Multi-impulse smooth trajectory design for long-range rendezvous with an orbiting target using multi-objective non-dominated sorting genetic algorithm

Sanaz Samsam[a], Robin Chhabra[a,*]

[a]*Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, K1S 5B6, Ontario, Canada*

Abstract

This paper develops a methodology to generate Pareto optimal trajectories for long-range rendezvous of a servicing satellite with a moving target, in an on-orbit servicing mission. The methodology employs a multi-impulse shape-based trajectory planning algorithm for in-plane orbit transfer, based on the two-body problem. We first derive the necessary and sufficient conditions that determine the set of smooth impulsive trajectories connecting the servicing satellite to the orbiting target. The Pareto optimal trajectories from this set are then obtained using a constrained multi-objective optimization algorithm developed based on the Non-dominated Sorting Genetic Algorithm-II (NSGA-II). In a mission, an optimal solution from the Pareto frontier set may be selected based on the mission requirements. Transfer time and control effort are the two Pareto cost functions that are considered in the multi-objective optimization. To reduce the risk of collision in populated orbits and to remain in an orbital regime, we include restrictions on

*Corresponding Author
Email address: robin.chhabra@carleton.ca (Robin Chhabra)

orbital elements as part of the constraints. Further, a maximum available impulse is considered as an upper-bound for velocity changes in an impulsive trajectory. The number of impulses along with the location of the first impulse in the parking orbit and the orbital parameters of the intermediate orbits form the set of design variables. The key advantage of the proposed trajectory optimization methodology compared to its counterparts using continuous thrust is the significant reduction of the number of design variables. Finally, we demonstrate the superiority of the developed trajectory planner by comparing its results with those obtained from another multi-objective evolutionary algorithm called the Multi-Objective Genetic Algorithm and an optimal Lambert approach based on single-objective optimization.
Keywords: On-orbit servicing, Long-range rendezvous, Dynamic target, Multi-impulse maneuver, Shape-based trajectory, Multi-objective optimization, Genetic algorithm

1. Introduction

On-Orbit Servicing (OOS) comprises a range of missions to provide the operating satellites with various services, such as, refueling, repairing, upgrading, orbit modification, assembly, and debris removal. Such operations intend to increase the satellites' lifetime and enhance their performance, as well as to reduce the operating costs associated with space programs. Examples of manned OOS missions are repairing the Hubble Space Telescope and assembling the International Space Station [1]. The world's first unmanned OOS mission was conducted by Engineering Test Satellite (ETS-VII), which was equipped with a deployable robotic arm, to demonstrate the capability

of docking to a target and the utilization of robotic technologies [2]. Other successful unmanned OOS missions have been performed. The most recent attempt that will be commissioned in 2022 is On-orbit Servicing, Assembly, and Manufacturing 1 (OSAM-1) that is a mission to extend satellites' lifespan even if they are not designed to be serviced on orbit [3]. An autonomous servicing satellite (SS) (hereafter called servicer) that is equipped with robotic arms can perform valuable services, e.g., visual inspection, refueling, and debris removal.

The highly increasing demand for OOS operations suggests the necessity to move towards industrialization, whose main challenge is the design of an efficient mission architecture for continuous supply of multiple services to a number of target satellites (hereafter called target). The phase of a servicing mission that mostly contributes to fuel and time consumption is the long-range rendezvous. Therefore, transfer trajectory optimization must be considered as one of the crucial steps in the industrialization process [4]. Various approaches for addressing the spacecraft trajectory optimization problem have been reviewed in [5, 6, 7]. In a survey by Shirazi *et al.* different aspects of the satellite transfer trajectory design are separately studied, including modeling, objective functions, and solutions [7]. In the following, the relevant research works conducted in these three aspects are surveyed in detail. Figure 1 is the visual illustration of different approaches to the trajectory optimization problem.

1.1. Model

In the design of transfer trajectories, two key characteristics of the spacecraft model must be considered: transfer type and equations of motion.

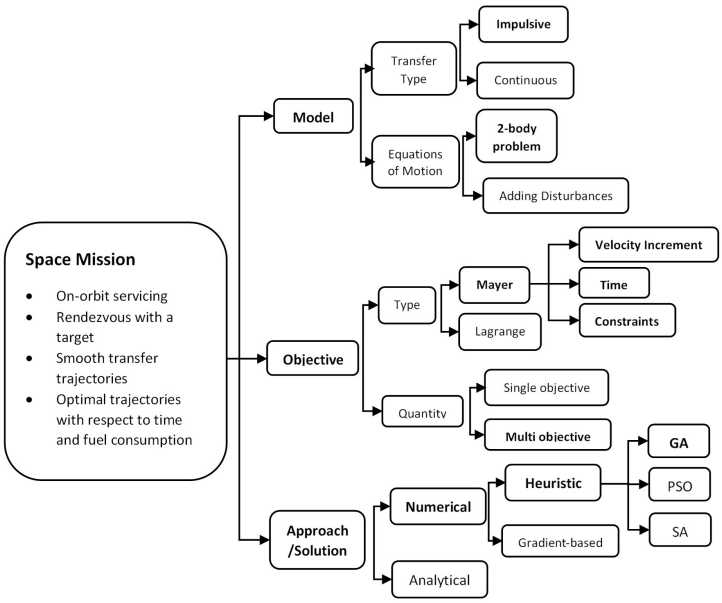

Figure 1: Spacecraft trajectory optimization process

Transfer type. Modelling the shape of the control input in an orbit transfer is an important step in trajectory optimization. Impulsive [8] and continuous [9] maneuvers are the two typical transfer types that enjoyed more attention in the literature. Although continuous transfer type captures a more accurate model of spacecraft motion, it adds to the complexity of the optimization problem, as the system always experiences non-zero inputs. On the other hand, the impulsive model reduces the computational effort, with an almost negligible compromise in the optimality of the objective function. Low specific impulse or high thrust level engines execute the impulsive maneuvers that generate conic arcs as the result of the velocity increment Δv at the impulse location [10]. Specifically when Δv is tangent to the path of motion, the generated trajectory is considered smooth. Of the impulsive ma-

neuvers, Hohmann transfer between two circular orbits is the most popular transfer method in which a two-impulse trajectory is generated. As another popular two-impulse transfer trajectory, one can refer to the Lambert transfer with the ability to generate a transfer orbit between any two arbitrary spatial points with a desire time interval. Improved versions of this approach have also been suggested in the literature, considering multiple orbital revolutions [11], orbital perturbations [12], and optimal Lambert solutions [13]. Beside the Lambert and Hohmann, shape-based approaches are other popular trajectory design methods for both impulsive and continuous maneuvers that work based on the trajectory's geometry. The distinguished feature of shape-based approaches is having fewer design variables comparing to their counterparts that do not consider constraints. Shakouri *et al.* propose a set of constraint equations to generate multi-impulse shape-based transfer trajectories between eccentric orbits in a central gravity [14]. A favorite geometry for designing continuous shape-based maneuvers is the use of spiral functions [15].

Equations of motion. Motion of the Earth's satellites is dominantly governed by the central gravity of the Earth (two-body problem). In addition, other factors, such as the gravity of other celestial objects [16], Earth oblateness [17], atmospheric drag [18], solar radiation pressure [19], etc., may be considered in the form of disturbing effects in the equations of motion. The magnitude of these disturbing forces changes with the satellite's orbital altitude. For example, the dominant disturbances in the Low Earth Orbital (LEO) regime are perturbations induced by Earth's oblateness (short periodic, long periodic, and mean change) and atmospheric drag [18]. Including

such effects in the equations of motion depends on the application [20]. For example, in a rendezvous mission in LEO, comprising the two phases of long-range and short-range, the effects of the long-periodic, mean changes, and atmospheric drag dominate those of the short periodic in the long-range rendezvous [21]. On the other hand, in the short-range rendezvous relative maneuvers with respect to target need be designed that require including relative equations of motion for both position and attitude dynamics [22].

1.2. Objective Function

Another step in spacecraft trajectory optimization process is defining objectives based on the mission requirements. Objective functions (or cost functions) may include fuel mass, total velocity increment, state errors, transfer time, or control effort. For a general spacecraft trajectory optimization, the objective function is formulated in two parts: (i) the Mayer term, which demonstrates the cost related to the final states, and (ii) the Lagrange term or the accumulated cost associated with the states and control efforts [23].

1.3. Approach and Solution

Solutions of the trajectory optimization problem are divided in two main categories: analytical and numerical approaches. Analytical approaches work based on the optimal control theory to determine a time history of control signals that minimizes a cost function, while satisfying a set of constraints [24]. Although the analytical approaches generate solutions with zero approximation, they are not necessarily attainable especially when the complexity of the model and the problem increases. As an alternative, numerical approaches are used to solve the satellite trajectory optimization problem, which are

divided in two categories of direct and indirect methods [25]. The former attempts to find the minimum cost function over the states and inputs of the system, and the latter numerically solves Pontryagin's minimum principle. Direct methods are more popular than indirect methods because of the ease of implementation, larger domain of convergence, and smaller problem size, indeed with some compromises in accuracy. Both methods attempt to minimize cost functions and constraint violations using discrete approximations based on some gradient- or heuristic-based approaches [26]. The high sensitivity of gradient-based methods to the initial guess of all system parameters renders them less desirable. Heuristic approaches offer alternative solutions to spacecraft trajectory optimization [13, 27], whose examples include the Genetic Algorithm (GA) [28, 29], the particle warm optimization [30], and the simulated annealing [29, 31].

Several methods exist to solve multi-objective optimization problems that can be divided in two general categories of decomposition and heuristic methods. In decomposition methods, a multi-objective optimization problem is converted to a single-objective problem [32, 33, 34, 35, 36]. Unlike heuristic methods, decomposition algorithms must be run several times to find a set of Pareto-optimal solutions. This makes heuristic methods faster and more reliable, comparing to decomposition ones [37]. Among all heuristic methods, Evolutionary Algorithms (EA), and more specifically GAs, are suitable to handle multi-objective optimization problems with constraints, especially for highly constrained problems in space applications [38] . A key feature of Multi-Objective EAs (MOEA) is maintaining a diverse set of solutions, since they work with a population of solutions. The Multi-Objective GA (MOGA),

Non-dominated Sorting GA (NSGA), and Niched-Pareto GA (NPGA) are the pioneering MOEAs that are able to find multiple Pareto-optimal solutions in one single run [32]. Although these algorithms are proved effective in generating multiple non-dominated solutions, they still lack elitism that is an index for better convergence of an MOEA. Other algorithms, such as Strength Pareto EA (SPEA), Pareto Archived Evolution Strategy (PAES), and Pareto Envelope-based Selection Algorithm (PESA) have been introduced as elitist MOEAs [32]. Deb *et al.* suggest a non-dominated sorting-based multi-objective EA, called NSGA-II, which has attracted attention in the research community [37]. They have developed NSGA-II to tackle the drawbacks of the NSGA, which include high computational complexity, lack of elitism, and the need for a sharing parameter to preserve diversity. This algorithm also alleviates issues with the SPEA, PAES, and PESA by converging to a Utopian Pareto frontier while maintaining diversity of solutions. Deb *et al.* also developed the NSGA-III, specialized for many-objective problems, which substitutes the crowding distance sorting algorithm in the NSGA-II with a reference-point-based algorithm. To further enhance its performance, NSGA-III is combined with a specific multiple-shooting discretization and tested in a highly constrained multi-objective trajectory optimization task [38]. Although both the NSGA-II & III have been proved effective, the NSGA-II is more efficient when handling 2 objective functions [39].

1.4. Statement of Contributions

This paper proposes a multi-impulse shape-based transfer trajectory generator for chasing a target in the long-range rendezvous phase of an OOS mission. The transfer trajectories are designed based on the two-body prob-

lem that captures the most dominant motion of the Earth's satellites. The benefit of including the target's orbital motion in the long-range rendezvous phase is to reduce the duration and fuel consumption in the short-range rendezvous. We develop a multi-objective constrained optimization architecture to minimize both the transfer time and control effort, using the concept of Pareto optimality. The proposed architecture is based on the constrained multi-objective NSGA-II with some modifications in sorting and mutation operators, for the sake of efficiency in our satellite trajectory design task. In addition, we implement the modified Inverted Generational Distance (IGD^+) as the convergence measure along with the modified NSGA-II. Smoothness of the generated trajectories and avoiding some highly populated orbital regions are among the constraints that are considered in this study. The former reduces the number of design variables and the computational complexity with an acceptable accuracy penalty, and the latter decreases the risk of collision with other orbiting objects. We consider the location and number of impulses in addition to some intermediate orbital elements as the design variables. Including the waiting time before applying the first impulse among the design variables considerably enhances the fuel consumption, with probably some compromise in transfer time. We investigate the efficacy of the developed optimal transfer trajectory generator in two case studies, where the results are compared with those obtained from MOGA [40] and an optimal Lambert approach. In the case studies, the convergence of the optimization, and the effects of number of impulses and waiting time in parking orbit are studied.

The remaining sections of this paper are organized as follows. The mathematical model describing a shape-based trajectory to chase a target is derived

in Section 2. The developed multi-objective optimization architecture is presented in Section 3. Section 4 reports some simulation results in two case studies. Finally, Section 5 includes some concluding remarks.

2. Multi-impulse Shape-based Trajectories for Target Chasing

In this section, we derive a set of constraint equations to generate multi-impulse smooth transfer trajectories for a servicer chasing a target in the long-range rendezvous phase. The motion of the servicer and the target is formulated based on the two-body problem, and smoothness requirement dictates that they must move in a plane. We model an impulse in an impulsive trajectory with an instantaneous velocity change. As the result, the proposed transfer trajectories are constructed by joining a number of co-planar co-focal elliptical arcs.

Motion of a satellite in a central gravity is fully described by the set of classical orbital elements denoted by the vector $\mathbf{q} = [a, e, \iota, \Omega, \omega, \nu]^T$. Here, a is the semi-major axis, e is the eccentricity, ι is the inclination, Ω is the Right Ascension of Ascending Node (RAAN), ω is the argument of perigee, and ν is the true anomaly. In a given plane in the three-dimensional space, identified by known ι and Ω, the satellite's motion is fully described by only four orbital elements, i.e., a, e, ω, and ν. Using the polar coordinates of the plane (r, θ), depicted in Figure 2, ν can be replaced by $\theta = \nu - \omega$ to introduce the vector $\mathbf{p} = [a, e, \theta, \omega]^T$ as the set of planar orbital elements. The parameters r, ν, θ, a, e, and ω are shown in Figure 2, where the origin is the Earth's center.

Conversely, knowing the satellite's position (r, θ) and its velocity $(\dot{r}, \dot{\theta})$ relative to the Earth-Centered Inertial (ECI) coordinate frame, we find the

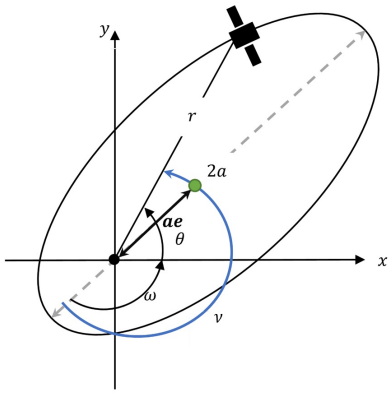

Figure 2: Oblique ellipse parameters in polar coordinate system

planar orbital elements by [10, p. 210]:

$$a = \frac{\mu r}{2\mu - r(\dot{r}^2 + r^2\dot{\theta}^2)}, \tag{1}$$

$$e = \sqrt{1 - \frac{r^4\dot{\theta}^2}{\mu a}}, \tag{2}$$

$$\omega = \cos^{-1}\{\frac{1}{e}[\frac{a}{r}(1 - e^2) - 1]\} - \theta, \tag{3}$$

where μ is the gravitational constant of the Earth (≈ 398601.2 km^{3}s^{-2}).

When an impulse takes place, the satellite's velocity changes instantaneously resulting in a sudden variation of the orbital elements and accordingly the satellite's trajectory. The velocity change at an intersection is denoted by $\Delta\mathbf{v}$. Given (1)-(3), we can observe how the radial ($\Delta\dot{r}$) and tangential ($r\Delta\dot{\theta}$) components of $\Delta\mathbf{v}$ vary the orbital elements of a satellite. A trajectory is called smooth, if the curves before and after an impulse intersect and share an identical tangent direction at the intersection point [41]. The smoothness condition at the intersection of any two sequential arcs can be formulated in

the polar coordinates as follows [14]:

$$r^- = r^+, \tag{4}$$

$$\theta^- = \theta^+, \tag{5}$$

$$(\frac{dr}{d\theta})^- = (\frac{dr}{d\theta})^+. \tag{6}$$

The + and − signs correspond to the conditions right after and right before an impulse, respectively. Using the relationship between r and the planar orbital elements [10, p. 182], the tangent direction of a trajectory is calculated by

$$\frac{dr}{d\theta} = \frac{e\sin(\theta+\omega)}{1+e\cos(\theta+\omega)}r, \tag{7}$$

$$r = \frac{a(1-e^2)}{1+e\cos(\theta+\omega)}. \tag{8}$$

The conditions in (4) and (5) imply that the two curves must have an intersection at the impulse location, and the polar coordinates of the satellite remain unchanged over the course of an impulse. Further, (6) demonstrates that at the location of the impulse the direction of the velocity vector does not change, and only its magnitude must vary. This condition implies that the two curves before and after an impulse must remain in a plane.

Problem 2.1 (Smoothness)**.** Given two locations in two co-planar orbits and an integer N, find the necessary and sufficient conditions for the smoothness of an N-impulse trajectory for a satellite joining these two locations.

For a general N-impulse smooth co-planar maneuver, $N-1$ arcs need to be designed to generate a transfer trajectory, each of which must satisfy (4)-(6) at the impulse locations. We denote the initial and final orbital locations

by $\mathbf{p}_1$ and $\mathbf{p}_{N+1}$, respectively, and we index the intermediate orbital elements by $i = 2, \ldots, N$. The vectors $\mathbf{p}_i = [a_i, e_i, \theta_i, \omega_i]^T$, $i = 1, \cdots, N+1$, are the set of planar orbital elements describing the full satellite trajectory starting from $\mathbf{p}_1$ and ending at $\mathbf{p}_{N+1}$. The parameter θ_i, for $i = 1, \cdots, N$, describes the location of the impulse where the satellite leaves the i^{th} orbit, and based on (5) it is equal to the satellite's polar angle immediately after the impulse. Note that $\theta_{N+1} = \theta_N$ that indicates the point of entry to the final orbit. To guarantee the smoothness of an N-impulse transfer trajectory from a known initial orbital location $\mathbf{p}_1$ to a given final orbital location $\mathbf{p}_{N+1}$, we impose the following set of $2N$ nonlinear equations based on (4)-(6), for $i = 1, \ldots, N$:

$$\begin{aligned} f_{i1}(\mathbf{p}_i, \mathbf{p}_{i+1}) := e_i \sin(\theta_i + \omega_i) + e_i e_{i+1} \sin(\omega_i - \omega_{i+1}) \\ - e_{i+1} \sin(\theta_i + \omega_{i+1}) = 0, \end{aligned} \tag{9}$$

$$\begin{aligned} f_{i2}(\mathbf{p}_i, \mathbf{p}_{i+1}) := a_i(1 - e_i^2)[1 + e_{i+1} \cos(\theta_i + \omega_{i+1})] \\ - a_{i+1}(1 - e_{i+1}^2)[1 + e_i \cos(\theta_i + \omega_i)] = 0. \end{aligned} \tag{10}$$

The number of unknowns in this set of equations is $4N - 5$ including the elements of vectors $\mathbf{p}_2$ to $\mathbf{p}_N$ minus θ_N that must be equal to the known θ_{N+1} based on (5).

Problem 2.2 (Chasing)**.** Given co-planar orbital locations of a servicer and a target at the time $t = 0$, find N-impulse smooth transfer trajectories for the servicer to intercept the target at its point of entry to the target's orbit.

The time-dependent orbital elements of the servicer in the parking orbit and the target are denoted by $\mathbf{p}_S(t) = [a_S, e_S, \theta_S(t), \omega_S]$ and $\mathbf{p}_T(t) =$

$[a_T, e_T, \theta_T(t), \omega_T]$, respectively. In this problem, knowing $\theta_S(0)$ and $\theta_T(0)$ we seek the necessary and sufficient conditions to generate N-impulse smooth transfer trajectories for the servicer, such that $\mathbf{p}_1$ is the location of the first impulse in the parking orbit and $\mathbf{p}_{N+1}$ is the point of entry of the servicer to the target's orbit intercepting the target. That is, we must impose the following condition in addition to the smoothness conditions in (9)-(10):

$$\mathbf{p}_{N+1} = \mathbf{p}_T(t_f), \tag{11}$$

where t_f indicates the servicer's arrival time. For an N-impulse trajectory, t_f depends on the intermediate orbital elements, and is calculated by

$$t_f(\mathbf{p}_1, \cdots, \mathbf{p}_N) = \sum_{i=1}^{N} \Delta t_i(\mathbf{p}_i, \theta_{i-1}), \tag{12}$$

$$\Delta t_i(\mathbf{p}_i, \theta_{i-1}) = \frac{M_i^- - M_{i-1}^+}{\sqrt{\frac{\mu}{a_i^3}}}, \quad i = 1, \cdots, N \tag{13}$$

where M_i^+ and M_i^- are the mean anomalies of the sevicer at the i^{th} impulse in the $(i+1)^{st}$ and i^{th} orbit, respectively. Note that M_0^+ is equal to $M_S(0)$, the servicer's initial mean anomaly in its parking orbit. To calculate the mean anomalies, we use the following set of equations ($i = 1, \cdots, N$) [10, p. 160]:

$$M_i^+ = E_i^+ - e_{i+1} \sin E_i^+, \tag{14}$$

$$E_i^+ = 2 \tan^{-1}\left(\sqrt{\frac{1 - e_{i+1}}{1 + e_{i+1}}} \tan \frac{\theta_i + \omega_{i+1}}{2}\right), \tag{15}$$

$$M_i^- = E_i^- - e_i \sin E_i^-, \tag{16}$$

$$E_i^- = 2 \tan^{-1}\left(\sqrt{\frac{1 - e_i}{1 + e_i}} \tan \frac{\theta_i + \omega_i}{2}\right). \tag{17}$$

The parameters E_i^+ and E_i^- are the eccentric anomalies of the servicer at the i^{th} impulse in the $(i+1)^{st}$ and the i^{th} orbit, respectively. At the time t_f the mean anomaly of the target satellite $M_T(t_f)$ is calculated by [10, p. 158]:

$$M_T(t_f) = \sqrt{\frac{\mu}{a_T^3}} t_f + M_T(0), \tag{18}$$

where $M_T(0)$ is the target's initial mean anomaly in its orbit. To ensure interception of the target, the condition on the final polar angles of the servicer and target in (11) can be substituted by

$$M_T(t_f) = M_N^+. \tag{19}$$

Note that $M_T(t_f)$ is a function of the number of impulses N, intermediate orbital elements $\mathbf{p}_i$, and the initial location of the servicer $\theta_S(0)$.

Comparing the chasing problem with smooth trajectory generation, the location of the initial and final impulses, θ_1 and $\theta_N = \theta_{N+1}$, are unknown; however, the condition in (19) must be added to (9) and (10) to ensure interception of the target. In summary, to generate an N-impulse smooth trajectory to chase a target, the following constraints should be satisfied:

$$f_{i1}(\mathbf{p}_i, \mathbf{p}_{i+1}) = 0, \quad i = 1, \cdots, N \tag{20}$$

$$f_{i2}(\mathbf{p}_i, \mathbf{p}_{i+1}) = 0, \quad i = 1, \cdots, N \tag{21}$$

$$f_3(\mathbf{p}_1, \cdots, \mathbf{p}_{N+1}) := (M_T(0) - M_N^+) + \sum_{i=1}^{N} (\frac{a_i}{a_T})^{\frac{3}{2}} (M_i^- - M_{i-1}^+) = 0. \tag{22}$$

The constraint equation in (22) is the result of substituting (12) and (18) in (19). In the problem of N-impulse smooth trajectory generation for target chasing, $2N+1$ constraints in (20)-(22) should be satisfied, while we have $4N-3$ unknowns, i.e., $\mathbf{p}_i\,(i = 2, \cdots, N)$ and θ_1. Including the location of

the first impulse among the unknowns allows us to study its effect on the long-range rendezvous maneuvers, which will be discussed in Section 4.

Some additional assumptions must be considered to find a unique solution to (20)-(22), since the number of unknowns is more than that of the equations for $N \geq 3$. We implement the Newton method in the current paper to numerically solve for an N-impulse smooth trajectory in chasing problems (see Algorithm 1). Once we have a solution, the magnitude of the velocity

Algorithm 1: Smooth trajectory generation for chasing problem.

Data: a_1, e_1, ω_1, a_T, e_T, ω_T, $\theta_S(0)$, $\theta_T(0)$, N, plus $2N-4$ unknowns

Result: Impulse locations and intermediate orbital elements: θ_1, $\mathbf{p}_i$ $(i = 2, \cdots, N)$

1 **for** $i \leftarrow 1$ **to** N **do**
  Form the system of nonlinear equations from (20)- (21);
  end
2 Add (22) to the system of equations, using (14)-(17).
3 Initialize the Newton algorithm using `ga()`.
4 Solve using Newton algorithm.

change at the i^{th} impulse location $\Delta v_i = \|\Delta \mathbf{v}_i\|$ is calculated based on the orbital elements of the smooth transfer trajectory by

$$\Delta v_i = v_i^+ - v_i^-, \tag{23}$$

where,

$$v_i^+ = \sqrt{\mu(\frac{2}{r_i} - \frac{1}{a_{i+1}})} \quad \text{and} \quad v_i^- = \sqrt{\mu(\frac{2}{r_i} - \frac{1}{a_i})}. \tag{24}$$

The velocity magnitude before and after the i^{th} impulse are denoted by v_i^- and v_i^+, respectively. The radius r_i is the radial component of the location of

the servicer at the i^{th} impulse in polar coordinates, based on (8). Note that at the i^{th} impulse location, the direction of $\Delta\mathbf{v}_i$ is the same as the servicer's velocity direction due to the smoothness conditions.

3. Trajectory Design Optimization

For $N \geq 3$, since we have $2N-4$ unknowns more than the number of equations in (20)-(22), there exist infinitely many smooth trajectories for chasing a target. Each set of $2N-4$ unknowns results in a feasible transfer trajectory with a specific control effort and transfer time. In this section, we implement a multi-objective non-dominated sorting genetic algorithm to find the set of Pareto optimal smooth transfer trajectories. Employing optimization helps enhance the performance of the derived multi-impulse smooth trajectories for chasing a target in the long-range rendezvous phase of OOS.

Problem 3.1 (Optimality)**.** Given the control effort and transfer time as Pareto cost functions, and the number of impulses N along with the set of $2N-4$ unknowns as design variables, find the set of Pareto-optimal trajectories satisfying (20)-(22). The optimization must also include constraints on the orbital elements of the transfer trajectories.

The remainder of this section details the employed constrained multi-objective optimization, including the description of design variables, Pareto cost functions, and constraints, to find the optimal transfer trajectories.

3.1. Design variables

Since N is a part of design variables and its change linearly affects the total number of design variables, a maximum number of impulses N_{max} is

fixed. This introduces a total number of $2N_{max} - 3$ design variables in the trajectory design optimization. When $N = 2$, there is a unique solution to (20)-(22). If $3 \leq N \leq N_{max}$, a subset of $2N - 4$ design variables identifies the solution to (20)-(22). The collection of all design variables is denoted by the vector $\mathbf{x} \in \mathbb{R}^{2N_{max}-3}$. For an N, we have the freedom to choose the design variables from the vectors $\mathbf{p}_i$, $i = 1, \cdots, N$. To capture the required unknowns to solve (20)-(22) for every value of $3 \leq N \leq N_{max}$, we have to cautiously choose the design variables to include enough information. Any two variables from the components of $\mathbf{p}_2$ and $\mathbf{p}_3$, and θ_1 (9 variables in total) can be chosen to be two of the design variables for $N_{max} = 3$, e.g., the vectors $\mathbf{x} = [N, \theta_1, \omega_2]$ or $\mathbf{x} = [N, a_2, e_3]$ are acceptable choices. However, some parameters are preferred over the others, e.g., since θ_i and ω_i appear in the argument of trigonometric functions they are more reasonable choices comparing to a_i and e_i. As it is evident in (9)-(10), knowing θ_1 and ω_2 for $N = 3$ gives explicit answers for a_2 and e_2, without needing a nonlinear solver. In the case of $N_{max} = 4$, we keep the first two design variables as the ones chosen for $N_{max} = 3$ and add two more design variables from $\mathbf{p}_2$, $\mathbf{p}_3$, and $\mathbf{p}_4$, i.e., the vectors $\mathbf{x} = [N, \theta_1, \omega_2, \omega_3, \omega_4]$ or $\mathbf{x} = [N, \theta_1, \omega_2, e_3, e_4]$ can serve as the vector of design variables. But as discussed, the former choice is preferred. This process can be extended to any maximum number of impulses considered in the optimization. Note that in any design candidate $\mathbf{x}$ if $N < N_{max}$, only a subset of design variables identifies the cost functions.

3.2. Constraints

In every real-world optimization problem, certain physical or manufacturing requirements ought to be satisfied, leading to a set of constraints on

the design variables. In the problem of trajectory design optimization, four different types of constraints are considered

1. To meet the smoothness and chasing requirements of the N-impulse transfer trajectories, (20)-(22) are the first set of constraints.
2. A belt of orbits is defined to be a set of neighbouring elliptic orbits of the Earth. Such a belt can be identified by the following inequalities on a, e, and ω:

$$a_L < a < a_U \text{ and } e_L < e < e_U \text{ and } \omega_L < \omega < \omega_U,$$

   where the subscripts L and U indicate the lower and upper bound for an orbital element, respectively. Therefore, to avoid a certain belt of orbits, a set of inequalities are imposed on the parameters of the intermediate orbital elements ($i = 2, \cdots, N$):

$$a_i \geq a_U \text{ or } a_i \leq a_L, \text{ or } e_i \geq e_U \text{ or } e_i \leq e_L, \text{ or } \omega_i \geq \omega_U \text{ or } \omega_i \leq \omega_L. \tag{25}$$

   A belt of orbits may represent a populated orbital region where the risk of collision with other spatial objects is high. Note that avoiding a belt of orbits guarantees that the servicer never uses any orbit in the belt for transfer; however, it may cross the belt in its path to the target.
3. To ensure that the servicer stays within an orbital region, constraints may also be applied on functions of the intermediate orbital elements. For example, to always stay within LEO throughout a long-range rendezvous with a target, the following inequalities must be satisfied to ensure that the maximum and minimum altitude of the servicer re-

main between 100-2000 kilometers. For $i = 2, \cdots, N$:

$$a_i(1 + e_i) \leq 8371\text{km and } a_i(1 - e_i) \geq 6471\text{km}. \tag{26}$$

4. The maximum amount of available Δv at each impulse is subject to some limitations. Defining Δv_U as the maximum available velocity increment at each impulse, the following set of inequalities must be satisfied for $i = 1, \cdots, N$ to find a feasible transfer trajectory.

$$\Delta v_i \leq \Delta v_U. \tag{27}$$

3.3. Cost Functions

Both total Transfer Time (TT) and Control Effort (CE) are critical performance indexes in the servicer trajectory planning problem. For any impulsive trajectory design in the two-body problem context, TT and CE can be respectively computed by the following functions:

$$J_t = t_f(\mathbf{p}_1, \cdots, \mathbf{p}_N), \tag{28}$$

$$J_c = \sum_{i=1}^{N} |\Delta v_i(\mathbf{p}_i, \mathbf{p}_{i+1})|. \tag{29}$$

The functions t_f and Δv_i are already defined in (12) and (23), respectively.

3.4. Optimization problem

Let J denote the collection of the objective functions J_t and J_c, i.e.,

$$J = [J_t, J_c]^T \ . \tag{30}$$

The optimality problem in Problem 3.1 can now be formulated as the following multi-objective constrained optimization:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} J(\mathbf{x}) \tag{31}$$

$$\text{subject to (20)-(22) and (25)-(27).}$$

Here, $\mathbf{x}^*$ denotes a member of the set of optimal solutions. The set of equality constraints in (20)-(22) are already taken into consideration when defining the design variables as a subset of parameters appearing in $\mathbf{p}_2$ to $\mathbf{p}_{N_{max}}$ along with θ_1 and N.

Let $g_j(\mathbf{x})$ for $j = 1, \cdots, m$ be a set of functions of the design variables. A constrained optimization problem with a set of inequality constraints in the form of $g_j(\mathbf{x}) \leq 0$ $(j = 1, \cdots, m)$ can be converted into an unconstrained problem. Note that the constraints in (25)-(27) can be reformatted to this form. Let Φ be a cost function defined as:

$$\Phi(\mathbf{x}; \alpha_t, \alpha_c) = J(\mathbf{x}) + [\alpha_t, \alpha_c]^T \sum_{j=1}^{m} G_j(\mathbf{x}),$$

where $G_j(\mathbf{x}) = \max\{0, g_j(\mathbf{x})\}$ and $\alpha_t, \alpha_c \gg 1$ are two penalty constants. The following unconstrained multi-objective optimization problem is equivalent to (31):

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \Phi(\mathbf{x}; \alpha_t, \alpha_c). \tag{32}$$

In a multi-objective optimization, if two objectives are cooperative, i.e., changing design variables has the same effect on both, the optimal solution is a single point. However, if we deal with two competitive objectives, we have multiple optimal solutions whose collection is called the Pareto frontier set. In the trajectory optimization problem in (32), the Pareto cost functions J_t and J_c are normally competitive and the solutions to (32) form the Pareto frontier set. A solution $\mathbf{x}^*$ in the Pareto frontier set dominates every other feasible solution to (32) in the space of objectives. Point A dominates point

B in the Pareto sense, if A is better than B in at least one objective function and not worse with respect to all other objective functions. Then, Pareto frontier set is the collection of all of the solutions that cannot dominate one another [42], i.e., no solution is superior to others.

3.5. Non-dominated Sorting Genetic Algorithm

Non-dominated Sorting Genetic Algorithm (NSGA-II) is a guided random search multi-objective optimization algorithm with the capability of exploring the diverse regions of the solution space. This algorithm proposes a direct approach using the concept of Pareto-optimality for solving multi-objective problems with four main advantages to its direct rival multi-objective optimization algorithms: (i) it uses a fast non-dominated sorting procedure, (ii) it is an elitist-preserving approach, (iii) it uses an operator for exploring diverse regions, and (iv) it provides strategies for handling constraints [37]. The flowchart in Figure 3 demonstrates the implementation of NSGA-II to find the optimal transfer trajectories from (32). The steps of NSGA-II for finding the optimal answers are elaborated in the following.

3.5.1. Initialization

Considering N_{max} number of impulses, $2N_{max}-3$ design variables must be initialized for K number of population. This initialization can be performed whether randomly or by a specific pattern. For example, if there are some "good" solutions available, they can be included in the initial population. The original NSGA-II sorts the population after the initialization. However, we perform sorting after producing the children to save some computations.

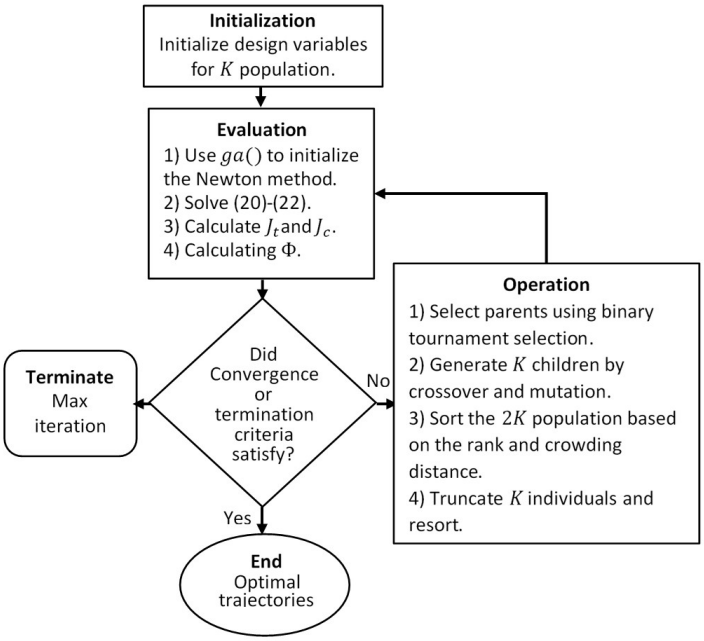

Figure 3: Multi-objective optimization architecture

3.5.2. Cost Function evaluation

To evaluate the cost functions, the nonlinear solver in Algorithm 1 calls the initialized design variables to find K smooth transfer trajectories that chase a dynamic target by $N \leq N_{max}$ number of impulses based on (20)-(22). The Newton method as the nonlinear solver has a fast convergence near the optimal point, but it is too sensitive to the initialization of the parameters computed by this solver. To address this drawback, the ga() function from MATLAB with a low number of iterations is used to find a set of initial parameters for which the functions f_{i1}, f_{i2}, and f_3 $(i = 1, \cdots, N)$ are close enough to zero. The transfer time and control effort are calculated based on (12) and (23) after computing the feasible intermediate orbital elements from the Newton solver. Moreover, those solutions which do not satisfy the

constraints in (25)-(27) are penalized using α_t and α_c in the cost function Φ to have less chance to be part of the next generation.

3.5.3. Sorting Operators

The NSGA-II uses the rank and crowding distance operations in the space of objectives including CE and TT to sort the individuals in a generation [37]. For each individual $\mathbf{x}_k$ $(k = 1, \cdots, K)$ in a generation, the rank operator is evaluated based on the following algorithm. Two entities are calculated: (i) domination count $n_{\mathbf{x}_k}$ that is the number of individuals dominating $\mathbf{x}_k$, and (ii) a set of individuals $S_{\mathbf{x}_k}$ that $\mathbf{x}_k$ dominates. Initially, $n_{\mathbf{x}_k} = 0$ and $S_{\mathbf{x}_k} = \emptyset$. Then, we form $S_{\mathbf{x}_k}$ and compute $n_{\mathbf{x}_k}$ by comparing all individuals in the generation. If $n_{\mathbf{x}_k} = 0$, the rank of $\mathbf{x}_k$, denoted by $\text{rank}(\mathbf{x}_k)$, is assigned to be 1. For every $\mathbf{x}_k$ with rank 1, we reduce the domination count of every element in the set $S_{\mathbf{x}_k}$ by 1. Excluding the rank 1 individuals, if the recomputed $n_{\mathbf{x}_k} = 0$, then $\text{rank}(\mathbf{x}_k) = 2$. Repeating the same procedure, we rank all of the individuals in the generation. We denote the set of all individuals with rank l by PF_l. Clearly, the lower the rank, the closer the individuals are to the Pareto-optimal solutions. The algorithmic evaluation of the rank operator is detailed in Appendix A.

The Crowding Distance (CD) operator guides the algorithm at each stage towards a uniform Pareto frontier set, and helps generate diverse solutions. In a generation, the CD operator is evaluated for an individual $\mathbf{x}_k$, based on the following algorithm. The two extreme individuals in PF_l are identified:

$$\mathbf{x}_t^l = \arg \max_{\mathbf{x}_k \in \text{PF}_l} J_t(\mathbf{x}) \quad \& \quad \mathbf{x}_c^l = \arg \max_{\mathbf{x}_k \in \text{PF}_l} J_c(\mathbf{x}).$$

For the extreme individuals $\mathbf{x}_t^l$ and $\mathbf{x}_c^l$ with rank l, we define the CD to

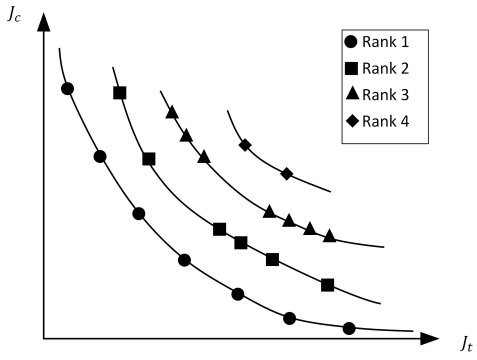

Figure 4: Non-dominated sorting ranking.

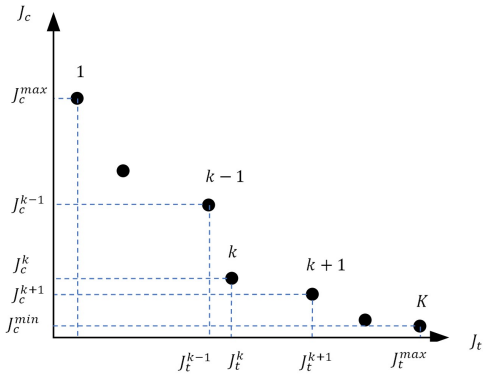

Figure 5: Crowding distance for two objectives.

be infinity. We sort the members of PF_l based on the value of one of the objective functions. Then, for the individuals in PF_l except the extremes we define the CD to be

$$\mathrm{CD}(\mathbf{x}_k) = \frac{J_c(\mathbf{x}_{k+1}) - J_c(\mathbf{x}_{k-1})}{J_c(\mathbf{x}_c^l) - J_c(\mathbf{x}_t^l)} + \frac{J_t(\mathbf{x}_{k+1}) - J_t(\mathbf{x}_{k-1})}{J_t(\mathbf{x}_t^l) - J_t(\mathbf{x}_c^l)}. \tag{33}$$

Note that $\mathbf{x}_{k-1}$, $\mathbf{x}_k$, and $\mathbf{x}_{k+1}$ are three consecutive individuals in the set PF_l. The higher the CD, the more preferred individual is in PF_l, since it identifies rare regions in the search space. Figure 5 demonstrates the crowding distance in a set PF_l, and the algorithmic evaluation of the CD operator is detailed in Appendix A.

3.5.4. Binary Tournament Selection

In any step of the optimization that we need to select parents, we use binary tournament selection method. In this method, two individuals are randomly chosen from the population and the one with the lower rank is kept in the process. If the two individuals have the same rank, then the one

with higher CD is preferred to become the parent. To select other parents, we repeat the same procedure [43].

3.5.5. Arithmetic Crossover

Crossover is an inheritance operator, that causes the children to have attributes from all parents. Initially, a crossover rate (between 0 and 1) is fixed that is the indicator of the execution probability of this operator. During the optimization, each time that the crossover operator is called, a random number between 0 and 1 is generated. If this number is smaller than the crossover rate, then this operator is executed. During crossover, a random $2N_{max}-3$ dimensional vector $\boldsymbol{\beta}$ whose elements are between 0 and 1 is generated, based on which we define the children $\mathbf{x}_1^c, \mathbf{x}_2^c$ of the two selected parents by the binary tournament selection, $\mathbf{x}_1^p, \mathbf{x}_2^p$.

$$\mathbf{x}_1^c = \boldsymbol{\beta} * \mathbf{x}_1^p + (\mathbf{1} - \boldsymbol{\beta}) * \mathbf{x}_2^p, \quad \& \quad \mathbf{x}_2^c = \boldsymbol{\beta} * \mathbf{x}_2^p + (\mathbf{1} - \boldsymbol{\beta}) * \mathbf{x}_1^p,$$

where the operator $*$ refers to the element-wise multiplication of two vectors and the vector $\mathbf{1}$ is the $2N_{max}-3$ dimensional vector whose elements are all equal to 1 [43].

3.5.6. Mutation

In the iteration $1 \leq d \leq D$, where D is the maximum number of iterations, a number of individuals are randomly selected based on the binary tournament selection to be mutated with a probability, to explore the search region. A mutation probability distribution is fixed at the initialization phase of the NSGA-II algorithm. This distribution allows the mutation operator to

execute with a higher probability at the beginning of the optimization, which leads to avoiding convergence to local optima. In this paper, the mutation probability PM in the iteration d is defined by (see Figure 6)

$$\mathrm{PM}(d) = \mathrm{PM}_0(1 + \cos\frac{\pi d}{D}),$$

where PM_0 is a parameter, fixed at the beginning of the optimization loop. Each time that the mutation operator is called for an individual, $2N_{max} - 3$ random numbers between 0 and 1 are generated. If any of these numbers is smaller than $\mathrm{PM}(d)$, then the corresponding design variable in the individual is randomly reassigned within its bounds [43].

3.5.7. Replacement

After applying the crossover and mutation operators on the current population, K children are generated and added to the current population. This new population of $2K$ individuals is sorted first based on their rank and then their crowding distance. To maintain a fix number for the population in each generation, K best individuals are truncated. Note that the new K individuals must be sorted again, since the sorting operators depend on all members of a population. This new K individuals are the population for the next generation (iteration) [37].

3.5.8. Termination Criteria and Convergence

In an optimization, several termination criteria can be considered: (i) maximum number of iterations, (ii) maximum number of iterations with no improvement, (iii) maximum allowed CPU time, and (iv) reaching an admissible fitness. For multi-objective problems, another criterion that is widely

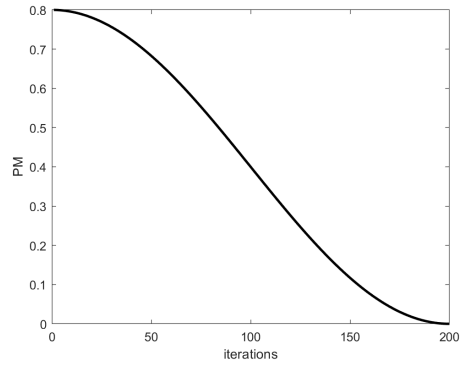

Figure 6: Mutation probability

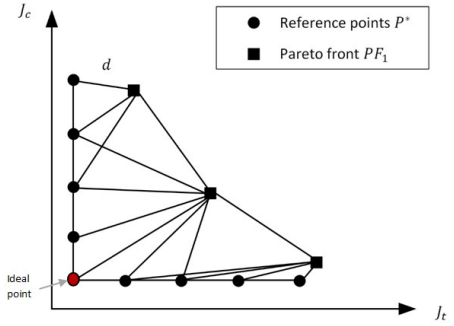

Figure 7: Inverted Generational Distance

used to measure both convergence and diversity is Inverted Generational Distance (IGD) [44]. This performance metric is defined in the following. Let PF* denote the ideal Pareto frontier, defined as the one-dimensional hyperplane in the performance space according to the minimum values of J_t and J_c and the constraints over them. If no constraints are specified for the objective functions, the PF* includes two lines parallel to the objective axes passing through the minimum values of J_t and J_c. The two lines meet at a point closest to the origin, coined as the ideal point. Considering a set of reference points P^*, i.e., uniformly distributed points along the ideal Pareto frontier, and PF_1 as the set of non-dominated solutions generated by the optimization algorithm, the modified Inverted Generational Distance (IGD$^+$) is defined as:

$$\mathrm{IGD}^+ = \frac{\sum_{P \in P^*} dist(P, \mathrm{PF}_1)}{|P^*|}.$$

Here, $dist(P, \mathrm{PF}_1)$ is the minimum Euclidean distance $d(P, Y)$, between $P \in P^*$ and the point Y in PF_1, and $|P^*|$ is the cardinality of the set P^*. The

Euclidean distance $d(P, Y)$ for the two-objective problem is calculated by

$$d(P, Y) = \sqrt{\sum_{j=1}^{2} \max(P_j - Y_j, 0)^2}.$$

As opposed to the IGD, the modified measure IGD^+ can differentiate the quality of PF_1 when it is nondominated by the reference points in P^*. If $|P^*|$ is large enough, IGD^+ can measure both diversity and convergence of the optimization algorithm. The details of finding the reference points P^* are explained in [44]. Minimizing $\text{IGD}^+$is a convergence criterion that indicates how far we are from the ideal Pareto frontier. In this paper, minimizing the IGD^+ is numerically studied to ensure the convergence of the optimizations in different case studies.

4. Case Studies

In this section, we aim to evaluate the proposed N-impulse optimal smooth trajectory generator, presented in Sections 2 and 3, in two case studies. We consider transfer time and control effort as the two Pareto cost functions. In all case studies, we solve the unconstrained multi-objective optimization problem and investigate the effect of constraints in the formation of the Pareto frontier. To examine the efficiency of the proposed optimization algorithm, we compare the resulted unconstrained Pareto frontier with that of the MOGA. The optimization parameters of the MOGA and the modified NSGA-II are selected the same, wherever possible. Moreover, the unconstrained solutions closest to the ideal points are compared with the optimal Lambert approach developed for chasing a satellite, in this paper. In the proposed Lambert approach, the design variables are the transfer time

and the location of the first impulse, and the control effort is the objective function. The optimization performed to find the optimal Lambert solutions uses a single-objective GA with the same parameters. Note that the trajectories generated by the Lambert approach are not necessarily smooth. However, the proposed smooth 2-impulse trajectories can be considered as a specific case of the Lambert approach, where the smoothness constraint is applied. In the following case studies, it is assumed that $\theta_S(0) = 270\,\text{deg}$, $\theta_T(0) = 0\,\text{deg}$, $N_{max} = 5$. In the constrained optimizations, we consider the following constraints: (i) $\Delta v_i \leq 6\,\text{km/sec}$ for $i = 1, \cdots, N$, and (ii) staying within LEO regime, i.e., (26). The typical value of the crossover rate used in the literature is [0.8, 0.95] [37]. For constant mutation functions, the rate is selected to be reciprocally proportional to the number of design variables, which is almost 0.4 in our case studies; hence, $PM_0 = 0.4$. Based on an investigation conducted on the optimization time and the quality of the produced solutions, the number of population and the maximum number of iterations are obtained to be $K = 50$ and $D = 200$, respectively. Moreover, the optimal number of reference points in calculating the IGD^+ is found to be around $|P^*| = 100$. To ensure convergence, the behaviour of IGD^+ over the course of iterations is monitored. We consider convergence of IGD^+ to a number less than 1 as the indicator of the multi-objective optimization convergence.

As indicated in [37], the computational complexity of the NSGA-II for a problem with two objective functions is $\mathcal{O}(2K^2)$. The Newton method is implemented to evaluate the objective functions by solving a system of $2N_{max}+1$ nonlinear equations with the computational complexity $\mathcal{O}((2N_{max}+1)^3)$ [45]. It is initialized using $\mathsf{ga}()$ with the computational complexity $\mathcal{O}(F(2N_{max} +$

1)), where F is the number of population multiplied by the number of generations. Overall, the computational complexity of the proposed algorithm for solving (32) is $\mathcal{O}(K^2 N_{max}^3)$.

4.1. Case study 1

In this case study, the initial and final orbital elements are:

$$a_S = 13756\,\text{km} \qquad e_S = 0.5 \qquad \omega_S = 10\,\text{deg}$$

$$a_T = 13756\,\text{km} \qquad e_T = 0$$

This case study is simulated with and without constraints. The Pareto optimal solutions for the unconstrained optimization along with the convergence criterion IGD^+, as well as the transfer trajectories for the two extreme solutions are shown in Figures 8-11, respectively. The Pareto frontier set in this case only includes solutions with three and four number of impulses. The initial/final locations of the servicer and target, and the location of impulses are shown in Figures 10-12. The blue-dashed curve is the proposed optimal smooth transfer trajectory. It can be observed in Figure 10, which is the trajectory for the solution with the minimum time and maximum control effort, the servicer loiters in an inner orbit with high eccentricity until it catches the target. However, in the solution with minimum control effort and maximum time, the servicer waits in the first orbit for a long period of time and transfers to the target orbit using elliptic arcs with low eccentricity. The optimal transfer time and control effort for this case study are shown in Table 1. The minimum transfer time corresponds to a trajectory with $N = 3$, which is almost 16 times faster than the second extreme solution. However, the extreme

solution with minimum J_c has four impulses and demonstrates a control effort that is ten times smaller than that of the other extreme solution. It is worth noticing that all of the 50 initial population become non-dominating solutions. As we yet to apply the constraints, Figure 8 shows all the possible solutions to chase a target with the given initial condition, even if they are not feasible. Figure 8 also depicts the Pareto Frontier obtained using MOGA (blue stars). Both algorithms have the same maximum iteration and initial population. However, the computational time of solving the problem using MOGA is almost 2 times more than that of the NSGA-II, due to the increased computational complexity of the MOGA. Further, the MOGA can converge to only 10 optimal solutions for this case study, while all of them are dominated by the solutions of NSGA-II. To have a reasonable comparison between the proposed optimal trajectory design and the optimal Lambert, the solution closest to the ideal point is compared with the optimal Lambert in Figure 12. In the proposed optimal Lambert, the servicer waits in the initial orbit to find the right location for the first impulse, which is shown with the green star. Then, it catches the target with the Lambert trajectory shown with the red curve. It can be interpreted from Table 1 that the J_c^* of the Lambert solution is 4 times bigger than that of the proposed optimal transfer trajectory, with almost 30% smaller transfer time. The Pareto optimal solutions after applying constraints in (26) are shown in Figure 13. For the constrained optimization we find 40 non-dominated solutions out of 50 population, with three impulses. We plot the changes in the planar orbital elements in time for the trajectory closest to the ideal point in Figure 14 to visualize the dynamics of the optimal solution.

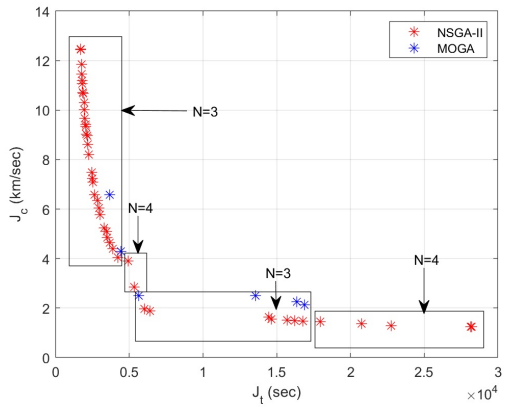

Figure 8: Case 1; Unconstrained Pareto Frontier

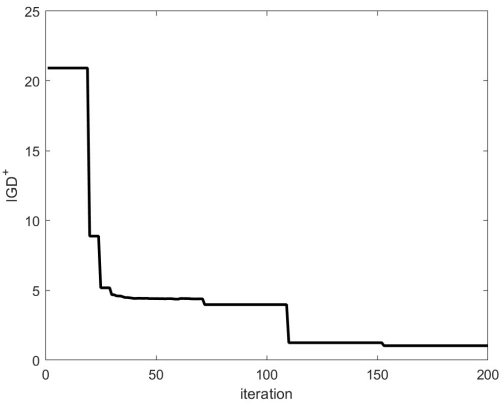

Figure 9: Case 1; IGD$^+$;

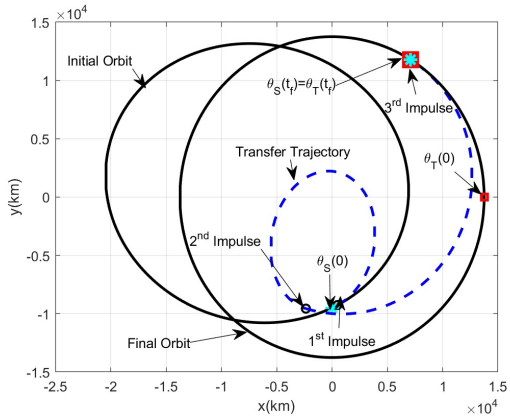

Figure 10: Case 1; Extreme point 1

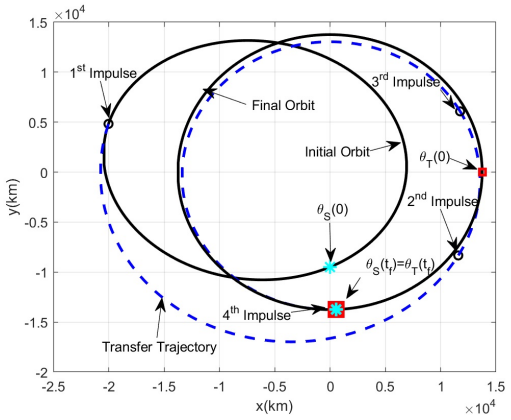

Figure 11: Case 1; Extreme point 2

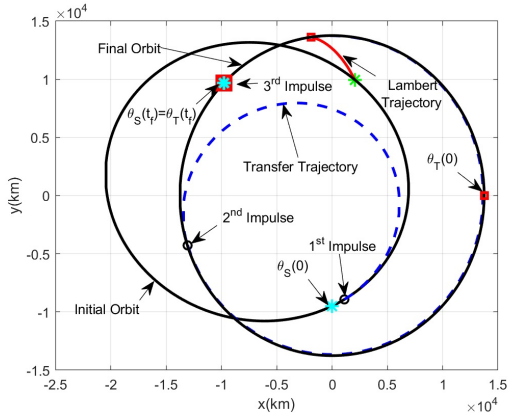

Figure 12: Case 1; Ideal point vs. Lambert

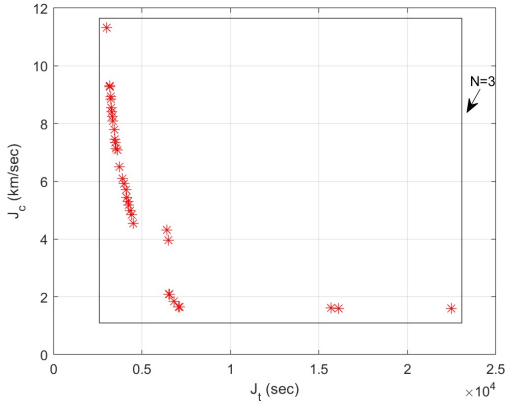

Figure 13: Case 1; Constrained Pareto Frontier

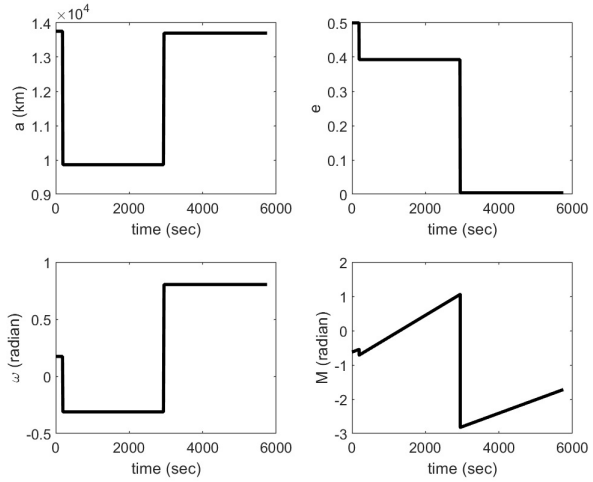

Figure 14: Case 1; Dynamics of the optimal solution closest to the ideal point

Table 1: Summary of the optimal results; Case study 1

Method	N	J_c [km/s]	J_t [s] ($\times 10^3$)	Design variables
Extreme point 1	3	12.46	1.72	ω_2, θ_1
Extreme point 2	4	1.25	28.21	ω_2, ω_3, ω_4, θ_1
Closest to ideal point	3	1.97	6.03	ω_2, θ_1
Optimal Lambert	2	7.74	4.36	t_f, θ_1

4.2. Case study 2

In this case study, the transfer trajectories from an eccentric orbit to a circular orbit in LEO are studied. The initial and final orbital elements are:

$$a_S = 7000\,\text{km} \qquad e_S = 0.05 \qquad \omega_S = 10\,\text{deg}$$

$$a_T = 7500\,\text{km} \qquad e_T = 0$$

The determined unconstrained Pareto optimal solutions are shown in Figure 15. Also, the transfer trajectories for the two extremes (both minimum

transfer time and minimum control effort are for $N = 3$) and the solution closest to the ideal point ($N = 4$) are depicted in Figures 17, 18, and 19, respectively. Given the termination and convergence criteria defined for the modified NSGA-II (Figure 16), 50 Pareto optimal trajectories were found in this case study. Evidently, the trajectory corresponding to the minimum transfer time drastically reduces the mission duration (almost 9 times), with a large compromise in the control effort (almost 43 times), comparing to the one with minimum control effort. Figure 15 also depicts the Pareto Frontier obtained using the MOGA (blue stars). Similar to the previous case study, the Pareto frontier consists of 17 solutions, all of which are dominated by the NSGA-II solutions. The optimal solution closest to the ideal point is compared with that of the optimal Lambert approach in Table 2 and Figure 19, proving that the Lambert solution is not efficient. The optimal control effort and the corresponding transfer time in the Lambert approach are almost 8.6 and 1.3 times larger than those of the proposed optimal trajectory generator, respectively. Figure 20 shows the Pareto frontier after enforcing the trajectories to be in LEO and considering the limitation in Δv_i. Here, the number of constrained optimal solutions is 50 and they include most of the three-impulse unconstrained solutions with $J_c^* \leq 0.7$. The dynamics of the planar orbital elements in time for the trajectory closest to the ideal point is depicted in Figure 21.

5. Conclusion

In this paper, we developed a multi-impulse shape-based trajectory optimization methodology for the long-range rendezvous phase of a servicer

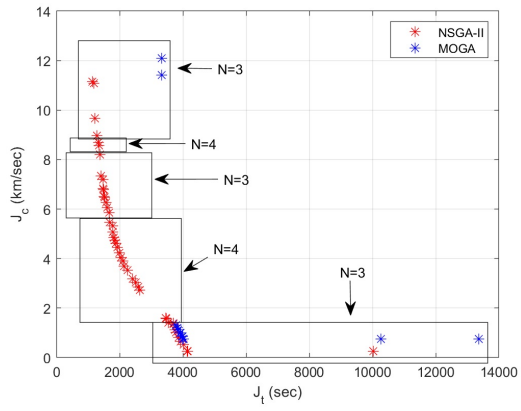

Figure 15: Case 2; Unconstrained Pareto Frontier

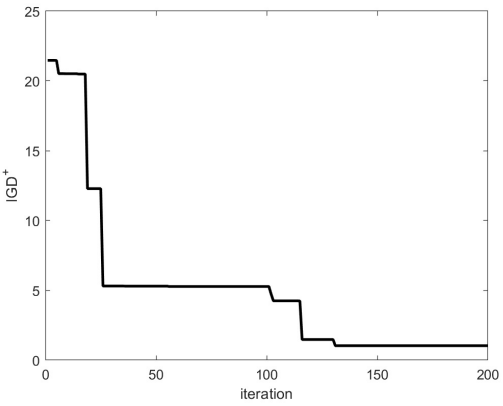

Figure 16: Case 2; IGD$^+$

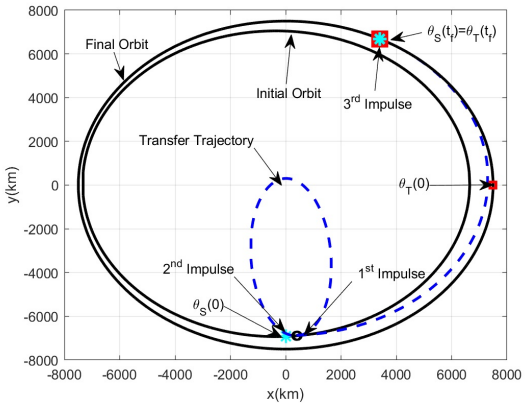

Figure 17: Case 2; Extreme point 1

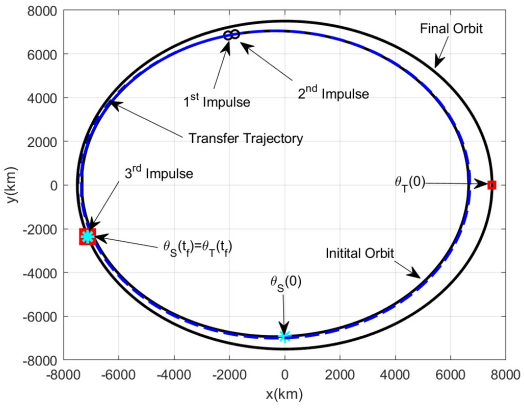

Figure 18: Case 2; Extreme point 2

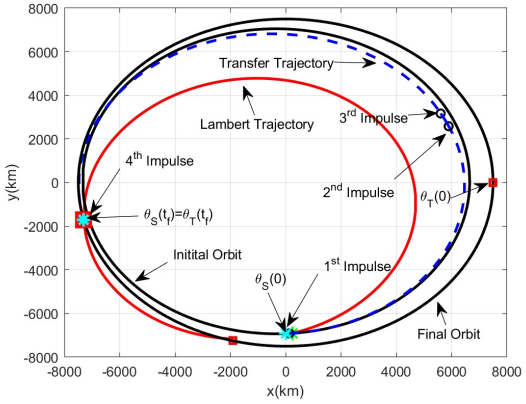

Figure 19: Case 2; Ideal point vs. Lambert

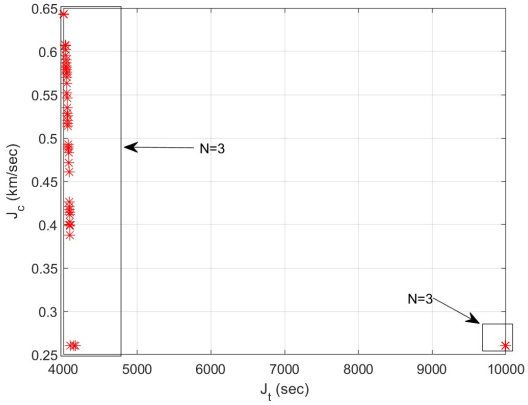

Figure 20: Case 2; Constrained Pareto Frontier

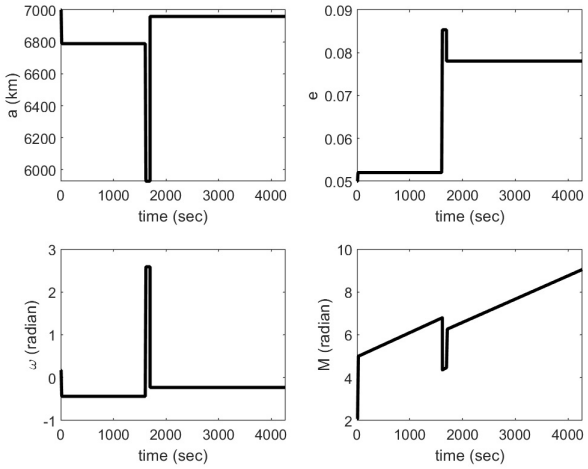

Figure 21: Case 2; Dynamics of the optimal solution closest to the ideal point

Table 2: Summary of the optimal results; Case study 2

Method	N	J_c [km/s]	J_t [s] ($\times 10^3$)	Design variables
Extreme point 1	3	11.14	1.13	ω_2, θ_1
Extreme point 2	3	0.26	10.02	ω_2, θ_1
Closest to ideal point	4	1.59	3.46	ω_2, ω_3, ω_4, θ_1
Optimal Lambert	2	13.64	4.58	t_f, θ_1

that chases a target in an OOS mission. In this process, three main problems were tackled: (i) smoothness, (ii) chasing, and (iii) optimality. The first two problems identified a number of constraints on the feasible trajectories. A non-dominated sorting genetic algorithm, NSGA-II, was implemented to solve the resulting constrained multi-objective optimization problem. To study the convergence of the proposed algorithm, IGD$^+$was used. To demonstrate the efficacy of the proposed methodology, we compared the obtained

optimal solutions with those generated by the MOGA and a developed optimal Lambert approach for chasing a target, in two case studies. In addition to the inherent complexity of the MOGA that resulted in slower optimization process, it was evident from the case studies that the proposed optimization based on the NSGA-II was superior to the MOGA. The size of the Pareto frontier using the NSGA-II was larger and its solutions dominated those that obtained from the MOGA. In the Lambert approach, the design variables were the transfer time and the impulse location in the initial orbit, while the objective function was the control effort. Overall, the multi-impulse smooth trajectory design methodology provided improved solutions in terms of both transfer time and control effort, as observed from Tables 1-2.

Acknowledgments

This work is supported by a grant from the Natural Sciences and Engineering Research Council of Canada (DGECR-2019-00085).

References

[1] On-orbit satellite servicing study project report, national aeronautics and space administration goddard space flight center. https://nexis.gsfc.nasa.gov/library.html (accessed 22 july 2021).

[2] M. Oda, Experiences and lessons learned from the ets-vii robot satellite, in: Proceedings of IEEE International Conference on Robotics and Automation, 2000, pp. 914–919. doi:10.1109/ROBOT.2000.844165.

[3] Osam-1; on-orbit servicing, assembly, and manufacturing 1. https://nexis.gsfc.nasa.gov/osam-1.html (accessed 22 july 2021).

[4] P. Rousso, S. Samsam, R. Chhabra, A Mission Architecture for On Orbit Servicing Industrialization, in: IEEE Aerospace Conference, 2021, pp. 1–14.

[5] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, Y. Xia, a review of optimization techniques in spacecraft flight trajectory design, Prog. Aerosp. Sci.109 (2019).

[6] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, C. P. Chen, review of advanced guidance and control algorithms for space/aerospace vehicles, Prog. Aerosp. Sci. 122 (2021).

[7] A. Shirazi, J. Ceberio, J. A. Lozano, Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions, Prog. Aerosp. Sci. 102 (2018) 76–98.

[8] M. S. Mohammadi, A. Naghash, Robust optimization of impulsive orbit transfers under actuation uncertainties, Aerosp. Sci. Technol. 85 (2019) 246–258.

[9] Q. Hu, J. Xie, X. Liu, Trajectory optimization for accompanying satellite obstacle avoidance, Aerosp. Sci. Technol. 82 (2018) 220–233.

[10] H. D. Curtis, Orbital mechanics for engineering students, 2nd Edition, Elsevier, Butterworth-Heinemann, Amsterdam, 2013.

[11] G. Zhang, D. Mortari, D. Zhou, Constrained multiple-revolution lambert's problem, J. Guid. Control Dyn. 33 (6) (2010) 1779–1786.

[12] J. A. Kechichian, The algorithm of the two-impulse time-fixed non-coplanar rendezvous with drag and oblateness effects, J. Astronaut. Sci. (1998) 47–64.

[13] O. Abdelkhalik, D. Mortari, N-impulse orbit transfer using genetic algorithms, J. Spacecr. Rockets 44 (2) (2007) 456–460.

[14] A. Shakouri, M. Kiani, S. H. Pourtakdoust, A new shape-based multiple-impulse strategy for coplanar orbital maneuvers, Acta Astronaut. 161 (2019) 200 – 208.

[15] E. Taheri, O. Abdelkhalik, Initial three-dimensional low-thrust trajectory design, Adv. Space Res. 57 (3) (2016) 889–903.

[16] M. Lv, M. Tan, D. Zhou, Design of two-impulse earth–moon transfers using differential correction approach, Aerosp. Sci. Technol. 60 (2017) 183–192.

[17] R. A. Phinney, R. Burridge, Representation of the elastic-gravitational excitation of a spherical earth model by generalized spherical harmonics, Geophys. J. Int. 34 (4) (1973) 451–487.

[18] D. A. Danielson, C. P. Sagovac, B. Neta, L. W. Early, Semianalytic satellite theory, Tech. rep., Naval Postgraduate School Monterey Ca Dept of Mathematics (1995).

[19] M. Ziebart, Generalized analytical solar radiation pressure modeling algorithm for spacecraft of complex shape, J. Spacecr. Rockets 41 (5) (2004) 840–848.

[20] B. M. Moghaddam, R. Chhabra, On the guidance, navigation and control of in-orbit space robotic missions: A survey and prospective vision, Acta Astronaut. 184 (2021) 70–100.

[21] J. Zhang, G.-j. Tang, Y.-z. Luo, Optimization of an orbital long-duration rendezvous mission, Aerosp. Sci. Technol. 58 (2016) 482–489.

[22] Y.-Z. Luo, G.-J. Tang, H.-y. Li, Optimization of multiple-impulse minimum-time rendezvous with impulse constraints using a hybrid genetic algorithm, Aerosp. Sci. Technol. 10 (6) (2006) 534–540.

[23] D. Kirk, Optimal Control Theory: An Introduction, Dover Books on Electrical Engineering Series, Dover Publications, 2004.

[24] J. Li, S. Gong, X. Wang, Analytical design methods for determining moon-to-earth trajectories, Aerosp. Sci. Technol. 40 (2015) 138–149.

[25] B. A. Conway, Spacecraft Trajectory Optimization, Cambridge Aerospace Series, Cambridge University Press, 2010, Ch. 1.2. doi:10.1017/CBO9780511778025.

[26] A. V. Rao, A survey of numerical methods for optimal control, Adv. Astronaut. Sci. 135 (1) (2009) 497–528.

[27] Y. Zhou, Y. Yan, X. Huang, L. Kong, Mission planning optimization for multiple geosynchronous satellites refueling, Adv. Space Res. 56 (11) (2015) 2612–2625.

[28] S. Samsam, R. Chhabra, Multi-impulse shape-based trajectory opti-

mization for target chasing in on-orbit servicing missions, in: IEEE Aerospace Conference, 2021, pp. 1–11.

[29] A. Shirazi, Analysis of a hybrid genetic simulated annealing strategy applied in multi-objective optimization of orbital maneuvers, IEEE Aerosp. Electron. Syst. Mag.32 (1) (2017) 6–22.

[30] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, solving constrained trajectory planning problems using biased particle swarm optimization, IEEE Trans. Aerosp. Electron. Syst. (2021).

[31] Y.-Z. Luo, G.-J. Tang, Rendezvous phasing special-point maneuvers mixed discrete-continuous optimization using simulated annealing, Aerosp. Sci. Technol. 10 (7) (2006) 652–658.

[32] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen, et al., Evolutionary algorithms for solving multi-objective problems, Vol. 5, Springer, 2007, Ch. 1-2.

[33] R. Chai, A. Savvaris, A. Tsourdos, Fuzzy physical programming for space manoeuvre vehicles trajectory optimization based on hp-adaptive pseudospectral method, Acta Astronautica 123 (2016) 62–70.

[34] R. Chai, A. Savvaris, A. Tsourdos, Y. Xia, An interactive fuzzy physical programming for solving multiobjective skip entry problem, IEEE Trans. Aerosp. Electron. Syst. 53 (5) (2017) 2385–2398. doi:10.1109/TAES.2017.2696281.

[35] R. Chai, A. Tsourdos, A. Savvaris, Y. Xia, S. Chai, Real-time reentry

trajectory planning of hypersonic vehicles: a two-step strategy incorporating fuzzy multiobjective transcription and deep neural network, IEEE Trans. Ind. Electron. 67 (8) (2019) 6904–6915.

[36] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, Y. Xia, unified multiobjective optimization scheme for aeroassisted vehicle trajectory planning, J. Guid. Control Dyn. 41 (7) (2018) 1521–1530.

[37] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[38] R. Chai, A. Savvaris, A. Tsourdos, Y. Xia, S. Chai, Solving multiobjective constrained trajectory optimization problem by an extended evolutionary algorithm, IEEE Trans. Cybern. 50 (4) (2020) 1630–1643. doi:10.1109/TCYB.2018.2881190.

[39] K. Deb, H. Jain, an evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2013) 577–601.

[40] C. M. Fonseca, P. J. Fleming, Multiobjective genetic algorithms, in: IEEE colloquium on genetic algorithms for control systems engineering, Iet, 1993, pp. 6–1.

[41] B. A. Barsky, T. D. DeRose, Geometric continuity of parametric curves: three equivalent characterizations, IEEE Comput. Graph. Appl. 9 (6) (1989) 60–69.

[42] Y. Censor, Pareto optimality in multiobjective problems, Appl. Math. Optim. 4 (1) (1977) 41–59.

[43] K. Deb, An introduction to genetic algorithms, Sadhana 24 (4-5) (1999) 293–315.

[44] Y. Sun, G. G. Yen, Z. Yi, Igd indicator-based evolutionary algorithm for many-objective optimization problems, IEEE Trans. Evol. Comput. 23 (2) (2019) 173–187. doi:10.1109/TEVC.2018.2791283.

[45] J. Zhao, E. A. Vollebregt, C. W. Oosterlee, A fast nonlinear conjugate gradient based method for 3d concentrated frictional contact problems, J. Comput. Phys.288 (2015) 86–100.

Appendix A. Algorithms for Crowding Distance and Ranking

Algorithm 2: Crowding Distance [37].

Data: $J_t(\mathbf{x}_k)$ and $J_c(\mathbf{x}_k)$ for $\mathbf{x}_k \in \mathrm{PF}_l$

Result: $\mathrm{CD}(\mathbf{x}_k)$ for $\mathbf{x}_k \in \mathrm{PF}_l$

$\mathbf{x}_t^l = \arg\max_{\mathbf{x}_k \in \mathrm{PF}_l} J_t(\mathbf{x})$

$\mathbf{x}_c^l = \arg\max_{\mathbf{x}_k \in \mathrm{PF}_l} J_c(\mathbf{x})$

PF_l =sort(PF_l, $J_t(\mathbf{x})$ or $J_c(\mathbf{x})$)

for $\boldsymbol{x}_k \in PF_l$ **do**

  if $\boldsymbol{x}_k = \boldsymbol{x}_t^l$ *or* $\boldsymbol{x}_k = \boldsymbol{x}_c^l$ **then**

    $\mathrm{CD}(\mathbf{x}_k) = \infty$

    else

      $\mathrm{CD}(\mathbf{x}_k) = \frac{J_c(\mathbf{x}_{k+1}) - J_c(\mathbf{x}_{k-1})}{J_c(\mathbf{x}_c^l) - J_c(\mathbf{x}_t^l)} + \frac{J_t(\mathbf{x}_{k+1}) - J_t(\mathbf{x}_{k-1})}{J_t(\mathbf{x}_t^l) - J_t(\mathbf{x}_c^l)}$

    end

  end

end

Algorithm 3: Ranking [37].

Data: $\mathbf{x}_k$, $J_t(\mathbf{x}_k)$, $J_c(\mathbf{x}_k)$ for $k = 1, \cdots, K$

Result: $S_{\mathbf{x}_k}$ and $n_{\mathbf{x}_k}$ for $k = 1, \cdots, K$, and PF_l

for $\boldsymbol{x}_k$, $k \leftarrow 1$ **to** K **do**
  $S_{\mathbf{x}_k} = \emptyset$
  $n_{\mathbf{x}_k} = 0$
  for $\boldsymbol{x}_{k'}$, $k' \leftarrow 1$ **to** K **do**
    if $\boldsymbol{x}_k$ *dominates* $\boldsymbol{x}_{k'}$ **then**
      $S_{\mathbf{x}_k} = S_{\mathbf{x}_k} \cup \{\mathbf{x}_{k'}\}$
      else
        $n_{\mathbf{x}_k} = n_{\mathbf{x}_k} + 1$
      end
    end
  end
end
if $n_{\boldsymbol{x}_k} = 0$ **then**
  $\mathrm{rank}(\mathbf{x}_k) = 1$
  $\mathrm{PF}_1 = \mathrm{PF}_1 \cup \{\mathbf{x}_k\}$
end
$l = 1$ **while** $PF_l \neq \emptyset$ **do**
  $Q = \emptyset$ Used to store the member of the next rank
  for $\boldsymbol{x}_k \in PF_l$ **do**
    for $\boldsymbol{x}_{k'} \in S_{\boldsymbol{x}_k}$ **do**
      $n_{\mathbf{x}_{k'}} = n_{\mathbf{x}_{k'}} - 1$
    end
  end
  if $n_{\boldsymbol{x}_{k'}} = 0$ **then**
    $\mathrm{rank}(n_{\mathbf{x}_{k'}}) = l + 1$
    $Q = Q \cup \{n_{\mathbf{x}_{k'}}\}$
  end
  $l = l + 1$
  $\mathrm{PF}_l = Q$
end