Research Article

Evolutionary Optimization of Multirendezvous Impulsive Trajectories

Lorenzo Federici, Alessandro Zavoli, and Guido Colasurdo

Department of Mechanical and Aerospace Engineering, Sapienza University of Rome, Via Eudossiana 18, 00184 Rome, Italy

Correspondence should be addressed to Alessandro Zavoli; alessandro.zavoli@uniroma1.it

Received 12 March 2021; Revised 19 April 2021; Accepted 29 April 2021; Published 27 May 2021

Academic Editor: Franco Bernelli-Zazzera

Copyright © 2021 Lorenzo Federici et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper investigates the use of evolutionary algorithms for the optimization of time-constrained impulsive multirendezvous missions. The aim is to find the minimum-ΔV trajectory that allows a chaser spacecraft to perform, in a prescribed mission time, a complete tour of a set of targets, such as space debris or artificial satellites, which move on the same orbital plane at slightly different altitudes. For this purpose, a two-level design approach is pursued. First, an outer-level combinatorial problem is defined, dealing with the simultaneous optimization of the sequence of targets and the rendezvous epochs. The suggested approach is first tested by assuming that all transfer legs last exactly the same amount of time; then, the time domain is discretized over a finer grid, allowing a more appropriate sizing of the time window allocated for each leg. The outer-level problem is solved by an in-house genetic algorithm, which features an effective permutation-preserving solution encoding. A simple, but fairly accurate, heuristic, based on a suboptimal four-impulse analytic solution of the single-target rendezvous problem, is used when solving the combinatorial problem for a fast guess at the transfer cost, given the departure and arrival epochs. The outer-level problem solution is used to define an inner-level NLP problem, concerning the optimization of each body-to-body transfer leg. In this phase, the encounter times are further refined. The inner-level problem is tackled through an in-house multipopulation self-adaptive differential evolution algorithm. Numerical results for case studies including up to 20 targets with different time grids are presented.

1. Introduction

A multirendezvous (MRR) trajectory concerns the motion of an active spacecraft, the chaser, which executes a sequence of maneuvers to rendezvous with multiple passive targets, such as space debris or artificial satellites in Earth’s orbit. MRR trajectories are gradually increasing in popularity within the aerospace community, since missions based on a chaser spacecraft visiting multiple artificial bodies are becoming of practical interest. Typical operational scenarios include Active Debris Removal (ADR) and On-Orbit Servicing (OOS) missions. ADR missions are particularly significant, as they are deemed to be the only effective option to ensure the long-term operation of spacecraft, close to being compromised by the huge amount of Earth orbiting wreckage [1]. On the other hand, servicing missions have recently drawn the attention of several private companies [2]. Indeed, the possibility of extending the life span of already operative satellites by sending servicing spacecraft for refueling, maintenance, and orbital relocation appears to be economically advantageous [3]. In both mission scenarios, in order to make the program economically affordable, any single chaser is expected to visit (i.e., remove or service) as many targets as possible in a limited amount of time, making the best use of the on-board propellant. As a result, a minimum-propellant multirendezvous trajectory has to be designed for the chaser spacecraft.

A number of authors addressed the optimization of ADR trajectories in Low Earth Orbit (LEO) or Sun Synchronous Orbit (SSO). Most of these studies focus on long-term or time-free ADR missions, typically aimed at deorbiting the target debris at a rate of three to ten per year [4, 5]. A typical mission strategy consists in exploiting the $J_2$ orbital perturbation for the alignment of the orbital planes of consecutive
targets before starting the rendezvous maneuver in order to reduce the total mission cost \[6, 7\], at the expense of a longer flight time. Conversely, when time-constrained or time-fixed missions are considered, the removal sequence and rendezvous epochs of the ADR mission must be optimized simultaneously, making the problem more complex to solve. For this reason, most of the literature deals with very small debris clusters \[8, 9\], usually composed of less than five elements. Notable exceptions are Ref. \[10, 11\], which consider clouds with dozens or hundreds of debris.

The optimal planning of OOS missions in the geosynchronous orbit was also investigated in several research papers; the same approach for the preliminary design of ADR mission, with a single active chaser aimed at multiple passive targets, can be also found in several works about OOS \[12\]. However, nonstandard, and even more complex, mission scenarios become more frequent in OOS literature, encompassing the presence of multiple servicing spacecraft \[13\], cooperative targets \[14\], a fuel depot \[15\], constellation reconfiguration maneuvers \[16\], or complex dynamical models \[17\]. As for ADR missions, the great part of the studies about OOS deals with small constellations of satellites or time-free missions, in order to reduce the problem complexity.

The targets of an MRR mission are usually selected out of a wide set, looking for favourable body positions (e.g., GTOC9 problem \[18\]); the expenditure to cancel out the phase error is negligible, and suitable solutions \[19, 20\] can be used to estimate the cost of the time-free orbit transfer. However, most of the proposed approaches become impractical in the presence of long target sequences or tight time constraints, as the ones considered in this work. In the present manuscript, the rendezvous cost is mainly due to the recovery of the phase error.

The single-target time-constrained rendezvous is a well-known problem in spaceflight mechanics. Several optimization methods have been proposed assuming either finite \[21\] or impulsive thrust \[22, 23\]. In the latter case, four burns permit the achievement of the optimal solution \[24\]. The extension of the solution method to a series of consecutive rendezvous is straightforward if the sequence of targets is assigned \textit{a priori} \[25\]. However, the combinatorial aspect introduced by allowing permutations of the target sequence changes radically the problem nature and increases drastically its complexity. In this respect, the MRR problem presents several analogies with the Traveling Salesman Problem (TSP) \[26\], a well-known problem in operational research, whose solution is the shortest tour which allows a salesman to visit once a prescribed set of cities, starting and ending the trip in his home city. For the classic TSP, branch-and-bound-based rapid exact solvers are available \[27\], which allow solving problem instances involving up to several hundred cities in a brief time. However, 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 dynamics.

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. This is the case of exact solution techniques such as brute-force approaches \[28\] and branch and bound search \[29\] which consist of a systematic enumeration of candidate solutions of the combinatorial problem. However, these solution methods are generally viable only in the presence of small sets of targets or in time-free scenarios, due to the \textit{curse of dimensionality}, which causes the search to run out of time and/or memory very soon as the problem dimension increases \[30\]. For this reason, heuristic and metaheuristic approaches have rapidly gained popularity, as they usually allow to find good-quality solutions in a limited amount of time, even in the case of large-size problems. Greedy heuristics, such as the Series Method \[31\], are among the fastest algorithms, but they can provide solutions quite far from the global optimum. Heuristic and metaheuristic tree-search procedures, such as Beam Search \[32\], Ant Colony Optimization (ACO) \[33\], or A* search \[34\], have been widely exploited for both ADR/OOS and asteroid exploration mission planning. A hybridization between Beam Search and ACO was also discussed \[35\]. These methods have proven capable to achieve high-quality results in a reduced computational time, especially when the aim of the mission is to visit the largest number of targets in a wider set, given some propellant and/or time constraints. Indeed, during the search, they allow pruning all those branches of the search tree that lead to unfeasible solutions. On the other hand, the performance of tree-search algorithms tends to worsen when a complete tour of the targets is searched for, as they repeatedly explore the same branches while looking for the optimal target permutation. Conversely, other metaheuristic methods consist of the iterative refinement of a set of (possibly random) initial solutions and guarantee the tour completeness at any point in the optimization process. For this reason, they are particularly suited to plan MRR missions aimed at visiting all the targets of the chosen set. Prominent examples are Tabu Search (TS) \[36\], Simulated Annealing (SA) \[37, 38\], and Genetic Algorithms (GA) \[39, 40\].

Among the aforementioned optimization methods, the genetic algorithm is the most flexible, 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 the solution of the complete tour problem here studied and motivate its choice.

This paper focuses on the optimization of a time-constrained multi-target rendezvous mission, assuming an impulsive thrust model for the chaser. The aim is to minimize the overall mission $\Delta V$, while performing a complete tour of a set of targets, which move on circular coplanar orbits with similar altitudes, in a prescribed amount of time. The problem is mathematically formulated as a Mixed-Integer Nonlinear Programming (MINLP) problem, involving the simultaneous optimization of both integer variables...
(that define the sequence of targets) and real-valued variables
(which identify the rendezvous epochs and the spacecraft
trajectory from a target to the next one). A two-level design
procedure is proposed to simplify the MINLP problem solu-
tion, by defining (i) an outer-level combinatorial problem
that concerns the determination of the optimal rendezvous
sequence and epochs, discretized over a time grid, and (ii)
an inner-level NLP problem, which optimizes each body-
to-body transfer leg and is also able to refine the encounter
times, while keeping the body sequence that solved the
outer-level problem. The task of solving the outer-level prob-
lem is entrusted to a genetic algorithm, which encodes the
solution as a permutation of the target bodies and uses
permutation-preserving crossover and mutation operators.
Specifically, two variants of the outer-level problem, with
different complexities, are considered in this paper, by
assuming either equal or different (in a set of discrete
values) time-lengths for the target-to-target transfers. In
the latter case, the permutation of the targets is efficiently
replaced by a permutation of the available rendezvous
epochs. The solution of the outer-level problem is given as
an input to the inner-level problem that is tackled through
EOS, an in-house multi-population self-adaptive Differen-
tial Evolution algorithm. As case studies, different ADR
missions are considered, including up to 20 targets in close
coplanar and circular orbits.

2. Problem Statement

Let us consider a set of $N$ prescribed target bodies that move
on circular coplanar orbits at slightly different altitudes
under a Keplerian dynamical model. For each target body
$i \in [1, N]$, orbital radius $r^{(i)}$ and initial anomaly $\theta^{(i)}(t_0)$ are
known. The velocity of the $i$-th target is constant and equal
to $v^{(i)} = \sqrt{\mu/r^{(i)}}$, while its angular position at any time $t$
is given by $\theta^{(i)}(t) = \theta^{(i)}(t_0) + \sqrt{\mu/(r^{(i)})^3}(t-t_0)$, where $\mu$
is the gravitational parameter of the central body. The cor-
positioning position $r^{(i)}(t)$ and velocity $v^{(i)}(t)$ vectors can be easily
derived. At the initial time, the chaser spacecraft is assumed
to be on a circular orbit of radius $r^{(0)}$ with anomaly $\theta^{(0)}(t_0)$
$= 0$, on the same orbital plane as all targets. The problem is
thus planar. The goal is to design a minimum-$\Delta V$ impulsive
trajectory that allows the chaser to rendezvous with all the $N$
vehicles within a specified mission time $T_M$.

Let us assume that the body encounter sequence $p = \{p_1$
$\ldots, p_N\}$ and the corresponding set of (monotonically
increasing) encounter times $t = \{t_1, t_2, \ldots, t_N\}$ are known,
where the integer $p_i \in [1, N]$ identifies the target met at time
$t_i$ and $t_1 < T_M$. The overall trajectory of the chaser can be
decomposed into $N$ target-to-target transfer legs. The $(k + 1)$
-th leg departs from body $p_k$ (with $p_0 = 0$ denoting the initial
state of the chaser) at time $t_k$ and arrives at body $p_{k+1}$ at time
$t_{k+1}$, for $k = 0, \ldots, N - 1$. The rendezvous condition requires
that, at time $t_k$, the position $r$ and velocity $v$ of the chaser are
the same as the encountered target:

\[
\begin{align*}
   r_k &= r^{(p_k)}(t_k), \quad k = 1, 2, \ldots, N, \quad (1) \\
   v_k &= v^{(p_k)}(t_k), \quad k = 1, 2, \ldots, N. \quad (2)
\end{align*}
\]

where subscript "$k$" is used to identify the chaser state at time
$t_k$.

With four being the maximum number of impulses required for an optimal time-fixed planar rendezvous [24],
each body-to-body transfer leg can be thought as a sequence of
three ballistic arcs, named A, B, and C, joined by impulsive
maneuvers, located at the departure point $(k)$, at two internal
points (with labels $k + 1/3$ and $k + 2/3$), and at the arrival
point $(k + 1)$, for the $(k + 1)$-th leg, respectively. A sketch of
this transfer leg is shown in Figure 1. A position formulation
is here considered; that is, the $(k + 1)$-th transfer leg is
parameterized with respect to radii $r_{k+1/3}$ and $r_{k+2/3}$
and anomalies $\theta_{k+1/3}$ and $\theta_{k+2/3}$ of the chaser at the internal
maneuvering points. Since the chaser state $r$, $v$ at the end-
points of any trajectory leg is fixed because of the rendezvous
conditions in Equations (1) and (2), the chaser position at
any impulse is completely defined. Conversely, the unknown
chaser velocity immediately before (superscript "-") and after
(superscript "+") the impulsive maneuvers can be found by
solving either a geometrical problem or a Lambert problem.

2.1. Geometrical Problem.

Consider the arc A between the
points $k$ and $k + 1/3$. As shown in Figure 2, for any given
semimajor axis $a > a_m$, two elliptical transfer orbits exist that
connect $r_k$ with $r_{k+1/3}$, where $a_m = \left( t_k + t_{k+1/3} + c \right)/4$
is the smallest possible semimajor axis and $c = \| r_k - r_{k+1/3} \|$ is the
chord distance between the two endpoints [41]. An ellipse
is known as the fast solution, since it is characterized by a
transfer time $\Delta t < t_m$; the second ellipse, instead, represents
the slow solution, since its transfer time is $\Delta t > t_m$
(See Figure 2(a)). As a consequence, the knowledge of the semi-
major axis $a$ is not sufficient to identify the elliptic arc connect-
ing the two points.

A nondimensional parameter $y \in [0, 1]$ is introduced,
which is implicitly defined by the relationship

\[
a = \frac{a_m}{4y(1 - y)}. \quad (3)
\]

Note that the same value of the semimajor axis $a$ is
obtained by using either $y$ or $1 - y$. So, by pairing the fast
solutions with values $y < 0.5$ and the slow solutions with
values $y > 0.5$, the elliptic arc connecting the two assigned
endpoints is uniquely identified for any choice of $y$.
Figure 2(b) depicts the geometric construction that allows
for an unambiguous definition of the transfer ellipse, that
is, of its semimajor axis $a$, eccentricity $e$, and argument of
peri center $\omega$, when $y_k$, $r_k$, $r_{k+1/3}$, and $\Delta \theta_{k+1/3}$ are known.
More precisely, the chord length

\[
c = \sqrt{r_k^2 + r_{k+1/3}^2 - 2r_k r_{k+1/3} \cos \Delta \theta_{k+1/3}} \quad (4)
\]

and the lengths $c_1 = 2a - r_k$ and $c_2 = 2a - r_{k+1/3}$ are evaluated
first. Angle $\gamma$ follows from

International Journal of Aerospace Engineering
\[
y = \cos^{-1}\left(\frac{r_k^2 - c_k^2 + c^2}{2r_k c}\right).
\]

Hence, \(y_i = y - y_o\), where

\[
y_c = \begin{cases} 
\cos\left(\frac{c_i^2 - c^2 + c_k^2}{2c_i c}\right), & \text{if } y \leq 0.5, \\
-\cos\left(\frac{c_i^2 - c^2 + c_k^2}{2c_i c}\right), & \text{if } y > 0.5.
\end{cases}
\]

The eccentricity can be found as

\[
e = \sqrt{c_i^2 + r_k^2 - 2c_i r_k \cos y_c}. \tag{7}
\]

Eventually, the initial true anomaly \(v_k\) and final true anomaly \(v_{k+1/3}\) along the transfer orbit are evaluated by means of the following relations:

\[
r_k = \frac{a(1 - e^2)}{1 + e \cos v_k}, \tag{8}
\]

\[
r_{k+1/3} = \frac{a(1 - e^2)}{1 + e \cos v_{k+1/3}},
\]

and ambiguities are easily solved as the angle \(\Delta \theta_{k+1/3} = v_{k+1/3} - v_k\) is known.

Velocities at both endpoints \((v_k^+\) and \(v_{k+1/3}^+)\) follow from the standard equations of the two-body problem. The transfer time \(\Delta t_A\) can be also evaluated through the Kepler equation, and consequently, the epoch of the intermediate maneuver \(t_{k+1/3} = t_k + \Delta t_A\) can be obtained.

The same procedure can be applied to the arc \(C\), which connects point \(k + 2/3\) to point \(k + 1\). If this geometrical construction is defined as a procedure \(y_{\text{Arc}}()\), one has as a result

\[
\begin{align*}
[y_{k+1/3}^-, y_{k+1/3}^+] & \leftarrow y_{\text{Arc}}(r_k, r_{k+1/3}, \Delta \theta_{k+1/3}, y_{k+1/3}^-), \\
[y_{k+2/3}^-, y_{k+2/3}^+] & \leftarrow y_{\text{Arc}}(r_{k+2/3}, r_{k+1}, \Delta \theta_{k+2/3}, y_{k+2/3}^-).
\end{align*} \tag{9}
\]

The cost of the maneuvers performed at the departure and arrival points of the \((k + 1)\)th leg are thus evaluated as

\[
\begin{align*}
\Delta V_{k+1} &= ||v_{k+1}^- - v_k||, \\
\Delta V_{k+2/3} &= ||v_{k+2/3}^- - v_{k+1}||. \tag{10}
\end{align*}
\]

### 2.2. Multirevolution 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 \(y_{k+1/3}\) and \(y_{k+2/3}\), the maneuvering epochs \(t_{k+1/3}^\pm\) and \(t_{k+2/3}^\pm\) and, consequently, the travel time along arc \(B\), \(\Delta t_B = t_{k+2/3} - t_{k+1/3}\), are assigned. Conversely, a multirevolution Lambert problem can be formulated, being the position vectors \(r_{k+1/3}^\pm\) and \(r_{k+2/3}^\pm\) and the travel time \(\Delta t_B\) known. This problem admits \(1 + 2n_{\text{max}}\) solutions, where \(n_{\text{max}}\) is the maximum possible number of revolutions: one solution for the 0-revolution transfer arc and two solutions, namely, left and right branch, for each \(n\)-revolution transfer orbit. Let us introduce an integer parameter \(L \in [-n_{\text{max}}, n_{\text{max}}]\), which indicates the solution corresponding to the \(|L|\)-revolution transfer orbit.
right branch for $L > 0$ and left branch otherwise. By knowing $L$, the velocity vector immediately after the second impulse $v_{k+1/3}^+$ and right before the third one $v_{k+2/3}^-$ can be evaluated according to the algorithm by Izzo [42], that is,

$$
[v_{k+1/3}^+, v_{k+2/3}^-] \leftarrow \text{Lambert}(r_{k+1/3}, r_{k+2/3}, t_{k+2/3} - t_{k+1/3} ; L).
$$

(11)

Hence, the total cost of the two internal maneuvers is

$$
\Delta V_k = ||v_{k+1/3}^+ - v_{k+1/3}^-|| + ||v_{k+2/3}^+ - v_{k+2/3}^-||.
$$

(12)

Instead of treating $L$ as an optimization variable, an enumeration approach is used; that is, all the $2n_{\text{max}} + 1$ possible solutions are computed and the one with the lowest total $\Delta V$ is chosen. When considering transfers between close orbits, an educated guess safely restricts 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 a total of six solutions (three right and three left branches) to be compared.

2.3. MINLP Formulation. According to the proposed problem formulation, the overall multi-target trajectory can be parameterized by using a set of $8N$ design variables. Hence, the solution vector is

$$
\mathbf{x} = \bigcup_{k=0}^{N-1} \mathbf{x}_k,
$$

(13)

where

$$
\mathbf{x}_k = \left\{ r_{k+1/3}, \Delta \theta_{k+1/3}, y_{k+1/3}, r_{k+2/3}, \Delta \theta_{k+2/3}, y_{k+2/3} \right\} \cup \{ p_{k+1}, t_{k+1} \}.
$$

(14)

Eventually, the impulsive MRR trajectory optimization problem can be formulated as

$$
P = \begin{cases} 
\min \Delta V_{\text{tot}}(\mathbf{x}) \\
\text{s.t.} \quad \mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U,
\end{cases}
$$

(15)

where the overall cost of the MRR trajectory is

$$
\Delta V_{\text{tot}} = \sum_{k=0}^{N-1} \left( \Delta V_{k+1/3} + \Delta V_{k+2/3} + \Delta V_{k+2} \right)
$$

(16)

and $\mathbf{x}_L$ and $\mathbf{x}_U$ 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-valued variables (radii and anomalies at the internal maneuvers, $y$ parameters, and encounter epochs). For this reason, problem (15) is an unconstrained Mixed-Integer Nonlinear Programming (MINLP) problem.
3. Two-Level Optimization Approach

The MINLP problem in Equation (15) belongs to the class of NP-hard problems [43]; hence, no deterministic algorithm is able to find the optimal solution in polynomial time. For this reason, a variety of stochastic metaheuristic techniques have been developed over the last decades with the aim of attaining (often suboptimal) good-quality solutions to the problem in a short time. However, as the problem dimension increases, also the computational time required to obtain a nearly optimal solution increases and may become prohibitive.

Instead of solving the problem as a whole, an alternative option is to decompose the MINLP into simpler subproblems, which could be, more or less easily, solved sequentially and whose solutions are eventually recomposed into a, hopefully near-optimal, solution to the original problem. For the problem at hand, a two-level approach can be pursued, by and whose solutions are eventually recomposed into a, hopefully near-optimal, solution to the problem in a short time. However, as the problem dimension increases, also the computational time required to obtain a nearly optimal solution increases and may become prohibitive.

The two layers are interconnected: the outer level requires a measure of the cost associated with each transfer leg to evaluate the quality of any encounter sequence, even though the actual $\Delta V$ of each leg can be evaluated only by solving the NLP problem referred to this transfer, that is, the inner-level problem. On the other hand, the inner level requires the definition of the encounter sequence and rendezvous epochs, which, in turn, are the output of the combinatorial, outer-level problem. In practice, the two problems could be solved sequentially provided that an alternative way, that is, a “heuristic”, exists for attaining a reasonable and, possibly, inexpensive estimation of any transfer cost during the solution of the outer-level problem, without the need of solving the complete NLP problem. Once the heuristic has been established, the outer-level combinatorial problem, or tour problem, is isolated and solved first; then, its solution is given as input to the inner-level problem, which returns the full-problem solution as output.

3.1. Cost Estimate for a Single Rendezvous Leg

This section presents an analytical, suboptimal, four-impulse transfer strategy that gives, in a short computational time, a conservative estimate of the optimal $\Delta V$ for a planar, time-fixed, impulsive rendezvous between circular orbits. When the allowed travel time is sufficiently large, this strategy fairly approximates the performance of the optimal solution [44]. Furthermore, with respect to other heuristics adopted in similar studies [45], the proposed approach represents a real, feasible, mission scheme.

Let $\theta_{1,0}$ and $\theta_{2,0}$ be the true anomaly at time $t = 0$ of the chaser and target, which fly on circular orbits of radius $r_1$ and $r_2$, respectively. The two orbits are coplanar and have angular velocities $\omega_1$ and $\omega_2$, respectively. Assuming that the two orbits are not extremely far apart, the minimum-$\Delta V$ solution is represented by a Hohmann transfer, of semimajor axis $a_{12} = \left( r_1 + r_2 \right)/2$ and duration $T_{H12} = \pi \sqrt{a_{12}^3/\mu}$. The transfer may be preceded by a coasting arc on the departure orbit, which allows the two bodies to achieve the correct relative phasing $\gamma = -\omega_2 T_{H12}$ required by this kind of maneuver. After the phase error $\Delta \theta = \theta_{2,0} - \theta_{1,0} - \gamma$ has been evaluated and normalized between 0 and $2\pi$, the time length $T_{\text{wait}}$ of the coast arc can be evaluated as

$$T_{\text{wait}} = \begin{cases} \frac{\Delta \theta}{\omega_1 - \omega_2}, & \text{if } r_1 < r_2, \\ \frac{2\pi - \Delta \theta}{\omega_2 - \omega_1}, & \text{if } r_2 < r_1. \end{cases}$$

(17)

By comparing the maximum available transfer time $T_{\text{max}}$ with the sum of the coasting time $T_{\text{wait}}$ and the time spent on the Hohmann transfer $T_{H12}$, it is possible to obtain a condition for the viability of the Hohmann transfer

$$T_{\text{wait}} + T_{H12} \leq T_{\text{max}}.$$  

(18)

Whenever Equation (18) is satisfied, the rendezvous has the cost $\Delta V_{H1}$ of the Hohmann transfer. Otherwise, the cost is evaluated according to the mission scheme depicted in Figure 3. The chaser spacecraft is injected into a circular (either internal or external) waiting orbit of radius $r_3$ with a Hohmann transfer “1-3,” of semimajor axis $a_{13} = \left( r_1 + r_3 \right)/2$ and duration $T_{H13} = \pi \sqrt{a_{13}^3/\mu}$. Once the chaser has reached the correct phasing with respect to the target body, a second Hohmann transfer “3-2,” of semimajor axis $a_{32} = \left( r_2 + r_3 \right)/2$ and duration $T_{H32} = \pi \sqrt{a_{32}^3/\mu}$, is performed to complete the rendezvous.

The problem now reduces to finding the radius $r_3$ that satisfies the rendezvous condition. The phase angle recovered by the chaser is

$$\Theta = 2\pi + \left( T_{\text{max}} - T_{H13} - T_{H32} \right) \omega_3 - T_{\text{max}} \omega_2,$$  

(19)

with $\Theta > 0$ for $r_3 < r_1$ and vice versa. If the initial phase angle $\Delta \theta_0 = \theta_{2,0} - \theta_{1,0}$ is normalized between 0 and $2\pi$, the equality between the chaser and target positions at the end of the maneuver requires

$$\Theta = \Delta \theta_0 - 2k_{\text{ext}} \pi,$$  

(20)

where $k_{\text{ext}} = 1$ for an external maneuver and $k_{\text{ext}} = 0$ when $r_3 < r_1$. The cheaper solution is selected.

An alternate 3-impulse heuristic outlines the rendezvous maneuver as an $n$-revolution bielliptic transfer split into two legs, each performing $1/2$ or $(n - 1)/2$ revolutions. In other words, the intermediate constant-radius arc of the 4-impulse heuristic is replaced by adding $(n - 1)$ complete revolutions to either ellipse. The radius of the intermediate impulse is selected in order to eventually achieve the
rendezvous with the target body. The angular length of the transfer is constrained at an integer number of revolutions. The time length is lower than the allocated time and the maneuver is completed by adding a coasting either on the initial or the final orbit. The 3-impulse heuristic better resembles the optimal rendezvous maneuver than the 4-impulse heuristic, but other features advice against its use.

3.2. Outer-Level Problem. Under the assumption that all bodies fly on coplanar circular orbits and that the allowed mission time is sufficiently large, the optimal transfer between any pair of bodies is a Hohmann transfer. The cost \( \Delta V \) of the transfer from body \( i \) to body \( j \) is independent of the specific departure and arrival epochs. The problem reduces to the search of the body sequence that minimizes the total velocity change.

Let \( p = (p_1, p_2, \ldots, p_N) \in \mathcal{P}^N \) be a permutation of the \( N \) integers \( \{1, 2, \ldots, N\} \), which identify the target bodies. So, the set \( \mathcal{P}^N \) is made up of all the \( N! \) possible permutations of \( N \) integers. The time-free optimization problem can be written as

\[
\min_{p \in \mathcal{P}^N} \sum_{k=1}^{N} \Delta V(p_{k-1}, p_k),
\]

where \( \Delta V(p_i, p_j) \) indicates the cost to move from target \( i \) to target \( j \), leaving at time \( t_{k-1} \) and arriving at time \( t_k \). Again, all transfer costs can be preliminarily evaluated and collected in a three-dimensional cost tensor \( \Delta V_{ijk} \) with dimension \((N + 1) \times N \times N\).

By extending the analogy between time-free tour and TSP, this novel combinatorial optimization problem is similar to a time-dependent version of TSP \([46, 47]\), named TDTS-A by the authors in a previous work \([38]\), where the cost of any city-to-city transfer also depends on its position in the whole tour.

3.2.2. Discrete Time-Length Tour. If the possible rendezvous epochs are discretized over a finer time grid, one obtains a combinatorial optimization problem with a solution closer to that of the original time-continuous problem and a formulation which is still close, to a certain extent, to that of the uniform time-length tour defined in the previous section.

Let \( \tau = \{\tau_1, \tau_2, \ldots, \tau_N\} \) be a time grid composed of \( N_t \) equidistant points, such that \( \tau_{k+1} - \tau_k = \Delta \tau = T_M/N_t \). Let \( \Delta V_i(j, k) \) denote the cost to move from body \( i \) to body \( j \), departing at time \( \tau_i \) and arriving at time \( \tau_k \). In this case, both the target sequence and the location of the rendezvous epochs over the grid are required for the computation of the tour overall cost. However, the problem can still be formulated using a single permutation \( \Pi \in \mathcal{P}^N \) of the integers \( \{1, 2, \ldots, N\} \), i.e., with a number of elements equal to the discrete number of available rendezvous epochs. In fact, the elements of \( \Pi \) with a value lower or equal to \( N \) constitute the target sequence \( p = (p_1, p_2, \ldots, p_N) \), while their position in \( \Pi \) reveals which elements of \( \tau \) make up the vector of rendezvous epochs \( \tau = \{\tau_1, \tau_2, \ldots, \tau_N\} \).
The proposed heuristic is reasonably accurate if the chaser has MINLP and hence more divisions make the problem too similar to the original one. It is advisable to keep the time division factor $D$ sufficiently small, because (i) a rough evaluation of the encounter epochs is sufficient, as the attainable epochs can be further refined in the inner-level optimization step; (ii) a large number of divisions make the problem too similar to the original MINLP and hence more difficult to solve; and (iii) the proposed heuristic is reasonably accurate if the chaser has enough time to perform several revolutions around the central body; so, reducing the minimum allowed travel time $\Delta t$ makes the heuristic less reliable and may undermine our efforts.

### 3.3. Inner-Level Problem

This step is aimed at improving the provisional trajectory provided by the solution of the outer-level problem, based on suboptimal maneuvers and rendezvous located at the nodes of a discrete time grid. Assuming that the optimal target sequence $p^* = \{p_1^*, \ldots, p_N^*\}$ and the discrete rendezvous epochs $\tilde{t} = \{\tilde{t}_1, \ldots, \tilde{t}_N\}$ have been determined, the MINLP problem in Equation (15) reduces to an unconstrained NLP problem, where the design variables are

$$x_{\text{NLP}} = \bigcup_{k=0}^{N-1} \left\{ r_{k+1/3}, \Delta\theta_{k+1/3}, y_{k+1/3}, r_{k+2/3}, \Delta\theta_{k+2/3}, y_{k+2/3} \right\} \cup \{a_{k+1}\},$$

with

$$t_k = \tilde{t}_k + \begin{cases} \alpha_k(\tilde{t}_k - \tilde{t}_{k-1}), & \text{if } \alpha_k < 0, \\ \alpha_k(\tilde{t}_{k+1} - \tilde{t}_k), & \text{if } \alpha_k \geq 0, \end{cases}$$

the continuous rendezvous epochs.

Two scenarios can be investigated: (i) time-fixed: the encounter epochs $t_k$ are kept fixed at their nominal values $\tilde{t}_k$ (so, $\alpha_k = 0$, $\forall k$), and (ii) time-free: the encounter epochs are free to be optimized. In the first case, each body-to-body transfer can be optimized independently from the others, thus reducing the $6N$-variable problem to solving $N$ easier subproblems, each one of just 6 variables. In the second case, the encounter epochs may vary in a neighborhood of the reference values, leading to an improved solution, but the entire trajectory must be optimized as a single $7N$-variable problem.

The lower and upper bounds of the design variables in Equation (26) are selected as follows:

$$r \in [0.95r_0, 1.05r_0],$$
$$\Delta\theta \in [0, 2\pi],$$
$$y \in [0.1, 0.9],$$
$$\alpha_k \in \begin{cases} [-1, 0.5], & \text{if } k = 1, \\ [-0.5, 0.5], & \text{if } 1 < k < N - 1, \\ [-0.5, 1], & \text{if } k = N - 1, \\ [0, 0], & \text{if } k = N. \end{cases}$$

The inner-level problem is tackled in this paper by EOS (Evolutionary Optimization at Sapienza) [48]. EOS is an in-house optimization code for continuous-variable problems, which implements an improved self-adaptive, partially restarted Differential Evolution (DE) algorithm, with a synchronous island model to handle multiple populations in parallel. EOS was developed in the contest of the Global Trajectory Optimization Competitions [49, 50] and has proven to be a flexible and high-performance algorithm, able to successfully cope with a broad range of real-world unconstrained and constrained space trajectory optimization problems [51, 52]. The values of the hyperparameters used in this work are reported in Table 1. Interested readers are suggested to refer to Ref. [48] for further details about the algorithm.
4. Genetic Algorithm

A genetic algorithm (GA) is used in this paper for solving the outer-level combinatorial optimization problems described in Section 3.2. Genetic algorithms are well-known population-based metaheuristic optimization techniques inspired by the process of natural selection, firstly introduced by Holland in 1960 [53]. 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 can find further details in Ref. [54]. A block diagram of the basic algorithm is proposed in Figure 5, highlighting the presence of three fundamental genetic operators (selection, crossover, and mutation) which form the backbone of any GA implementation.

An initial population, that is, a collection of \( n_p \) candidate solutions to the optimization problem (often referred as individuals or chromosomes), is randomly generated, aiming at covering the search space as uniformly as possible. Each individual is composed of a number of consecutive elements, called genes, which identify specific values of the design variables. At the end of each iteration (or generation), the fitness \( f \) (that is, the value of the function to optimize) of each individual in the current population is evaluated. Then, the next generation starts with the creation of a mating population, 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 Stochastic Tournament [54], the latter being the one used in the present application. Next, the individuals (or parents) in the mating population are randomly paired and their genes combined to produce new solutions (or offspring), according to a crossover operator. The underlying idea is that, by combining somehow good solutions, it is possible to create new solutions which hopefully are better than the previous ones. The process is repeated until a new offspring population (usually with the same dimension as the previous one) is created. Then, a few randomly chosen individuals of the offspring population undergo a mutation process, with the aim of increasing the population diversity. The mutation operator changes, in a random way, some genes of the chosen individuals. This new population eventually replaces the previous one, and the process is repeated until some termination criterion (e.g., a maximum number of generations \( n_G \)) has been met. As a minor, yet important, tweak, elitism is enforced: at the end of each generation, the best individual of the parent population is copied into the offspring population, replacing the worst individual. In this way, the best solution is prevented from being accidentally lost in the evolution process.

While selection operators are usually problem-independent, crossover and mutation operators are tightly related to the adopted encoding, that is, the way the real-world problem is described in terms of numerical variables (that is, genes). On the basis of the features of the decision variables, several encodings have been defined, with the most common being the binary, the integer-valued, and the real-valued encodings. In order to match the permutation-based formulations developed in Section 3.2, a permutation encoding is here considered; that is, each individual is a permutation \( p \in \Pi^L \), where \( L = N \) for the problems in Equations (21) and (22), while \( L = N_t \) for the problem in Equation (23). So, each gene takes an integer value in the range \( \{1, \ldots, L\} \), and two genes of the same individual cannot 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-encoded individual without producing unfeasible offspring, that is, child individuals whose genotype contains multiple copies of the same integer value. The same goes for classical mutation operators, such as random gene flip. Fortunately, an abundance of permutation-preserving genetic operators, specifically developed in the context of the TSP problem, is documented in the literature [55] and can be employed in solving the problem under investigation.
The Partially Matched Crossover (PMX) devised by Goldberg and Lingle in 1985 [56] has been adopted in the present paper, as it is aimed at preserving both the order and the position of as many elements as possible from the parent individuals. A pseudocode of this algorithm is presented in Algorithm 1. The first offspring \( c^{(1)} \) is created starting from a swath (or cut) \( s = \{p_a^{(1)}, \ldots, p_b^{(1)} \} \) of the first parent \( (p^{(1)}) \), in the same fashion as in a two-point crossover, being \( a \) and \( b \) two random integers in \( \{1, \ldots, L\} \), such that \( b > a \). The remaining entries are taken from the second parent \( (p^{(2)}) \). For each index \( i \notin \{a, b\} \), if the value \( v = p_i^{(2)} \) does not appear in swath \( s \), it is directly copied to \( c^{(1)} \) in position \( i \). Otherwise, conflicts are resolved by using \( p_i^{(1)} \) as a map for \( p^{(2)} \). So, PMX looks at position \( j \) of value \( v \) in \( p^{(1)} \); if \( p_j^{(1)} \) is not in swath \( s \), the value \( p_j^{(2)} \) is copied to position \( i \) in \( c^{(1)} \); otherwise, the procedure is repeated with \( v = p_j^{(2)} \). The second child \( c^{(2)} \) is then created by swapping the role of the two parents.

Mutation operators are also of great relevance in the framework of GAs, as they allow escaping from situations of premature stagnation of the population on a suboptimal solution; thus, their impact on the effectiveness of the algorithm should not be underestimated. Once again, an abundance of permutation-preserving mutation operators is documented in the literature. In the present application, the mutation operator is randomly selected at each generation between three possible rules: insert, swap, and reverse. All operators begin by randomly selecting two indices \( a \) and \( b \) in \( \{1, L\} \), with \( a < b \). For the insert operator, the value in the \( a \)-th position is transferred to the \( b \)-th position and the other genes of the individual are shifted accordingly; the swap operator exchanges the two genes in position \( a \) and \( b \); finally, the order of the genes from \( a \)-th to \( b \)-th is reverted when considering the reverse operator.

Some control over the population diversity is often needed to avoid an excessive uniformity between individuals during the optimization process, which would result in a very inefficient use of the crossover operator and, consequently, in a premature stagnation of the population in a local optimum.

To overcome this issue, an active diversity control strategy is implemented: the fitness of each individual is scaled by a penalty factor, which measures how many other individuals in the current population are similar to it. Let \( d_{ik} \) be the distance in the state space between the \( i \)-th individual \( p^{(i)} \) and the \( k \)-th individual \( p^{(k)} \). Specifically, the Hamming distance [57] is used to measure the difference between permutations, essentially counting the number of positions at which the corresponding elements are different between the two individuals:

\[
    d_{ik} = \sum_{j=1}^{L} \left[ p_j^{(i)} \neq p_j^{(k)} \right],
\]

where \( [x] \) is the Iverson bracket function, or indicator function, which returns 1 if \( x \) is true, 0 otherwise.

The augmented (i.e., penalized) fitness of the \( i \)-th individual is evaluated as

\[
    f_i^{\text{aug}} = f_i e^{\delta_i / \delta_k},
\]

where \( d_i = \sum_k d_{ik} \) is the diversity score of the \( i \)-th individual and \( \delta_k \) is a control radius.

In the presence of a large population, the evaluation of the diversity score of each individual might be too expensive (as it goes with \( O(n^2) \)). Hence, the full distance \( d_i \) is substituted by an approximation \( \tilde{d}_i \) evaluated over a number of \( n_{dc} \) individuals randomly selected in the population, that is,

\[
    \tilde{d}_i = \sum_{k=1}^{n_{dc}} d_{ik} r_k,
\]

where \( r_k \) is a random integer in \( \{1, n_p\} \setminus \{i\} \). In this work, the following values of the abovementioned hyperparameters showed the best overall results: \( n_{dc} = 100 \) and \( \delta_{dc} = 1 \).
4.1. GA Hyperparameters. The performance of a GA clearly depends not only on the choice of the selection, mutation, and crossover operators but also on the value of several hyperparameters, which drive the different phases of the evolution process. In addition to the population size \( n \) and the maximum number of generations \( n_{\text{gen}} \) that are common to all population-based metaheuristics, typical GA hyperparameters are the crossover probability \( p_c \), which represents the probability that the parents are replaced by the generated offspring at the end of the crossover phase; the mutation probability \( p_m \), which is the probability that a (child) individual undergoes a random mutation during the mutation phase; and other operator-specific parameters, such as the depth of the reverse mutation operator. A proper tuning of these hyperparameters may greatly improve the effectiveness of a GA. However, the optimal tuning of a GA is documented to be problem-dependent and represents an NP-hard problem by itself. Therefore, a preliminary trial-and-error tuning is usually done on a simplified, possibly scaled, version of the optimization problem of interest, with the hope of capturing all the relevant features of the original problem. The results of the hyperparameter tuning performed in this study are reported in Section 5.2.

5. Numerical Results

In this paper, numerical results are presented for different ADR missions aimed at removing all the debris composing a predetermined cluster with up to 20 objects orbiting in close coplanar LEOs. ID and initial orbital parameters assumed for both the chaser and the targets composing the cluster are listed in Table 2. When a \( N \times D \) tour is analyzed, only the first \( N \) debris of the cluster are considered, and the total mission time is set to \( T_{\text{max}} = 7NT_0 \), where \( T_0 \) is the initial orbital period of the chaser.

5.1. Analysis of the Cost Heuristic. The effectiveness of the suboptimal solution proposed as heuristic in Section 3.1 has been verified by comparison with the solution obtained by EOS in meaningful test cases. In particular, the rendezvous with a target moving on a circular orbit of radius \( r_2/r_1 = 1.02 \) is analyzed in this section.

Figure 6(a) shows the \( \Delta V \) estimated by the 4-impulse heuristic as a function of initial phase angle \( \Delta \theta_0 \in [0, 2\pi] \) and transfer time \( T_{\text{max}} \in [3T_1, 10T_1] \). \( \Delta V \) is estimated using either the 4-impulse (a) or 3-impulse heuristic (b).
Figure 7: Cost of the rendezvous with $r_2/r_1 = 1.02$, as a function of the available transfer time $T_{\text{max}} \in [3T_1, 7T_1]$. The transfer is either internal (a) or external (b).

Figure 8: Optimal and suboptimal chaser trajectories. Black lines show the constant-radius target trajectory. Empty squares mark initial and final positions. Red dots indicate impulsive maneuvers.
The cost in this class of mission is ruled more by the propellant needed to recover an incorrect initial phasing than by the basic $\Delta V$ required by the passage to a different radius. The outer-level combinatorial problem looks for and privileges target sequences that imply transfers starting with a favourable phase angle, i.e., in a narrow range close to values permitting a Hohmann transfer. Therefore, the initial time of each rendezvous is dictated by the solution of the outer-level problem. When the initial time (i.e., the initial phasing) does not permit a Hohmann transfer, the required $\Delta V$ is significantly dependent on the time which is available to complete the maneuver. An analysis of the rendezvous cost is better carried out by investigating the influence of the available time for significant assigned values of the initial phase angle. Figure 7 compares, from this perspective, the performance estimated by the two heuristics with the optimal transfer (as optimized by EOS). Even though the 3-impulse heuristic is closer to the features of the optimal solution, the 4-impulse heuristic provides a sufficient guess at the rendezvous cost. On the other hand, the trend of the 4-impulse heuristic solution is quite regular, without the abrupt changes that make the 3-impulse heuristic less apt to use when the rendezvous epochs are discretized over a coarse time grid. It is worthwhile to remark the quality of the EOS optimal solution:

![Figure 9: GA success rate as a function of the population size $n_p$, for assigned numbers of function evaluations FES.](image)

![Figure 10: GA success rate along the evolution process for the 12-target mission, with different crossover $p_c$ or mutation $p_m$ probabilities. The population size is $n_p = 64$.](image)
performance is never lower than the ones estimated by heuristics. In particular, the same performance is numerically achieved by EOS when either heuristic corresponds to the theoretically best strategy.

Figure 8 presents the chaser trajectory in a few significant cases. The circular orbit of the target is shown as a solid black line, starting from its initial position. The 4-impulse heuristic always uses the entire available time. On the contrary, the angular length of the 3-impulse heuristic is constrained to an integer number of revolutions, and the maneuver is completed by a coast arc, which is either initial or final. An internal maneuver is presented in Figure 8(a). The optimal solution does not present the constant-radius arc: nevertheless, by means of four impulses, it is able to make use of all the available time, thus improving performance. The features of the optimal maneuver are better highlighted by the three cases of increasing time length that are illustrated in Figures 8(b)–8(d) for an external maneuver (Δθ0 = 300 deg). The 3-impulse heuristic solution for $T_{\text{max}}/T_1 = 3.3$ has a very short final coasting and is very close to the optimal solution that, in this case, has also three impulses, but its second impulse is displaced from the apo-gee to allow using the entire angle range with a lower apo-gee. For greater values of $T_{\text{max}}$, the optimal rendezvous requires four impulses and gradually approaches the corresponding heuristic solution (Figure 8(c)) that, for just one value of available time, is itself optimal. The optimal maneuver hereafter moves away from the 4-impulse heuristic resembling an impracticable 3-impulse heuristic solution with one more lap. The latter solution can only occur when the available time permits the replacement of the long coast arc with an additional revolution and is, just at that moment, optimal (Figure 8(d)). These considerations explain the trend of the optimal solution in Figure 7 that periodically and alternately coincides with the 3- and 4-impulse heuristic solutions.

5.2. Tuning of the GA Hyperparameters. A preliminary tuning of the GA hyperparameters has been carried out by analyzing several instances of either the uniform or discrete time-length tour, keeping the problem size sufficiently small. This allows for an exhaustive search to be completed in a reasonable amount of time; thus, the optimal solution of the considered problems can be easily retrieved and used for comparison with the results of the GA. In order to mitigate the effects due to the intrinsic stochastic nature of the algorithm, 100 independent runs
were carried out for each set of hyperparameters under investigation.

The first analysis concerned the influence of the population size \( n_p \) and the number of generations \( n_G \) on the problem solution. The aim is to identify, if possible, a putative optimal allocation of trials, that is, a pair \( n_p \) and \( n_G \) that achieves the best performance for a known, limited, amount of function evaluations \( \text{FES} = n_G \cdot n_p \).

Figure 9 presents the success rate (percentage of runs that achieve the optimal solution) as a function of the population size \( n_p \), for a few threshold values of \( \text{FES} \), for the \( 8 \times 2 \) (Figure 9(a)) and \( 8 \times 3 \) (Figure 9(b)) missions. Here, the crossover and mutation probability are set as \( p_c = 0.9 \) and \( p_m = 0.15 \), respectively, and the diversity control is active with its default values. The success rate obviously increases when both the population size and the number of generations increase. However, when the number of available FES is limited, the results are not so easy to interpret. An even split between number of individuals and iterations, that is, \( n_p = n_G = \sqrt{\text{FES}} \), seems close to the optimal configuration. It is also worthwhile noting that, for a given number of FES, the success rate is higher in the \( 8 \times 3 \) mission than in the \( 8 \times 2 \) mission, even though the total number of variables is larger (24 vs. 16), and reaches 100\% with a population of 1024 individuals. The reason is probably related to the fact the the fitness landscape is smoother with a higher \( D \), and this favours the chosen genetic operators.

The effects of crossover and mutation probability on the success rate have been also investigated. Figure 10 reports the success rate for different values of \( p_c \) (Figure 10(a)) and \( p_m \) (Figure 10(b)) against the number of generations \( n_G \). A 12-target mission is considered for this test, using a population of \( n_p = 64 \) individuals; the diversity control is active with default values. According to Figure 10(a), the performance of GA seems to improve faster with \( n_G \) when small crossover probabilities are used (\( p_c = 0.5 \) and \( p_c = 0.7 \)), but, at the same time, tends to stall earlier, thus showing an apparent premature-convergence behavior. Conversely, higher crossover probabilities (\( p_c = 0.9 \) and \( p_c = 0.95 \)) lead to slower, yet constant, improvements. Thus, with \( n_G = 2^{15} \), the best results in terms of success rate are obtained with both \( p_c = 0.7 \) (96\%) and \( p_c = 0.9 \) (95\%). Figure 10(b) shows that when the number of generations is less than or equal to 500, the success rate is almost independent on \( p_m \). Only when a very high number of iterations are carried out, the effect of increasing the mutation probability becomes apparent. This is an expected behavior, as mutation is primarily devoted to avoiding stagnation over long time horizons.

Finally, Figure 11 shows the effect of the diversity control (dc) on the GA performance in both a uniform and a discrete time-length tour. In these plots, the solid-line curves represent the median fitness along the evolution process, while the shaded regions around them represent the interquartile range. It is possible to note that, in all reported cases, the use of diversity control leads to an improved solution, preventing GA from a premature stagnation. The beneficial effect of diversity control is more evident in problem instances with a larger number of targets \( N \) (Figure 11(a)), since, on average, in these cases, the GA is more likely to get trapped in a local minimum.

### Table 4: Solutions of the 15-target problem after EOS time-free optimization.

<table>
<thead>
<tr>
<th>( p_k ) ( 15 \times 1 )</th>
<th>( t_k ) (h)</th>
<th>( \Delta V_k ) (m/s)</th>
<th>( p_k ) ( 15 \times 2 )</th>
<th>( t_k ) (h)</th>
<th>( \Delta V_k ) (m/s)</th>
<th>( p_k ) ( 15 \times 3 )</th>
<th>( t_k ) (h)</th>
<th>( \Delta V_k ) (m/s)</th>
<th>( p_k ) ( 15 \times 4 )</th>
<th>( t_k ) (h)</th>
<th>( \Delta V_k ) (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5.67</td>
<td>21.65</td>
<td>6</td>
<td>5.61</td>
<td>21.65</td>
<td>11</td>
<td>2.61</td>
<td>26.81</td>
<td>11</td>
<td>8.11</td>
<td>26.81</td>
</tr>
<tr>
<td>7</td>
<td>17.75</td>
<td>47.22</td>
<td>7</td>
<td>18.37</td>
<td>42.04</td>
<td>10</td>
<td>21.71</td>
<td>10.69</td>
<td>10</td>
<td>21.70</td>
<td>10.69</td>
</tr>
<tr>
<td>2</td>
<td>36.86</td>
<td>93.19</td>
<td>2</td>
<td>40.39</td>
<td>87.75</td>
<td>14</td>
<td>44.90</td>
<td>83.07</td>
<td>14</td>
<td>45.39</td>
<td>81.98</td>
</tr>
<tr>
<td>1</td>
<td>47.79</td>
<td>12.54</td>
<td>1</td>
<td>51.63</td>
<td>15.66</td>
<td>4</td>
<td>47.01</td>
<td>75.30</td>
<td>4</td>
<td>46.78</td>
<td>75.30</td>
</tr>
<tr>
<td>10</td>
<td>56.66</td>
<td>70.60</td>
<td>10</td>
<td>56.20</td>
<td>70.60</td>
<td>3</td>
<td>51.88</td>
<td>19.76</td>
<td>3</td>
<td>51.61</td>
<td>19.31</td>
</tr>
<tr>
<td>9</td>
<td>77.24</td>
<td>25.51</td>
<td>9</td>
<td>76.01</td>
<td>24.83</td>
<td>2</td>
<td>76.15</td>
<td>54.56</td>
<td>2</td>
<td>75.76</td>
<td>54.49</td>
</tr>
<tr>
<td>4</td>
<td>79.33</td>
<td>43.31</td>
<td>4</td>
<td>79.33</td>
<td>43.31</td>
<td>8</td>
<td>79.33</td>
<td>54.37</td>
<td>8</td>
<td>78.25</td>
<td>54.37</td>
</tr>
<tr>
<td>3</td>
<td>95.22</td>
<td>32.07</td>
<td>3</td>
<td>93.50</td>
<td>35.84</td>
<td>15</td>
<td>90.61</td>
<td>42.66</td>
<td>15</td>
<td>94.16</td>
<td>42.66</td>
</tr>
<tr>
<td>14</td>
<td>96.49</td>
<td>80.77</td>
<td>14</td>
<td>96.86</td>
<td>80.77</td>
<td>9</td>
<td>107.97</td>
<td>37.29</td>
<td>9</td>
<td>108.54</td>
<td>37.29</td>
</tr>
<tr>
<td>8</td>
<td>107.07</td>
<td>37.37</td>
<td>8</td>
<td>105.13</td>
<td>37.37</td>
<td>6</td>
<td>120.68</td>
<td>32.41</td>
<td>6</td>
<td>120.42</td>
<td>33.68</td>
</tr>
<tr>
<td>12</td>
<td>130.04</td>
<td>26.75</td>
<td>12</td>
<td>130.04</td>
<td>26.75</td>
<td>1</td>
<td>123.31</td>
<td>32.83</td>
<td>1</td>
<td>121.49</td>
<td>32.83</td>
</tr>
<tr>
<td>13</td>
<td>136.48</td>
<td>11.25</td>
<td>13</td>
<td>136.48</td>
<td>11.24</td>
<td>5</td>
<td>136.25</td>
<td>31.72</td>
<td>5</td>
<td>136.25</td>
<td>31.34</td>
</tr>
<tr>
<td>5</td>
<td>151.89</td>
<td>64.54</td>
<td>5</td>
<td>151.57</td>
<td>64.54</td>
<td>12</td>
<td>138.02</td>
<td>59.23</td>
<td>12</td>
<td>137.59</td>
<td>59.23</td>
</tr>
<tr>
<td>11</td>
<td>159.91</td>
<td>53.90</td>
<td>11</td>
<td>164.71</td>
<td>53.90</td>
<td>13</td>
<td>143.41</td>
<td>5.32</td>
<td>13</td>
<td>140.75</td>
<td>5.32</td>
</tr>
<tr>
<td>15</td>
<td>170.00</td>
<td>21.24</td>
<td>15</td>
<td>170.00</td>
<td>21.24</td>
<td>7</td>
<td>170.00</td>
<td>51.80</td>
<td>7</td>
<td>170.00</td>
<td>51.55</td>
</tr>
</tbody>
</table>

641.91 637.49 617.82 616.85 1.26
unknown global optimum of the corresponding MINLP problem.

Table 3 presents the best solutions, in terms of $\Delta V_{\text{tot}}$, obtained by GA at the end of the outer-level optimization and by EOS after the inner-level optimization, with either fixed or free rendezvous times. For either mission scenario (i.e., for assigned values of number of targets and total transfer time), $\Delta V_{\text{tot}}$ decreases as the time-division factor $D$ increases, whereas the target sequence remains essentially unaltered, with just a few exceptions. The maneuvers of the GA solutions reflect the suboptimal 4-impulse heuristic and can be improved by EOS. From Table 4, it is possible to note that the differences between GA solutions and corresponding EOS time-fixed solutions are always in the order of few percent, which is the expected error of the devised heuristic, thus confirming its effectiveness and accuracy.

Table 5: Solutions of the 20-target problem after EOS time-free optimization.

<table>
<thead>
<tr>
<th>$p_k$</th>
<th>$t_k$ (h)</th>
<th>$\Delta V_k$ (m/s)</th>
<th>$p_k$</th>
<th>$t_k$ (h)</th>
<th>$\Delta V_k$ (m/s)</th>
<th>$p_k$</th>
<th>$t_k$ (h)</th>
<th>$\Delta V_k$ (m/s)</th>
<th>$p_k$</th>
<th>$t_k$ (h)</th>
<th>$\Delta V_k$ (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.83</td>
<td>54.48</td>
<td>1</td>
<td>3.80</td>
<td>54.48</td>
<td>1</td>
<td>3.72</td>
<td>54.48</td>
<td>1</td>
<td>3.64</td>
<td>54.48</td>
</tr>
<tr>
<td>2</td>
<td>16.63</td>
<td>21.81</td>
<td>2</td>
<td>16.63</td>
<td>17.71</td>
<td>2</td>
<td>16.62</td>
<td>17.73</td>
<td>2</td>
<td>16.63</td>
<td>17.74</td>
</tr>
<tr>
<td>4</td>
<td>39.33</td>
<td>16.45</td>
<td>4</td>
<td>28.91</td>
<td>16.43</td>
<td>4</td>
<td>32.01</td>
<td>16.45</td>
<td>4</td>
<td>28.33</td>
<td>16.43</td>
</tr>
<tr>
<td>3</td>
<td>46.53</td>
<td>5.43</td>
<td>3</td>
<td>46.35</td>
<td>5.47</td>
<td>3</td>
<td>43.08</td>
<td>5.43</td>
<td>3</td>
<td>49.34</td>
<td>5.47</td>
</tr>
<tr>
<td>12</td>
<td>51.64</td>
<td>70.18</td>
<td>12</td>
<td>50.68</td>
<td>70.15</td>
<td>12</td>
<td>50.71</td>
<td>70.18</td>
<td>12</td>
<td>50.37</td>
<td>70.15</td>
</tr>
<tr>
<td>13</td>
<td>68.00</td>
<td>52.22</td>
<td>13</td>
<td>73.67</td>
<td>36.10</td>
<td>13</td>
<td>75.99</td>
<td>33.28</td>
<td>13</td>
<td>75.14</td>
<td>33.62</td>
</tr>
<tr>
<td>6</td>
<td>77.12</td>
<td>59.09</td>
<td>6</td>
<td>77.12</td>
<td>59.10</td>
<td>6</td>
<td>78.33</td>
<td>59.09</td>
<td>6</td>
<td>77.24</td>
<td>59.10</td>
</tr>
<tr>
<td>5</td>
<td>91.58</td>
<td>32.75</td>
<td>5</td>
<td>89.85</td>
<td>36.38</td>
<td>5</td>
<td>91.13</td>
<td>37.28</td>
<td>5</td>
<td>91.45</td>
<td>34.06</td>
</tr>
<tr>
<td>9</td>
<td>99.31</td>
<td>37.88</td>
<td>9</td>
<td>103.40</td>
<td>72.38</td>
<td>9</td>
<td>105.07</td>
<td>61.80</td>
<td>9</td>
<td>105.07</td>
<td>62.39</td>
</tr>
<tr>
<td>16</td>
<td>107.39</td>
<td>47.84</td>
<td>16</td>
<td>105.53</td>
<td>47.84</td>
<td>15</td>
<td>109.55</td>
<td>37.28</td>
<td>15</td>
<td>107.67</td>
<td>37.29</td>
</tr>
<tr>
<td>10</td>
<td>116.06</td>
<td>42.48</td>
<td>15</td>
<td>118.93</td>
<td>23.33</td>
<td>16</td>
<td>121.42</td>
<td>32.22</td>
<td>16</td>
<td>121.14</td>
<td>26.60</td>
</tr>
<tr>
<td>7</td>
<td>136.00</td>
<td>26.94</td>
<td>10</td>
<td>124.67</td>
<td>31.93</td>
<td>7</td>
<td>130.72</td>
<td>69.42</td>
<td>7</td>
<td>132.21</td>
<td>69.40</td>
</tr>
<tr>
<td>14</td>
<td>146.35</td>
<td>53.58</td>
<td>7</td>
<td>139.25</td>
<td>26.92</td>
<td>10</td>
<td>135.74</td>
<td>26.94</td>
<td>10</td>
<td>134.96</td>
<td>26.92</td>
</tr>
<tr>
<td>18</td>
<td>153.74</td>
<td>31.62</td>
<td>14</td>
<td>143.80</td>
<td>53.56</td>
<td>14</td>
<td>152.24</td>
<td>27.01</td>
<td>14</td>
<td>152.34</td>
<td>26.73</td>
</tr>
<tr>
<td>17</td>
<td>175.67</td>
<td>16.15</td>
<td>18</td>
<td>153.23</td>
<td>31.59</td>
<td>18</td>
<td>154.09</td>
<td>31.62</td>
<td>18</td>
<td>154.08</td>
<td>31.59</td>
</tr>
<tr>
<td>11</td>
<td>180.62</td>
<td>42.33</td>
<td>17</td>
<td>178.50</td>
<td>13.57</td>
<td>17</td>
<td>178.69</td>
<td>14.04</td>
<td>17</td>
<td>179.70</td>
<td>13.22</td>
</tr>
<tr>
<td>8</td>
<td>194.57</td>
<td>21.43</td>
<td>11</td>
<td>181.29</td>
<td>42.30</td>
<td>11</td>
<td>180.54</td>
<td>42.33</td>
<td>11</td>
<td>182.49</td>
<td>42.30</td>
</tr>
<tr>
<td>20</td>
<td>199.47</td>
<td>84.59</td>
<td>8</td>
<td>194.39</td>
<td>21.42</td>
<td>8</td>
<td>187.00</td>
<td>21.43</td>
<td>8</td>
<td>183.40</td>
<td>21.42</td>
</tr>
<tr>
<td>19</td>
<td>220.90</td>
<td>5.21</td>
<td>19</td>
<td>199.35</td>
<td>79.40</td>
<td>19</td>
<td>199.00</td>
<td>79.38</td>
<td>19</td>
<td>199.24</td>
<td>79.40</td>
</tr>
<tr>
<td>15</td>
<td>226.66</td>
<td>36.75</td>
<td>20</td>
<td>226.66</td>
<td>5.20</td>
<td>20</td>
<td>226.66</td>
<td>5.20</td>
<td>20</td>
<td>226.66</td>
<td>5.20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\Delta V_{\text{tot}}/\Delta V_{\text{H}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>3.06</td>
</tr>
<tr>
<td>1.03</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
</tr>
</tbody>
</table>

Figure 12: Chaser radius $r$ as a function of nondimensional time $\tau_k/\Delta \tau$ for the $20 \times 3$ mission.
Starting from the GA solution with $D = 1$, any further improvement is due to the allocation of more time to the legs that cannot be executed according to a Hohmann transfer. This allocation can be achieved either by increasing $D$ or by using EOS in its time-free version. It is a curious coincidence that in both scenarios, the GA solution ($D = 4$) and the EOS time-free solution ($D = 1$) show approximately the same $\Delta V_{\text{tot}}$. The rendezvous maneuvers of the best GA solution are suboptimal, and the target sequence is slightly different from that of the uniform time-length tour. The application of EOS to this new sequence provides a further reduction of the required propellant.

The solutions for the 15- and 20- target scenarios, obtained after the time-free optimization, are reported in Tables 4 and 5, for each time-division factor $D$. Several details (i.e., optimal target sequence and, for each rendezvous, arrival time and required $\Delta V$) are presented. The last column, referring to the $N \times 4$ solution, compares the $\Delta V$ of each rendezvous with the cost $\Delta V_{\text{H}}$ of the optimal time-free transfer. The best trajectories are composed, for the greater part, of Hohmann transfers; 3- and 4-impulse legs are anyway flown by the chaser when the target is unfavourably phased during the available time window. This result is evident in Figure 12, which shows the radial position of the chaser during the whole $20 \times 3$ mission. The Hohmann transfer is usually exploited to move between distant orbits. In these cases, the phase angle changes more rapidly, and it is easier to find a favourable rendezvous opportunity.

The overall improvement with respect to the first GA solution, which assumes uniform time-lengths, is significant (close to 20%). The overall cost $\Delta V_{\text{tot}}$ of the best missions (see the last row in Tables 4 and 5) is $20 – 25\%$ greater than the value of the time-free trajectory having the same target sequence. These evidences let one argue that this two-step procedure is able to attain a solution very close to that of the original MINLP problem.

**6. Conclusion**

A two-level optimization procedure for the design of planar multirendezvous trajectories has been devised and tested. The goal is to minimize the overall propellant consumption, while completing, within a prescribed amount of time, the tour all the bodies in a prescribed set. The numerical solutions have been obtained for an active debris removal mission comprising up to 20 target bodies orbiting in coplanar LEOs.

The first, outer-level step concerns the definition of the optimal target sequence together with a preliminary evaluation of the rendezvous epochs. This step results in a permutation-optimization problem. Two different formulations, of increasing complexity, have been used, by supposing uniform time-lengths of the transfer legs or by selecting the rendezvous epochs in a discrete time grid. The growth in the problem complexity is rewarded by the attainment of a solution closer to the global optimum of the complete mixed-integer nonlinear programming problem. A genetic algorithm, with a peculiar permutation encoding and effective permutation-preserving genetic operators, is able to solve the combinatorial optimization problem. A simple, but fairly accurate, analytic solution of the single-target rendezvous maneuver is adopted as heuristic for a fast evaluation of the $\Delta V$ associated to each leg, without the need for an accurate optimization.

The solution attained at the end of the outer-level optimization procedure is further refined in the inner-level optimization step, by assuming the body sequence fixed and by optimizing the multi-impulse rendezvous maneuvers. Two formulations of increasing complexity are proposed even in this case. The first one maintains the rendezvous epochs found in the first step and only replaces each suboptimal trajectory described by the 4-impulse epochs with an improved maneuver. More precisely, each body-to-body transfer is described by means of a peculiar parameterization based on the position of the impulses: the total velocity change is minimized. The second formulation keeps the target sequence but can modify the rendezvous epochs to achieve a more appropriate sizing of the time allocated for each leg. The contemporary optimization of the entire trajectory is necessary.

Numerical results provided by this two-level procedure make the authors confident about the attainment of a trajectory which is very close to the solution of the full mixed-integer nonlinear programming problem. The overall computational effort is low enough to be compatible with desktop personal computers.

The two levels of the optimization procedure appeared of equal importance in the investigated scenarios. Instead, if the number of bodies and the average time available for a single rendezvous are large, the correct target phasing occurs frequently, and a solution based on many Hohmann transfers and very few 4-impulse low-$\Delta V$ maneuvers can be soon achieved by the outer-level optimization. The inner-level step becomes more important in problems with few targets and narrow time windows.

**Data Availability**

The data that support the findings of this study are contained in the article.

**Conflicts of Interest**

The authors declare that they have no conflicts of interest.

**References**


