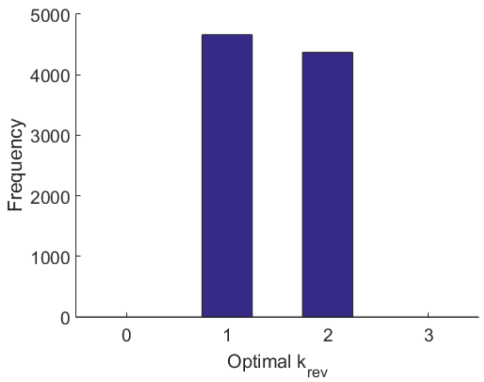
with $\Delta\theta_0 = \theta_{2,0} - \theta_{1,0} \in [0, 2\pi]$.

This nonlinear equation in r_3 admits a family of solutions, parameterized by $k_{rev} \in \mathcal{N}$, that accounts for the (possibly different) number of revolutions performed by the maneuvering spacecraft with respect to the target. However, we are interested to the minimum ΔV solution only, therefore we can safely restrict our search to the cases corresponding to the largest inner orbit ($k_{rev} = 1$) and the smallest external orbit ($k_{rev} = 2$). Depending on the initial relative phasing of the two bodies, relative angular velocity, and maximum allowed travel time, either solution might be the best one. Both solutions are thus evaluated and compared each other. The one with the lower cost is retained and the corresponding ΔV is used as a cost estimate.

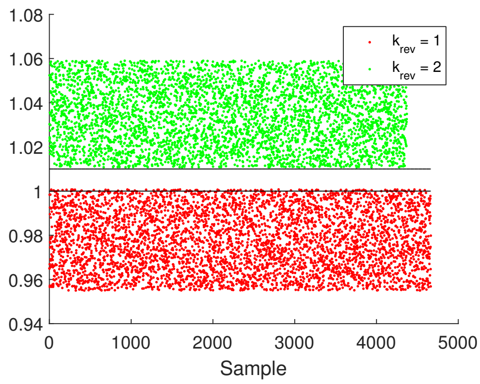
Figure 4(a) presents the results of a Monte Carlo analysis over 10^5 trials with randomly chosen target body semi-major axis (normalized with respect to r_1), departure phase, and maximum travel time, showing a relatively equal amount of cases where internal and external solution are optimal. Figure 4(b) presents the distribution of r_3 over a number of trials for an orbit rising maneuver, with assigned $r_2 > r_1$, confirming that waiting orbits are internal for $k_{rev} = 1$ and external in case of $k_{rev} = 2$.

The effectiveness of this sub-optimal solution has been verified by comparing the heuristic value with the optimal solution provided by full optimization in a few relevant cases. Typical differences are in the range of $0.2 \div 3\%$, with a few peaks at $+10\%$ when the travel times becomes too short for this mission scheme to be practical. The absolute

DOI: 10.13009/EUCASS2019-469



(a) Relative frequency of the optimal value of the parameter k_{rev}.



(b) Normalized distribution of the inner radius r_3 for a number of orbit raising maneuvers.

Figure 4: Monte Carlo simulation with randomly chosen target body radius, initial phasing, and maximum travel time.

values of the difference usually do not exceed 10 m/s, and thus are deemed reasonable. With respect to other heuristics, the proposed approach provides the additional benefit of being related to a real, feasible, mission scheme. As a result, the heuristic value represents a conservative estimate of the actual ΔV.

3.2 Outer-Level Optimization

3.2.1 Time-Free Tour

Under the assumption that both departure and arrival bodies fly on coplanar circular orbits and that the allowed mission time is sufficiently large, the optimal transfer is always a Hohmann transfer. The cost $\Delta V(i, j)$ for moving from a departure body i to an arrival body j is the same, regardless of the specific departure/arrival epochs. The problem thus reduces to the search for the sequence of encountered bodies that minimizes the total velocity increment.

Let $\boldsymbol{p} = \{p_1, p_2, \ldots, p_N\} \in \mathcal{P}^N$ be a permutation of N, non repeated, positive, integer elements $\{1, 2, \ldots, N\}$. The *time-free tour* optimization problem can be written as:

$$\min_{\boldsymbol{p}\in\mathcal{P}^N} \sum_{k=1}^{N} \Delta V(p_{k-1}, p_k) \tag{21}$$

where $p_0 = 0$ denotes the chaser. Transfer cost for any pair of arrival/departure bodies can be preliminary evaluated and collected in a matrix of transfer costs $\Delta V_{i,j}$ of dimensions $i \in [1..N] \cup \{0\}$ by $j \in [1..N]$.

An attentive reader will notice that this problem is quite similar to the well-known traveling salesman problem, for which a large literature is available. For this peculiar problem, fast solution techniques exist for solving problems involving up to a few hundreds of "entries".

3.2.2 Time-Fixed Time-Uniform Tour

The solution of the previous problem provides a lower bound on the overall tour cost, but may not provide a reasonable guess to the solution of the original problem. In fact, the perfect phasing required by the Hohmann transfer might never occur due to the existing time-constraint on the overall mission duration. For time-fixed rendezvous maneuvers, transfer costs are highly sensitive to the initial phasing, that is, to the departure epoch, and to the allowed transfer duration. However, under the assumption that the duration of all transfers is exactly the same, a relaxed problem can be formulated. The epoch of the k-th encounter is readily available as $t_k = k\,T_M/N$. The problem reduces again to the search for the optimal permutation of the target sequence $\boldsymbol{p} \in \mathcal{P}^N$, but, in this case, the cost associated to the transfer from a target i to a target j also depends on the position k of the leg into the sequence. The *time-fixed time-uniform tour* problem can be written as:

$$\min_{\boldsymbol{p}\in\mathcal{P}^N} \sum_{k=1}^{N} \Delta V(p_{k-1}, p_k, k) \tag{22}$$

DOI: 10.13009/EUCASS2019-469

where $p_0 = 0$ denotes the chaser, and $\Delta V(i, j, k)$ indicates the cost associated to the transfer from a target i to a target j, starting at time t_{k-1} and ending at time t_k. All transfer costs can be preliminary evaluated and collected in a 3-dimensional array $\Delta V_{i,j,k}$.

Following the analogy established between the time-free tour and the TSP, this combinatorial optimization problem is similar to the Time-Dependent TSP, where the cost for moving from a city to another varies with time (that is, the position in the sequence). Despite being harder to solve, some general-purpose techniques for the standard TSP still apply to the TD-TSP.[25]

3.2.3 Time-Fixed Time-Discrete Tour

Eventually, if the encounter epochs are discretized over a finer time-grid, one obtains a combinatorial optimization problem with a solution closer to the solution of the original problem and a formulation similar, to a certain extent, to the two previously defined touring problems.

Let $\boldsymbol{\tau} = \{\tau_0, \tau_1, \ldots, \tau_{N^s}\}$ be a discrete time grid with uniform, equidistant points, where $N^s = N\,D$, and D is the number of divisions introduced into each "previously considered" time-slot. $\Delta V(i, j, \tau_h, \tau_{h+m})$ denotes the cost for moving from body i to body j, departure epoch τ_h, and arrival epoch τ_{h+m}. In this case, both the target sequence and the location of the encounter epochs over the grid are searched for. However, all decision variables can be collapsed into "one" decision variable, that is a permutation $\boldsymbol{\Pi} \in \mathcal{P}^{N^s}$ with a number of elements equals to the (discrete) number of available encounter epochs. Permutation elements of value greater than N are considered as blanks, thus revealing the encounter sequence $\boldsymbol{p}$, and the position of non-blank elements reveals a vector of encounter epochs $\boldsymbol{t} = [t_1, t_2, \ldots, t_N]$, where the element t_k is the arrival epoch at the body p_k. An example is proposed in Figure 5 for $N = 5$ and $D = 3$, showing a permutation $\boldsymbol{\Pi} \in \mathcal{P}^{N^s}$ which reveals a target sequence $\boldsymbol{p} = \{1, 3, 2, 4, 5\}$ and encounter epochs $\boldsymbol{t} = \{\tau_2, \tau_5, \tau_7, \tau_{11}, \tau_{15}\}$.

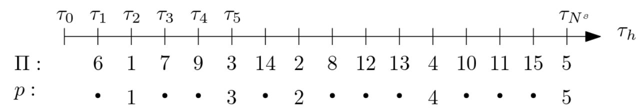


Figure 5: An example of a permutation encoding/decoding for an 5x3 Time-Fixed Time-Discrete tour.

The *time-fixed time-discrete tour* problem can be thus written as:

$$\min_{\boldsymbol{\Pi} \in \mathcal{P}^{N^s}} \sum_{k=1}^{N} \Delta V(p_{k-1}, p_k, t_{k-1}, t_k) \tag{23}$$

$$\boldsymbol{p} = \{\Pi_h \mid \Pi_h \le N \qquad \forall h \in [1, N^s]\} \tag{24}$$

$$\boldsymbol{t} = \{\tau_h = h\Delta T \mid \Pi_h \le N \qquad \forall h \in [1, N^s]\} \tag{25}$$

with $p_0 = 0$ being the chaser, and $\Delta T = T_M/(ND)$ the time unit of the time grid.

Even though the problem is now well different from a TSP, the proposed formulation still poses the problem as a permutation optimization problem (on a higher dimension), and thus it can be solved with the same algorithm used for the other touring problems.

Remark 1

The number of divisions D should be kept small, because i) a rough evaluation of the encounter epochs is sufficient, as the attained solution will be further refined within the inner-level optimization step; ii) a large number of divisions makes the problem too similar to the original MINLP, hence more difficult to solve; iii) the proposed heuristic works well if there is enough time to perform several revolutions; reducing the minimum valid travel time $\Delta T = T_M/(ND)$ makes the heuristic less reliable and may undermine our efforts. As a result, a number of divisions $D = 2$ or $D = 3$ appears as a good trade-off value for the problem at hand.

Remark 2

As in the other touring problems, one may pre-compute all transfer costs for speeding up the function evaluation (hence the whole optimization process). A 4D tensor of dimensions $(N + 1) \times N \times (N\,D - 1) \times (N(D - 1) - 1)$ is needed in principle. However, one may notice the monotonic, non-increasing, behavior of ΔV with transfer time and decide to limit the calculus to transfers of duration $M\Delta t$, assuming the same cost for longer transfer. A 4D tensor of dimensions $(N + 1) \times N \times (ND - 1) \times M$ would now be required. Apart from reducing the size of the tensor, this treatment has the

DOI: 10.13009/EUCASS2019-469

additional benefit of guiding the solver toward a trajectory with more uniform travel times, which might be good from an operational point of view.

3.3 Inner-Level Optimization

Assuming that the target sequence S_A has been selected and a rough estimation of the encounter epochs is available, the MINLP problem in Eq. (15) reduces to a NLP problem. Two scenarios can be investigated: i) encounter epochs t_k are kept fixed at their nominal values $\bar{t}_k$; ii) encounter epochs are free to be optimized. In the first case, each body-to-body transfer can be solved independently from the others, thus reducing the $8N$ problem to solving N easier sub-problems each one of dimension 8, being N the prescribed number of targets to encounter. In the second case, encounter epochs may vary in a neighborhood of the reference value, leading to an improved solution, but the whole trajectory must be fully optimized. Lower and upper bounds of the encounter epochs are selected so that:

$$t_k \in \left[\bar{t}_k - \frac{\bar{t}_k - \bar{t}_{k-1}}{2}, \bar{t}_k + \frac{\bar{t}_{k+1} - \bar{t}_k}{2}\right] \tag{26}$$

In the present paper, both scenarios are dealt with by using an in-house optimization code that implements a multi-population self-adaptive Differential Evolution algorithm with a synchronous island-model for parallel computation, that have been developed in the contest of the Global Trajectory Optimization Competitions[26,27] and successfully applied to other multi-encounter space trajectory optimization problems.[28] DE is a population-based evolutionary algorithm (EA), firstly introduced by R. Storn and K. Price in 1997[29] as a method to find the global optimum of non-linear, non-differentiable functions defined over a continuous parameter space. As indicated by a recent study,[30] DE exhibits much better performance in comparison with several others continuous-variable meta-heuristic algorithms on a wide range of real-world optimization problems, despite its simplicity. Being inspired by evolution of species, it exploits the operations of crossover, mutation and selection to generate new candidate solutions. However, unlike traditional EAs and GAs, the mutated solutions are generated as scaled differences of distinct individuals of the current population. This self-referential mutation tends to automatically adapt the different variables of the problem to the natural scale of the solution landscape, so improving the search potential of the algorithm. With respect a standard DE implementation, a self-adaptive scheme is implemented for automatically adjusting the values of scaling and crossover parameters. Moreover, a multi-population paradigm, where each population evolves according to a specific mutation rule, allows to achieve a nice balance between exploration and exploitation of the search space.

4. Genetic Algorithm

A Genetic Algorithm (GA) is used in the present paper for solving the combinatorial optimization problems described in the previous sections. Genetic algorithms are well-known population-based meta-heuristic techniques inspired by natural evolution. They have been successfully applied to a wide range of real-world problems of significant complexity. A brief overview of the algorithm is here presented. Interested readers are suggested to refer to Reference 31 for further details. A block diagram of the basic algorithm is proposed in Figure 6, highlighting the presence of three fundamental genetic operators (selection, crossover, and mutation) which form the backbone of any GA implementation.

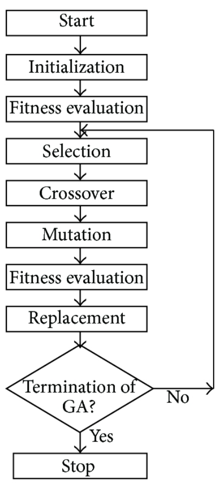


Figure 6: Flow chart of Genetic Algorithm

Algorithm 1 PMX

```
Choose two random index a, b so that 0 ≤ a < b ≤ L
for all i ∈ [a, b] do
    o1[i] ← p2[i]
end for
c ← p2[a : b]
for all i ∈ [1, a − 1] ∪ [b + 1, N] do
    J ← p1[i]                                   ▷ Proposal
    while J ∈ c do
        find iJ so that p2[iJ] == J             ▷ map p1 into p2
        J ← p1[iJ]                              ▷ New proposal
    end while
    o1[i] ← J                                   ▷ Accept the proposal
end for
```

Figure 7: Partially Mapped Crossover algorithm

DOI: 10.13009/EUCASS2019-469

At the beginning, an initial *population*, that is, a collection of solutions (often referred as *individuals* or *chromosomes*) is randomly generated, with an attempt to cover the search space as best as possible. At each iteration (i.e., *generation*), the fitness of each individual in the population is evaluated first. Then, a mating population is created by selecting a number of individuals from the current population, according to a *selection operator*, which tries to promote individuals with a good fitness, without sacrificing diversity too much. Typical selection operators are Stochastic Roulette Wheel and Tournament,[31] the latter being the one used in the present application. Next, pairs of individuals (parents) are randomly chosen from the mating population and combined for producing new solutions (offsprings) according to a *crossover* rule. The underlying idea is that combining good solutions in some way allows to create new, and hopefully better, solutions. The process is repeated until a new population (usually with the same dimension of the previous one) is created. Eventually, a few randomly-chosen individuals of the offspring population undergo a mutation process, with the aim of increasing the population diversity. This new population eventually replaces the previous one and the process is repeated until some termination criterion (e.g., maximum number of generation) has been met. As a minor, yet important, tweak, *Elitism* is enforced, that is, at the end of each generation, the N_p^e best individuals of the parent population are copied into the offspring population, thus preserving the best solution from being accidentally lost in the evolution process. Some control over the population diversity is often needed. The aim is to avoid an excessive uniformity between individuals that would result in a very inefficient use of the crossover operator, hence in a waste of function evaluations. To overcome this problem, an *epidemic* mechanism is introduced. A diversity metric is defined as the sum over the population of the infinity-norm between any pair of individuals in the search space. If the diversity score falls below a given threshold, a large part of the population dies, that is, it is randomly re-initialized. This mechanism cannot happens more than $N^{epidemic}$ times and not two times within a number $N_G^{epidemic}$ of generations.

While selection operators are usually problem-independent, crossover and mutation operators are tightly related to the adopted *encoding*, that is, the way the real-word problem is described in terms of numerical variables. Depending on the features of the decision variables, several encodings have been defined, with the most common being binary, integer-valued, and real-valued encoding. In order to match the formulation developed in Section 3.2, a permutation encoding is here considered, that is, each *individual* or *chromosome* is a permutation $\boldsymbol{p} \in \mathcal{P}^L$. Each gene takes an integer value in the range $[1, \ldots, L]$, and two genes in the same *individual* can not have the same value at one time. Standard, general-purpose, crossover operators (such as, one-point, two-point, or uniform crossover) cannot be applied to a permutation without producing unfeasible offspring, that is, introducing multiple copies of the "entries". However, several permutation-preserving genetic operators, which have been developed for the permutation encoding in the context of the TSP problem, can be employed for solving the problem under investigation, such as Partially Matched Crossover, Cycle Crossover, Order Crossover, Non-Wrapping Order Crossover, as well as many others.[32,33]

The Partially Matched Crossover (PMX) devised by Goldberg[34] has been adopted in the present paper, as it aims to preserve both the order and position of as many entries as possible from the parents. A Pseudo-code of this algorithm is presented in Figure 7. An offspring is created starting from a sub-string (or *cut*) of the first parent (p_1), in the same fashion as in a two-point crossover. The remaining entries are taken from the other parent (p_2). Entries not appearing in the cut are kept in the same position as they appear in p_2. Conflicts are resolving using p_1 as a map for p_2.

Mutation operators are also of great relevance in the framework of genetic algorithms, as they allow to escape from situations of premature stagnation of the population on a sub-optimal solution, and their impact on the effectiveness of the algorithm should not be underestimated. Once again, an abundance of permutation-preserving mutation operators are documented in literature. In the present application, a reverse mutation operator is adopted: whenever an individual $\boldsymbol{p}$ must be mutated, two indices $i < j$ are randomly chosen and the new individual $\boldsymbol{p}^{new}$ is created as a copy of the original one, but for the genes from i to j that are copied in a reversed order, that is $\boldsymbol{p}^{new}(i : j) = \boldsymbol{p}(j : i)$.

4.1 GA hyperparameter

The performance of a genetic algorithm clearly depends on the choice of selection, mutation and crossover operators, but also on several "hyper-parameters", such as population size (n_P), number of generations (n_G), crossover probability (p_c), that represents the percentage of the parents replaced by offspring, mutation rate (p_m), that is, the probability that one individual will undergo a random mutation, and other operator-specific parameters, such as the depth of the reverse operator (i.e., the maximum number of element of the swath that will be reversed). A proper tuning of these hyperparameters may greatly improve the solution capability of GA. However, the optimal tuning of a genetic algorithm is documented to be problem dependent, and represents an NP-hard problem by itself. Therefore, a preliminary tuning is usually done on a simplified, possibly scaled, version of the optimization problem of interest, with the hope to capture all the relevant features of the original problem.

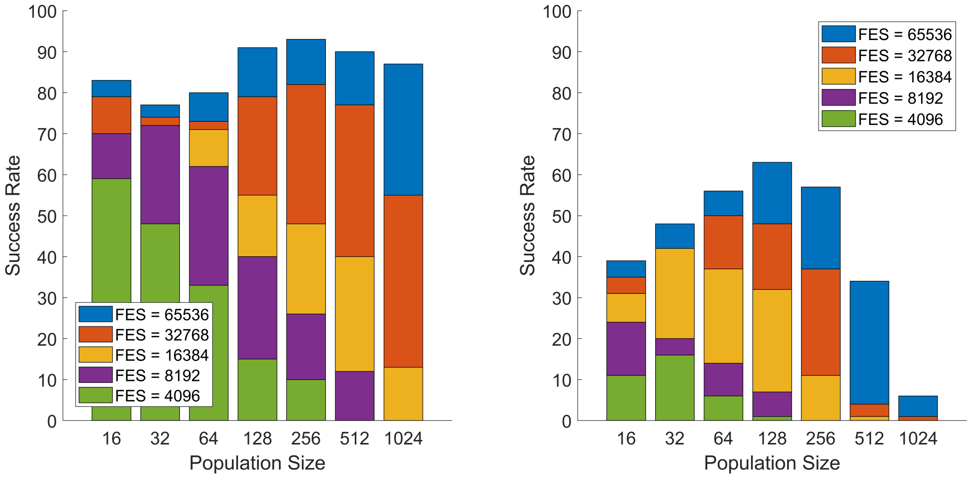
DOI: 10.13009/EUCASS2019-469

5. Numerical Results

In this section numerical results are proposed for various touring missions of pre-determinate length. Orbital parameters of both chaser and targets are provided in Table 1.

	Chaser	Targets														
ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
r [km]	7000	6900	6910	6930	6940	6950	6960	6980	7010	7020	7030	7050	7060	7070	7080	7090
θ_0 $[deg]$	0	-5	10	15	35	-30	-10	25	20	-25	-15	5	-35	30	-20	-40

Table 1: Chaser and targets initial orbital parameters.



(a) 10-target mission

(b) 12-target mission

Figure 8: Success rate as a function of population size, for an assigned number of function evaluations

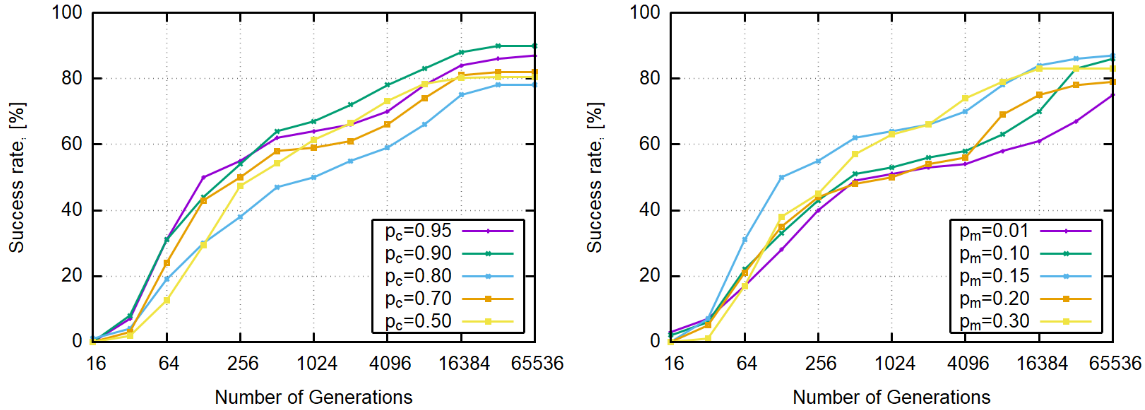
A preliminary analysis has been carried out for assessing the effectiveness of the proposed genetic algorithm for the combinatorial problems formulated in Section 3.2. The time-fixed time-uniform tour of dimension N reduces to the search of the optimal permutation $p^\star \in \mathcal{P}^N$ among the $N!$ existing ones. Instances of the time-fixed time-uniform tour for a number of targets $N \leq 12$ are sufficiently small that allow an exhaustive search to be completed in a reasonable amount of time. The influence of population size (n_P) and number of generations (n_G) for missions with 8, 10 and 12 targets has been studied. In particular, it is interesting to understand if, for a given amount of function evaluations $FES = n_G \cdot n_P$, an optimal allocation of trials exists. Figure 8 highlights the influence of the population size (n_P) on the success rate assuming the maximum number of function evaluations ($FES = n_G \cdot n_P$) fixed. It is apparent that the choice of the population size is not significant for low-dimension problems, such as the 10-target mission. Instead, for the 12-target mission, a population of 64 or 128 elements seems to correspond to the "best" allocation of trials.

Effects of crossover (p_c) and mutation (p_m) probability on the success rate for the 12-target mission have been also investigated. Figure 9 presents the success rate versus the number of generations for the 12-target time-uniform tour. For each configuration, 100 independent runs have been performed. According to the attained results, the best configuration appears to be $p_c = 0.9$ and $p_m = 0.15$. Also, the algorithm performance appears almost insensitive to variations of these hyperparameters.

The same genetic algorithm has been also adopted for the time-fixed time-discrete tour, which is significantly more difficult than the corresponding time-uniform version, but should provide a better starting guess to solve the original problem. The optimal solution of the time-discrete tour cannot be attained by a brute-force search in a reasonable amount of time, as the number of possible permutations for a $N \times D$ tour problem goes roughly as $(ND)!$. For the sake of conciseness, only the results concerning two cases, that is, the 8-target and 15-target missions, are here presented. For each of these missions, a number of independent runs of the genetic algorithm have been carried out. The best found solution is elected as putative optimum and used as an initial guess for the inner-level optimization problem, with the aim of approaching a solution sufficiently close to the unknown global optimum in terms of the required total velocity increment.

The inner-level optimization has been carried out by using a 4-island DE optimization engine, with 5 N_{var} agents per tribe and a maximum number of generations equal to 10000, being N_{var} the number of continuous decision vari-

DOI: 10.13009/EUCASS2019-469



(a) Effect of the crossover rate, for $p_m = 0.15$

(b) Effect of the mutation rate, for $p_c = 0.95$

Figure 9: Sensitivity analysis on the success rate of GA by varying the crossover probability (p_c) and the mutation rate (p_m) for the 12-target mission, with population size $n_P = 64$.

ables of the considered problem. A comparison between the minimum ΔV obtained through the GA, the DE with fixed encounter times and the DE with free encounter times, for the 8-target missions and the 15-target missions, are presented in Tables 2(a) and 2(b), respectively. Instead, the complete results for the 8-target and 15-target missions in terms of optimal target sequence, encounter times and ΔVs are reported in Tables 3 and 4, respectively. It is possible to note that the difference between the GA and fixed-times DE solutions is in the order of 1-2%, that is the typical error of the employed heuristic; thus, results confirm its effectiveness. The difference in the minimum ΔV is far greater (up to 20%) if the DE algorithm is left free to modify the encounter times. Indeed, in this case, the algorithm is able to perform a more appropriate sizing of the time window allocated for each leg, shortening or extending legs as needed. As a result, the obtained trajectories are composed, for the great part, by Hohmann transfers and 3-impulse legs; a few 4-impulse legs are anyway traveled when the time window is too short and a fast transfer is thus required. This result is clearly visible in Figure 10, which shows the radius versus time along the whole transfer of the chaser for the 15×3 mission.

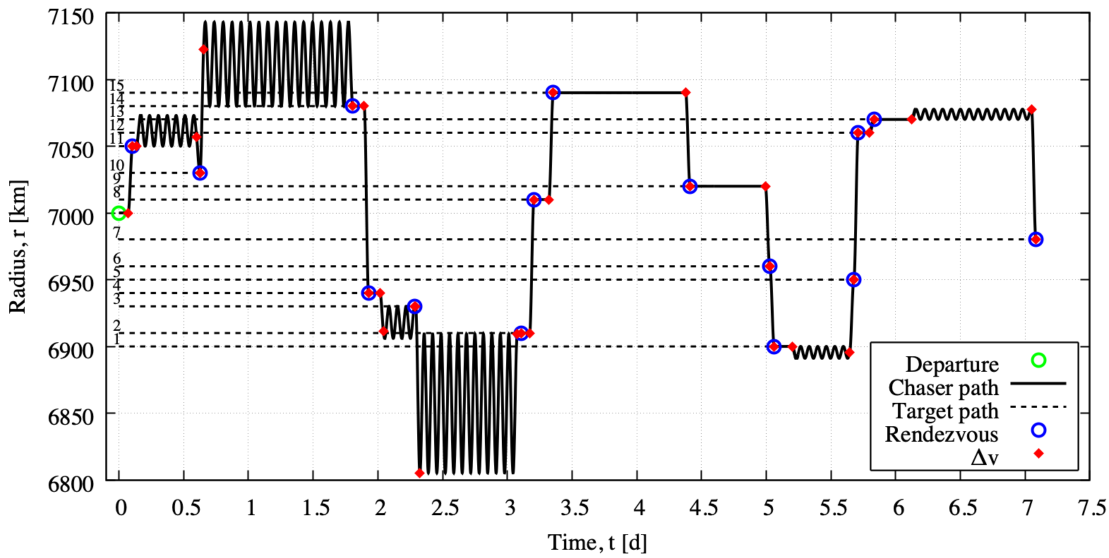


Figure 10: Radius r vs. time t for the 15x3 solution.

DOI: 10.13009/EUCASS2019-469

(a) 8-target missions

	GA	DE time-fixed	DE time-free
8x1: Δv_{tot} = km/s	0.5061	0.4931	0.4408
8x2: Δv_{tot} = km/s	0.3767	0.3722	0.3587
8x3: Δv_{tot} = km/s	0.3344	0.3296	0.2969

(b) 15-target missions

	GA	DE time-fixed	DE time-free
15x1: Δv_{tot} = km/s	0.8016	0.7860	0.6356
15x3: Δv_{tot} = km/s	0.6638	0.6287	0.6217

Table 2: Attained solutions at various stages of the optimization procedure. GA refers to the solution of the outer-level problem, DE (time-fixed) and DE (time-free) refer to the refined solution attained after completing the inner-level optimization, assuming the encounter epochs respectively fixed or free to be optimized.

Table 3: Optimal 8-target sequences.

Mission 8x1: Δv_{tot} = 0.44092 km/s								
ID	8	6	7	5	4	3	2	1
t [d]	0.5050	1.0851	1.5569	1.8889	2.4740	2.6984	3.3499	3.7777
Δv [km/s]	0.0340	0.0270	0.0109	0.0163	0.1450	0.0481	0.0951	0.0644
Mission 8x2: Δv_{tot} = 0.35851 km/s								
ID	6	8	5	7	4	3	2	1
t [d]	0.2830	0.5894	0.7794	1.1100	1.6894	1.9526	2.8445	3.7777
Δv [km/s]	0.0217	0.0270	0.0324	0.0644	0.1275	0.0054	0.0556	0.0244
Mission 8x3: Δv_{tot} = 0.29694 km/s								
ID	1	3	2	4	7	5	6	8
t [d]	0.0788	0.4176	0.6928	0.7870	1.5407	1.6361	1.7625	3.7777
Δv [km/s]	0.0545	0.0325	0.0294	0.0165	0.0808	0.0163	0.0054	0.0616

Table 4: Optimal 15-target sequences.

Mission 15x1: Δv_{tot} = 0.63545 km/s															
ID	6	7	2	1	10	9	4	3	14	8	12	13	5	11	15
t [d]	0.2361	0.6654	1.1236	2.1038	2.3611	3.0694	3.2726	3.8059	4.0351	4.7222	5.4189	5.6867	6.3350	6.6241	7.0833
Δv [km/s]	0.0217	0.0110	0.1206	0.0055	0.0706	0.0291	0.0433	0.0380	0.0807	0.0374	0.0267	0.0112	0.0645	0.0539	0.0212
Mission 15x3: Δv_{tot} = 0.62149 km/s															
ID	11	10	14	4	3	2	8	15	9	6	1	5	12	13	7
t [d]	0.1342	0.6262	1.8067	2.0162	2.2921	3.1081	3.2776	3.9645	4.5971	5.0277	5.2001	5.6771	5.7980	6.1231	7.0833
Δv [km/s]	0.0268	0.0229	0.0598	0.0753	0.0186	0.0694	0.0543	0.0426	0.0373	0.0324	0.0328	0.0323	0.0592	0.0053	0.0524

6. Conclusions

A bi-level optimization procedure has been proposed for the design of the multi-rendezvous trajectory of a chaser spacecraft visiting all the objects in a prescribed set. The goal is to minimize the overall propellant consumption, while completing the tour within a given amount of time. The features of the combinatorial, outer-level problem, i.e., the problem concerning the definition of the optimal encounter sequence together with a preliminary evaluation of the epochs at each encounter, have been highlighted. A formulation of the outer-level problem, which permits different time-lengths for each rendezvous maneuver, was also presented. The growth in the problem complexity is rewarded by an increased capability of attaining a solution closer to the optimum of the complete MINLP problem. A genetic algorithm with a permutation-preserving encoding has been used to solve the combinatorial optimization problem. A simple, sub-optimal, analytic solution of the single-target rendezvous problem was adopted as heuristic for a fast evaluation of the ΔV associated to each leg, without studying it in full details.

The attained solution was further refined by assuming the encounter sequence fixed and optimizing the multi-impulse rendezvous maneuver: each body-to-body transfer is described by means of a peculiar parameterization based on the position of the impulses, whose total velocity increment is minimized. Numerical solutions are presented for a set comprising up to 15 target bodies. Results suggest that, by coupling the proposed ΔV heuristic with a time-discrete time-fixed formulation with time-discretization factor 3, one attains a trajectory that is very close to the solution of the full mixed-integer nonlinear programming problem, whereas the overall computational effort is significantly reduced. Future research on the problem at hand will be direct to assess the effectiveness of other meta-heuristic algorithms, such as Simulated Annealing and Tabu-Search, either in place or together with Genetic Algorithms. The use of local optimization operators to speed up the convergence will be also investigated, leveraging on available classical TSP heritage.

DOI: 10.13009/EUCASS2019-469

References

[1] J.-C. Liou and N.L. Johnson. Instability of the present leo satellite populations. *Advances in Space Research*, 41(7):1046 – 1053, 2008.

[2] Hong-Xin Shen, Tian-Jiao Zhang, Lorenzo Casalino, and Dario Pastrone. Optimization of active debris removal missions with multiple targets. *Journal of Spacecraft and Rockets*, 55(1):181–189, 2017.

[3] Binfeng Pan, Ping Lu, and Zheng Chen. Coast arcs in optimal multiburn orbital transfers. *Journal of Guidance, Control, and Dynamics*, 35(2):451–461, March 2012.

[4] Xincheng Yue, Ying Yang, and Zhiyong Geng. Continuous low-thrust time-optimal orbital maneuver. In *Proceedings of IEEE Conference on Decision and Control*, pages 1457–1462, December 2009.

[5] J E Prussing and J H. Chiu. Optimal multiple-impulse time-fixed rendezvous between circular orbits. *Journal of Guidance Control and Dynamics*, 9(1):17–22, 1986.

[6] K. Mirfakhraie and B. A. Conway. Optimal cooperative time-fixed impulsive rendezvous. *Journal of Guidance Control Dynamics*, 17:607–613, May 1994.

[7] John E Prussing. *Optimal multiple-impulse orbital rendezvous.* PhD thesis, Massachusetts Institute of Technology, 1967.

[8] Fanghua Jiang and Gao Tang. Systematic low-thrust trajectory optimization for a multi-rendezvous mission using adjoint scaling. *Astrophysics and Space Science*, 361(4):117, Mar 2016.

[9] Eugene L Lawler, Jan Karel Lenstra, AHG Rinnooy Kan, David Bernard Shmoys, et al. *The traveling salesman problem: a guided tour of combinatorial optimization*, volume 3. Wiley New York, 1985.

[10] Federico Zuiani and Massimiliano Vasile. Preliminary design of debris removal missions by means of simplified models for low-thrust, many-revolution transfers. *International Journal of Aerospace Engineering*, 2012, 2012.

[11] Vitali Braun, A Lüpken, S Flegel, J Gelhaus, M Möckel, C Kebschull, C Wiedemann, and P Vörsmann. Active debris removal of multiple priority targets. *Advances in Space Research*, 51(9):1638–1648, 2013.

[12] Max Cerf. Multiple space debris collecting mission—debris selection and trajectory optimization. *Journal of Optimization Theory and Applications*, 156(3):761–796, 2013.

[13] Jan Karel Lenstra and AHG Rinnooy Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981.

[14] Dario Izzo, Ingmar Getzner, Daniel Hennes, and Luís Felismino Simões. Evolving solutions to tsp variants for active space debris removal. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1207–1214. ACM, 2015.

[15] Luís F Simões, Dario Izzo, Evert Haasdijk, and AE Eiben. Multi-rendezvous spacecraft trajectory optimization with beam p-aco. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 141–156. Springer, 2017.

[16] Tianjiao Zhang, Hongxin Shen, Yikang Yang, Hengnian Li, and Jisheng Li. Ant colony optimization-based design of multiple-target active debris removal mission. *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES*, 61(5):201–210, 2018.

[17] Özgür Gürtuna and Jonathan Trépanier. On-orbit satellite servicing: a space-based vehicle routing problem. In *Operations Research in Space and Air*, pages 123–141. Springer, 2003.

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

[19] Yong Liu, Jianan Yang, Yizhou Wang, Quan Pan, and Jianping Yuan. Multi-objective optimal preliminary planning of multi-debris active removal mission in LEO. *Science China Information Sciences*, 60(7):072202, 2017.

[20] Junko Murakami and Shinji Hokamoto. Approach for optimal multi-rendezvous trajectory design for active debris removal. In *61th International Astronautical Congress,(pp. IAC-10. C1. 5.8). Prague, CZ*, 2010.

DOI: 10.13009/EUCASS2019-469

[21] S Hokamoto. Genetic-algorithm-based rendezvous trajectory design for multiple active debris removal. *28th International Symposium on Space Technology and Science*, 2011.

[22] Francesco Cacciatore and Chiara Toglia. Optimization of orbital trajectories using genetic algorithms. *Journal of Aerospace Engineering, Sciences and Applications*, 1(1):58–69, 2008.

[23] Dario Izzo. Revisiting lambert's problem. *Celestial Mechanics and Dynamical Astronomy*, 121(1):1–15, 2015.

[24] A. Zavoli and G. Colasurdo. Indirect optimization of finite-thrust cooperative rendezvous. *Journal of Guidance, Control, and Dynamics*, 38(2):304–314, 2015. cited By 6.

[25] Chryssi Malandraki and Mark S Daskin. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation science*, 26(3):185–200, 1992.

[26] G. Colasurdo, A. Zavoli, A. Longo, L. Casalino, and F. Simeoni. Tour of jupiter galilean moons: Winning solution of gtoc6. *Acta Astronautica*, 102:190–199, 2014. cited By 3.

[27] L. Casalino, G. Colasurdo, A. Zavoli, and M. Berga. Gtoc8: Results and methods of team 22. volume 158, pages 4291–4300. Univelt Inc., 2016.

[28] Lorenzo Federici, Alessandro Zavoli, and Guido Colasurdo. Preliminary capture trajectory design for europa tomography probe. *International Journal of Aerospace Engineering*, 2018, 2018.

[29] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[30] Swagatam Das, Sankha Subhra Mullick, and Ponnuthurai N Suganthan. Recent advances in differential evolution–an updated survey. *Swarm and Evolutionary Computation*, 27:1–30, 2016.

[31] Melanie Mitchell. *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*. MIT Press, 1998.

[32] Vincent A. Cicirello. Non-wrapping order crossover: An order preserving crossover operator that respects absolute position. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'06)*, volume 2, pages 1125–1131. ACM Press, July 2006.

[33] AJ Umbarkar and PD Sheth. Crossover operators in genetic algorithms: A review. *ICTACT journal on soft computing*, 6(1), 2015.

[34] David E Goldberg et al. Alleles, loci, and the traveling salesman problem. In *Proceedings of an international conference on genetic algorithms and their applications*, volume 154, pages 154–159. Lawrence Erlbaum, Hillsdale, NJ.