Accepted article (Author's version)
DOI: 10.1016/j.asoc.2023.110473

Robust Estimation of Distribution Algorithms via Fitness Landscape Analysis for Optimal Low-Thrust Orbital Maneuvers

Abolfazl Shirazi[a,*]

[a]*Basque Center for Applied Mathematics BCAM, Mazarredo 14, Bilbao, 48009, Bizkaia, Spain*

Abstract

One particular kind of evolutionary algorithms known as Estimation of Distribution Algorithms (EDAs) has gained the attention of the aerospace industry for its ability to solve nonlinear and complicated problems, particularly in the optimization of space trajectories during on-orbit operations of satellites. This article describes an effective method for optimizing the trajectory of a spacecraft using an evolutionary approach based on EDAs, incorporated with fitness landscape analysis (FLA). The approach utilizes flexible operators that are paired with seeding and selection mechanisms of EDAs. Initially, the orbit transfer problem is mathematically modeled and the objectives and constraints are identified. The landscape feature of the search space is analyzed via the dispersion metric to measure the modality and ruggedness of the search domain. The obtained information are used as feedback in developing adaptive operators for truncation factor and constraints separation threshold of the employed EDA. A framework for spacecraft trajectory optimization has been presented where the dispersion value for a space mission is estimated using a k-nearest neighbors (k-NN) algorithm. The suggested method is used to solve several problems related to low-thrust orbit transfer of satellites in Earth's orbit. Results demonstrate that the suggested framework for trajectory design and optimization of space transfers is effective enough to offer fuel-efficient and energy-efficient maneuvers for different thrust levels of the propulsion system. Moreover, the performance of the proposed approach is evaluated against non-adaptive EDA and other advanced evolutionary algorithms. The obtained results certify that the proposed adaptive evolutionary approach is superior in identifying feasible minimum-fuel and minimum-energy transfer trajectories.

Keywords: Orbit Transfer, Fitness Landscape Analysis, Evolutionary Algorithms,

*Postdoctoral Researcher, Machine Learning Group
Email address: ashirazi@bcamath.org (Abolfazl Shirazi)

1. Introduction

Algorithm development for guidance and control of space vehicles is one of the binding technologies for design and analysis of space missions. The optimization of low-thrust, multi-revolution orbit transfer trajectories is often regarded as a difficult problem in modern space engineering. Over the past decades, considerable efforts have been dedicated to the design of high-performance methodologies and algorithms for spacecraft trajectory optimization. Works by Prussing [1], Marec [2], and Edelbaum [3] are the first attempts in this field, following by noticeable advances in trajectory optimization. There are two techniques that are typically used, namely the direct method and the indirect method. [4]. The direct schemes discretize the state and control variables and convert the optimal control problem into a nonlinear programming (NLP) problem [5, 6]. The indirect strategies on the other hand, are based on the calculus of variations and Pontryagin's maximum principle, which generate a two-point-boundary-value problem [7, 8]. In recent years, Evolutionary Algorithms (EAs) and Machine Learning (ML) techniques have been merged with both of these methods in a good deal of research to enhance the performance of the approach in finding optimal transfer trajectories for space systems [9].

1.1. Evolutionary Algorithms in Spacecraft Trajectory Optimization

Significant progress has been observed in recent endeavors related to the utilization and development of EAs for optimizing spacecraft trajectories. In this regard, Samsam and Chhabra [10] solved trajectory generation of on-orbit servicing mission through a constrained multi-objective optimization algorithm based on non-dominated sorting Genetic Algorithm (NSGA-II). In this research, a scheme based on the generation of Pareto optimal trajectories is developed for long-range rendezvous. Caruso et al. [11] presented an indirect methodology for trajectory approximation of low-thrust orbital maneuvers. In this research, a shape-based method combined with Genetic Algorithm (GA) is presented for Earth to asteroid transfer trajectory design and optimization. In their work, Pallone et al. [12] presented a new swarm intelligence algorithm, in which an indirect technique evolved from heuristic algorithms is implemented to multi-stage launch vehicle trajectory optimization. Deep feed-forward neural networks (NN) methodology is developed by Li et al. in [13] for solving perturbed orbit transfer problems, where Particle

Swarm Optimization (PSO) merged with an indirect technique is used in multi-target mission design. Also, Xie and Dempster [14] presented an ML technique to find optimal finite thrust interplanetary transfer. They applied an optimization strategy based on NN to low-thrust interplanetary trajectory optimization. In another research, Zhou et al. [15] introduced an evolutionary method in on-orbit servicing, where GA merged in one-level and two-level optimization models is used in scheduling problem of multiple geosynchronous satellites refueling mission.

1.2. Estimation of Distribution Algorithms

Estimation of Distribution Algorithms (EDAs) are a type of EAs that have demonstrated encouraging results when it comes to addressing difficult optimization problems in astrodynamics [16]. EDAs have been integrated into several methods to solve trajectory optimization problems of space vehicles, and they have been shown to have promising performance [17, 18]. This group of EAs has demonstrated their ability to effectively solve highly challenging problems in astrodynamics and remain competitive in doing so. However, in the field of spacecraft trajectory optimization, little attention was paid to EDAs, in contrast to GA and PSO [4]. The most enhanced version of EDAs is EDA++, which has been recently developed by the author for constrained continuous optimization problems [19]. EDA++ is equipped with several heuristic mechanisms to deal with the satisfaction of nonlinear constraints, and it outperforms its rival EAs in terms of efficiency and execution time. Yet, still it treats the optimization problem as a black box with no adaptations. Since the main framework is based on probabilistic models and estimation, it has a flexible structure and therefore has a potential environment for developing adaptive mechanisms in constrained continuous optimization problems.

1.3. Motivation of Research

Following the aforementioned efforts in discovering unknown variables for achieving the optimal transfer trajectory, it can be realized that researchers typically either develop a new EA or select an arbitrary one to obtain the desired solution. However, little research has been conducted to investigate why a specific EA outperforms other competing algorithms in spacecraft trajectory optimization problem. Also, in case of choosing an arbitrary EA, the effectiveness of the selected algorithm, or its parameters in finding the optimal solution has not given proper attention in the literature. In particular, no clear

connection can be found between the selection of the EA, or rather the choice of the EA parameters, and the complexity of the spacecraft trajectory optimization problem. In such research, usually EAs in their best setup suited for a specific problem is implemented, and the reported results are associated with the best obtained solution out of multiple runs of the algorithm. It is unclear how the performance of the employed EA setup will be if some of the mission parameters (e.g., the desired final semi-major axis in a non-coplanar low-thrust orbit transfer) are changed. The question has remained unanswered whether the employed EAs, or the newly developed EAs are robust enough to deal with any mission parameters without the need for adjusting their parameters prior to optimization runs. The current article is primarily motivated by the knowledge and understanding gained from these insights. The primary focus of this research is to investigate the level of difficulty associated with orbit transfer problems, and to create an adaptive EA that benefits from these findings for adjusting the algorithm parameters. This concept is referred to as Fitness Landscape Analysis (FLA) [20], and it is linked to the development of intelligent algorithms for complex systems and the process of auto-tuning [21]. However, adequate attention has not been devoted to this concept in the field of astrodynamic. FLA includes various techniques that are used to measure the difficulty of the optimization problems by means of some metrics for calculating the complexity of the search domain. The work of Choi and Park [22] may be the sole effort in using FLA metrics to examine the complexity of spacecraft trajectory optimization problems. The authors conducted a good research on exploring the complexity of some well-known problems from Global Trajectory Optimization Problems (GTOPs) database [23] via different FLA metrics. However, there has been a scarcity of studies aimed at developing EAs that utilize insights obtained from FLA techniques for optimizing spacecraft trajectories. In this regard, the aim of this research is to utilize FLA techniques to develop new and robust approaches for spacecraft trajectory optimization based on EDAs.

1.4. Main Contribution

The approach in this study is an advanced version of the technique that involves converting the system input into parameters directly. The resulting optimization problem is solved through an effective evolutionary approach, specifically designed for non-coplanar orbit transfers. In this regard, a new technique is suggested for the optimization of low-thrust trajectories of spacecraft in Earth's orbit. The proposed approach is adaptive in

nature. After identifying the complexity of optimizing spacecraft trajectory for low-thrust orbital maneuvers using FLA techniques, novel adaptive operators for EDA++ are proposed. The developed adaptive operators use the insights gained from FLA techniques to enhance the algorithm's exploration and exploitation abilities while seeking the best possible feasible transfer trajectories. The justification for targeting the mechanisms of EDA++ in developing novel adaptive operators in this research lies upon the fact that this algorithm has already outperformed the majority of modern constrained continuous optimization algorithms [19], and since it works based on the framework of EDAs, it contains many parameters and components associated with probabilistic models to control its exploration and exploitation capabilities, which provides high level of flexibility for adaptation. To evaluate the effectiveness of the suggested approach, it is applied to solve multiple low-thrust orbit transfer problems. The method's superiority is demonstrated by comparing the solutions obtained through it to those obtained by several other state-of-the-art algorithms.

1.5. Research Outline

The remainder of this paper is structured as follows. Initially, the problem of finding low-thrust orbit transfer trajectories are mathematically modeled in Section 2. The dynamical equations, describing the motion of the spacecraft, are provided and the associated optimal control problem is turned into an optimization problem via parameterization of unknown functions using Chebyshev approximation. The objective function and the constraints are defined and consequently in Section 3, the overall workflow of EDA++ is introduced along with key algorithm parameter, which have been aimed for adaptation in this research. To measure the difficulty of the orbit transfer problem, several variables are introduced as problem identifiers. An FLA method is proposed, which utilizes a dispersion metric to compute the complexity of the search domain. Following the obtained feedback from the landscape feature analysis, the suggested approach introduces two adaptive operators that control the exploration and exploitation abilities of the search process. These operators are used for the truncation factor and constraints separation threshold, and are intended to aid the algorithm in finding optimal feasible solutions. To reduce the computational complexity, a k-nearest neighbors (k-NN) algorithm is used to estimate the dispersion value for a given problem. Consequently, based on the proposed mechanisms, an Adaptive Estimation of Distribution Algorithms (AEDA) is presented

to find fuel-optimal and energy optimal transfer trajectories. Detailed simulation case studies of the optimization architecture are proposed in Section 4 for several orbit transfer scenarios. The suggested method's effectiveness is evaluated by comparing it to other constrained evolutionary algorithms like ARMOR-DE [24] and DC3 [25], as well as the non-adaptive EDA++. Finally, Section 5 contains the conclusions of the research.

2. Problem Formulation

The low-thrust orbit transfer problem is aimed in this research, where a space vehicle is expected to travel from the initial orbit to the desired orbit using low-thrust propulsion. A schematic representation of the problem is illustrated in Fig. 1.

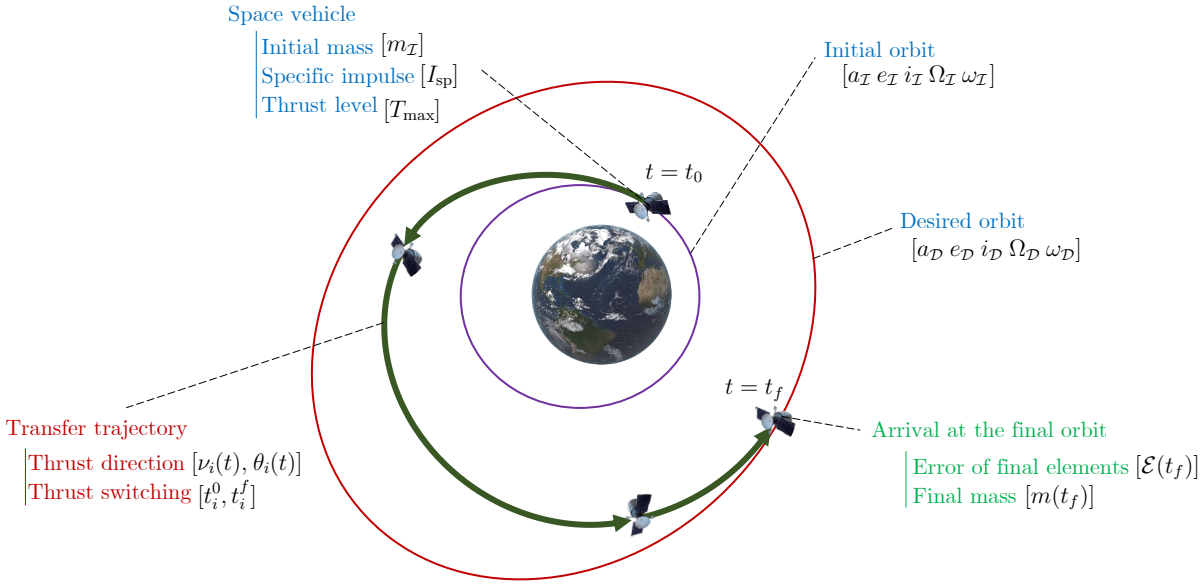

Figure 1: Representation of known variables (blue), unknown variables (red), objectives and constraints (green) in low-thrust trajectory optimization

The figure shows the known and unknown variables along with constraints and objectives with their corresponding symbols, which are going to be discussed in this section. The initial and final orbits are considered to be non-coplanar, which imposes more complexity to the problem in comparison to coplanar transfers. A feasible transfer trajectory is the one associated with reaching the final orbit with minimum errors of the orbital elements. Among all feasible trajectories, the optimal trajectory is the one associated with the least fuel or energy consumption for the spacecraft. The first step of solving any spacecraft trajectory optimization problem is to mathematically model the spacecraft dynamics along with problem inputs and constraints. Finding optimal space trajectories

in this research is an optimal control problem, which will be converted into an optimization problem via a direct method. The optimal control problem for spacecraft low-thrust trajectory optimization can be presented as follows

$$\begin{aligned} &\min_{u \in U} \mathcal{J} \\ &s.t. \\ &\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}, \mathcal{Q}, t) \\ &\mathcal{G}(\boldsymbol{x}_f, \mathcal{Q}) \leq 0 \\ &t \in [t_0, t_f] \end{aligned} \tag{1}$$

where $\mathcal{J}$ is the objective value as a function of state vector $\boldsymbol{x} : [t_0, t_f] \to \mathbb{R}^{n_x}$, control variables $\boldsymbol{u} \in L^\infty(U \subseteq \mathbb{R}^{n_u})$, and static parameters $\mathcal{Q} \in Q \subseteq \mathbb{R}^{n_Q}$. The functions $\boldsymbol{x}(t)$ belong to the Sobolev space $\mathcal{W}^{1,\infty}$, while the objective function is $\mathcal{J} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_Q} \to \mathbb{R}$. The objective is subject to a set of dynamic constraints with $\boldsymbol{F} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_Q} \times [t_0, t_f] \to \mathbb{R}^{n_x}$, and boundary conditions $\mathcal{G} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_Q} \to \mathbb{R}^{n_\mathcal{G}}$. One can say that the solution of the problem is a subset of U that satisfies the constraints. In the following subsections, the related equations for objectives, constraints, and other parameters shown in Fig. 1 are discussed to represent the conversion of optimal control problem as in Eq. 1 into a constrained continuous optimization problem.

2.1. Spacecraft Dynamics

Low-thrust orbital maneuvers are considered with two performance indices, including the energy-optimal and fuel-optimal indices. The propulsive thrust and the force due to Earth's gravitation are assumed to be the only affecting forces. The transfer follows the following set of dynamical equations [2, 4]

$$\begin{aligned} \dot{\boldsymbol{r}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= -\frac{\mu}{r^3}\boldsymbol{r} + \frac{T_{\max} u}{m}\boldsymbol{\psi} \\ \dot{m} &= -\frac{T_{\max} u}{I_{\text{sp}} g_0} \end{aligned} \tag{2}$$

where the spacecraft state is determined by its position vector $\boldsymbol{r} = (r_x, r_y, r_z)$, velocity vector $\boldsymbol{v} = (v_x, v_y, v_z)$, and mass m. Here, μ denotes the Earth's gravitational constant,

$T_{\max}$ represents the maximum thrust magnitude, $u \in [0, 1]$ is the engine thrust ratio that controls on-off switches, and $\boldsymbol{\psi}$ is the unit vector of thrust direction. The specific impulse of the thruster and average gravitational acceleration of Earth are respectively denoted as I_{sp} and g_0. These equations represent the function $\boldsymbol{F}$ in Eq. 1 with $\boldsymbol{x} = [\boldsymbol{r}; \boldsymbol{v}; m]$ and $\boldsymbol{u} = u\boldsymbol{\psi}$. The initial orbit and the desired orbit are known, and can be defined by five orbital elements as shown in Fig. 1. These elements define the shape and orientation of a space orbit, which include semi-major axis (a), eccentricity (e), inclination (i), right ascension of ascending node; RAAN; (Ω), and argument of perigee (ω). Having these elements along with true anomaly (or the epoch time), the vectors for the position and the velocity of the spacecraft on the orbit can be obtained. Therefore, the initial values of the state variables of Eq. 2 ($\boldsymbol{x}_i$) becomes known and it is possible to integrate the dynamical equations. Then, the terminal values ($\boldsymbol{x}_f$) can be evaluated and compared with the desired orbital elements to check the feasibility of the transfer trajectory. Overall, the known parameters for a unique low-thrust trajectory optimization problem as shown in Fig. 1 can be represented by the vector $\mathcal{Q}$ in Eq. 1 as

$$\mathcal{Q} = [a_{\mathcal{I}} \;\; e_{\mathcal{I}} \;\; i_{\mathcal{I}} \;\; \Omega_{\mathcal{I}} \;\; \omega_{\mathcal{I}} \;\; a_{\mathcal{D}} \;\; e_{\mathcal{D}} \;\; i_{\mathcal{D}} \;\; \Omega_{\mathcal{D}} \;\; \omega_{\mathcal{D}} \;\; m_{\mathcal{I}} \;\; T_{\max} \;\; I_{\text{sp}}] \tag{3}$$

Having the orbital elements for the initial and desired orbits as $[a_{\mathcal{I}} \; e_{\mathcal{I}} \; i_{\mathcal{I}} \; \Omega_{\mathcal{I}} \; \omega_{\mathcal{I}}]$ and $[a_{\mathcal{D}} \; e_{\mathcal{D}} \; i_{\mathcal{D}} \; \Omega_{\mathcal{D}} \; \omega_{\mathcal{D}}]$ respectively, along with initial mass of the spacecraft $m_{\mathcal{I}}$, specific impulse I_{sp}, and maximum available thrust level $T_{\max}$, a unique problem is established regarding the optimization of a non-coplanar orbital maneuver. The problem is to find the optimal on-off time intervals of thrust-arcs along with their associated thrust profile that establish a transfer trajectory with minimum-energy or minimum-fuel consumption subject to the satisfaction of initial condition and terminal constraints.

2.2. Direct Transcription

Discovering the optimal transfer trajectory for the aimed problem is subject to finding unknown profiles of thrust arcs along with their on-off timings during the transfer. These unknown functions and variables can be represented in $\mathcal{X}$, namely

$$\mathcal{X} = \left[t_1^{\mathcal{I}} \; \mathbf{T}_1 \; t_1^{\mathcal{F}} \;\; t_2^{\mathcal{I}} \; \mathbf{T}_2 \; t_2^{\mathcal{F}} \;\; \ldots \;\; t_i^{\mathcal{I}} \; \mathbf{T}_i \; t_i^{\mathcal{F}} \;\; \ldots \;\; t_{N_T-1}^{\mathcal{I}} \; \mathbf{T}_{N_T-1} \; t_{N_T-1}^{\mathcal{F}} \;\; t_{N_T}^{\mathcal{I}} \; \mathbf{T}_{N_T} \; t_{N_T}^{\mathcal{F}}\right] \tag{4}$$

where N_T is the number of thrust arcs, $t_i^{\mathcal{I}}$ and $t_i^{\mathcal{F}}$ $(1 \leq i \leq N_T)$ are the starting time and ending time of thrust arcs respectively, which construct the switching function u in Eq. 2 as

$$u = u(t_i^{\mathcal{I}}, t_i^{\mathcal{F}}) \quad i \in \{1, 2, ..., N_T\} \tag{5}$$

In Eq. 4, $\mathbf{T}_i$ are thrust profiles as functions of time in each respective time interval of $t_i^{\mathcal{I}} < t < t_i^{\mathcal{F}}$. This representation agrees with minimum-energy and minimum-fuel transfers, since the thrust magnitude is at maximum value within the thrust arcs as $|\mathbf{T}_i| = T_{\max}$ for $t_i^{\mathcal{I}} < t < t_i^{\mathcal{F}}$, and is equal to zero within the coast arcs as $|\mathbf{T}_i| = 0$ for $t_i^{\mathcal{F}} < t < t_{i+1}^{\mathcal{I}}$. This definition agrees with

$$\boldsymbol{u} = u\mathbf{T} \tag{6}$$

where $\mathbf{T}$ is a function of $\mathbf{T}_i$ for $i \in \{1, 2, ..., N_T\}$. The components of thrust vector within the thrust arcs can be defined as

$$\mathbf{T}_i(t) = T_{\max}\boldsymbol{\psi} = T_{\max}\begin{bmatrix} \cos\theta_i(t)\cos\nu_i(t) \\ \cos\theta_i(t)\sin\nu_i(t) \\ \sin\theta_i(t) \end{bmatrix} \tag{7}$$

with θ_i and ν_i as the steering angles of the space vehicle with respect to the reference frame that forms $\boldsymbol{\psi}$ in Eq. 2. As can be appreciated with the current definition, the optimal time history of steering angles are unknown and yet to be determined besides the on-off time intervals. Considering the upper and lower bounds for steering angles as $-\pi/2 < \theta_i(t) < +\pi/2, -\pi < \nu_i(t) < +\pi$, for each unique thrust arc, the variations are defined as approximated time-profiles via finite number of nodes for the steering angles as

$$[\theta(t), \nu(t)] = \mathscr{C}(\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_{N_p}, \hat{\nu}_1, \hat{\nu}_2, ..., \hat{\nu}_{N_p}) \tag{8}$$

with $\mathscr{C}(\cdot)$ as the approximation operator, which converts the given discrete nodes $\hat{\theta}_j, \hat{\nu}_j$ into continuous time-series. With respect to the fact that $-\pi/2 < \hat{\theta}_j < \pi/2, -\pi < \hat{\nu}_j < +\pi,\ (j = 1, ..., N_p)$, different schemes may be employed for this operator. In this research, Chebyshev polynomials are utilized in $\mathscr{C}(\cdot)$ to parameterize the time-histories of steering angles.

2.3. Chebyshev-based input parameterization

The basic idea behind the use of a Chebyshev approximation method for solving the optimal control problem is to find global polynomial approximations for the control functions in terms of their values at the points. Recalling the Weierstrass approximation theorem [26], let ψ be a representative of θ or ν as real-valued function defined on a compact interval $[t_i^{\mathcal{I}}, t_i^{\mathcal{F}}]$ of $\mathbb{R}$. Then, for any $\zeta > 0$, there exists a polynomial $\mathcal{P}(t)$ such that

$$|\psi(t) - \mathcal{P}(t)| \leq \zeta \tag{9}$$

for all $t \in [t_i^{\mathcal{I}}, t_i^{\mathcal{F}}]$. Having $\psi : [t_i^{\mathcal{I}}, t_i^{\mathcal{F}}] \to \mathbb{R}$, and $t_1 < ... < t_{N_p} \in [t_i^{\mathcal{I}}, t_i^{\mathcal{F}}]$ be given, there exists a unique polynomial of degree at most $N_p - 1$ that interpolates ψ at $t_1, ..., t_{N_p}$ in the sense that

$$\mathcal{P}(t_j) = \psi(t_j) \tag{10}$$

for $j = 1, ..., N_p$, where $\mathcal{P}$ is the interpolating polynomial of ψ at $t_1, ..., t_{N_p}$. The interpolating polynomial $\mathcal{P}$ can be expressed explicitly in Lagrange form by

$$\mathcal{P}(t) = \sum_{j=1}^{N_p} f(t_j)\mathcal{C}_j(t) \tag{11}$$

where the following equation stands for $\mathcal{C}_j(t)$

$$\mathcal{C}_j(t) = \prod_{k=1, k\neq j}^{N_p} \frac{t - t_k}{t_j - t_k} \tag{12}$$

If $\mathcal{P}$ and $\mathcal{C}$ are two polynomials of degree at most $N_p - 1$ that interpolate ψ at $t_1, ..., t_{N_p}$, then $w = \mathcal{P} - \mathcal{C}$, is a polynomial of degree at most $N_p - 1$ that vanishes at N_p points $t_1, ..., t_{N_p}$. It follows that w is identically zero, which implies $\mathcal{P} = \mathcal{C}$. The interpolation error can be identified by calculating the remainder term. Considering ψ to be an N_p times continuously differentiable function on a compact interval $[t_i^{\mathcal{I}}, t_i^{\mathcal{F}}]$ and $\mathcal{P}$ is the $N_p - 1$ degree polynomial that interpolates ψ at $t_1, ..., t_m \in [t_i^{\mathcal{I}}, t_i^{\mathcal{F}}]$, the remainder term can be defined as

$$\mathcal{R}(t) = \psi(t) - \mathcal{P}(t) = \frac{\psi^{(N_p)}(\xi)}{N_p!}(t - t_1)...(t - t_{N_p}) \tag{13}$$

for some $\xi \in [t_i^{\mathcal{I}}, t_i^{\mathcal{F}}]$. It follows that

$$|\psi(t) - \mathcal{P}(t)| \leq \frac{\phi(t)}{N_p!} \max_{y \in [t_i^{\mathcal{I}}, t_i^{\mathcal{F}}]} |\psi^{(N_p)}(y)| \tag{14}$$

where $\phi(t) = (t - t_1)...(t - t_{N_p})$ is the monic polynomial of degree N_p with roots at $t_1, ..., t_{N_p}$. Following this interpolation scheme, the points

$$t_k = \cos\Big(\frac{2k-1}{N_p}\frac{\pi}{2}\Big) \tag{15}$$

for $k = 1, ..., N_p$ are Chebyshev nodes of the first kind. They are roots of the degree N_p Chebyshev polynomial defined by

$$\mathcal{T}_{N_p}(t) = \cos(N_p \arccos t) \tag{16}$$

for $t \in [t_i^{\mathcal{I}}, t_i^{\mathcal{F}}]$. The Chebyshev polynomials satisfy the recursion formula $\mathcal{T}_0(x) = 1$, $\mathcal{T}_1(t) = t$,

$$\mathcal{T}_{k+1}(t) = 2t\mathcal{T}_k(t) - \mathcal{T}_{k-1}(t) \tag{17}$$

for $k \geq 1$, and thus the leading coefficient of $\mathcal{T}_k$ is 2^{k-1}. Moreover, observe that

$$|\mathcal{T}_{N_p}(t)| = |\cos(N_p \arccos(t))| \leq 1 \tag{18}$$

for $t \in [-1, 1]$. Thus, if $t_1, ..., t_{N_p}$ are Chebyshev nodes (of the first kind), then

$$|(t - t_1)...(t - t_{N_p})| = |\frac{1}{2^{N_p-1}}\mathcal{T}_{N_p}(t)| \leq \frac{1}{2^{N_p-1}} \tag{19}$$

for $t \in [-1, 1]$. If we define the map $l : [-1, 1] \leftarrow [t_i^{\mathcal{I}}, t_i^{\mathcal{F}}]$ by

$$l(t) = \frac{t_i^{\mathcal{F}} - t_i^{\mathcal{I}}}{2}t + \frac{t_i^{\mathcal{F}} + t_i^{\mathcal{I}}}{2} \tag{20}$$

then it follows that

$$|(t - l(t_1))...(t - l(t_{N_p}))| \leq \frac{1}{2^{N_p-1}}\Big(\frac{t_i^{\mathcal{F}} - t_i^{\mathcal{I}}}{2}\Big)^{N_p} \tag{21}$$

for $t \in [t_i^{\mathcal{I}}, t_i^{\mathcal{F}}]$. Combining Eq. 14 and Eq. 21 gives the following observations. Suppose that ψ is an N_p times continuously differentiable function on the compact interval $[t_i^{\mathcal{I}}, t_i^{\mathcal{F}}]$, and let $\mathcal{P}$ be the $(N_p - 1)$-degree polynomial that interpolates ψ at the points

$$t_k = \Big(\frac{t_i^{\mathcal{F}} - t_i^{\mathcal{I}}}{2}\Big)\cos\Big(\frac{2k-1}{N_p}\frac{\pi}{2}\Big) + \frac{t_i^{\mathcal{F}} + t_i^{\mathcal{I}}}{2} \tag{22}$$

for $k = 1, ..., N_p$. Then,

$$|\psi(t) - \mathcal{P}(t)| \leq \frac{2C_{N_p}}{N_p!}(\frac{t_i^{\mathcal{F}} - t_i^{\mathcal{I}}}{4})^{N_p} \tag{23}$$

for $t \in [t_i^{\mathcal{I}}, t_i^{\mathcal{F}}]$ where $C_{N_p} = \max_{y \in [t_i^{\mathcal{I}}, t_i^{\mathcal{F}}]} |f^{(N_p)}(y)|$. Following the presented parameterization approach, the vector of decision variables can be reformed by transformation of Eq. 4. For an arbitrary value of N_p to parameterize the steering angles, the total number of decision variables is $2N_T(1 + N_p)$. It is possible to have an estimation for the required number of thrust arcs N_T in minimum-fuel and minimum-energy transfers for a given space mission along with an initial guess for the time intervals of thrust arcs ($t_i^{\mathcal{I}}$ $t_i^{\mathcal{F}}$). One option is to take advantage of transfer trajectories obtained via impulsive maneuvers. There are numerous techniques available in the literature to obtain a solution for multi-impulse orbit transfers [27]. Such a solution contains a vector of impulse timings as $\tilde{t} = [\hat{t}_1 \ ... \ \hat{t}_{N_T}]$. Utilizing the impulse timings vector, the on-off timings are reformed as $t_i^{\mathcal{I}} = \hat{t}_i - \tau_i^{\mathcal{I}}$ and $t_i^{\mathcal{F}} = \hat{t}_i + \tau_i^{\mathcal{F}}$, with $\tau_i^{\mathcal{I}}$ and $\tau_i^{\mathcal{F}}$ being the time offsets with respect to the impulsive timing $\hat{t}_i$ in ith thrust arc. Following the new sets of variables, the decision vector, containing the variables which need to be optimized, is reformed to

$$\mathcal{X} = \Big[(\tau_i^{\mathcal{I}} \ \tau_i^{\mathcal{F}}) \ , \ (\hat{\theta}_{i,1} \ \hat{\nu}_{i,1} \ , \ ... \ , \hat{\theta}_{i,N_p} \ \hat{\nu}_{i,N_p})\Big]_{i \in \{1,...,N_T\}} \tag{24}$$

where $\hat{\theta}_{i,j}$ and $\hat{\nu}_{i,j}$ are the jth approximation nodes of ith thrust arc for $\theta(t)$ and $\nu(t)$ respectively. It is noteworthy that thrust arc timing offsets are bounded by the orbital period of coasting orbits in the solution of multi-impulse orbit transfer. Also, the choice of the number of nodes is nontrivial. However, it is possible to select a reasonable value for N_p based on the shape of thrust profiles obtained by different methods available in the literature. In this research, the choice of $N_p = 5$ has shown to be a proper selection for the number of interpolation points in each thrust arc. Merging the boundaries of timing offsets with the limits of steering angles as mentioned, the upper and lower bounds for the decision variables can be represented as $\mathcal{X}_{\min} \leq \mathcal{X} \leq \mathcal{X}_{\max}$, with $\mathcal{X}_{\min}$ and $\mathcal{X}_{\max}$ as the lower and upper bounds respectively.

2.4. Objective and Constraints

To achieve optimal orbit transfers, the cost function is defined as follows

$$\mathcal{J} = \lambda \frac{T_{\max}}{I_{\text{sp}} g_0} \int_{t_0}^{t_f} [u - \epsilon u(1-u)] dt \tag{25}$$

where λ is a normalization constant, and ϵ is the homotopy coefficient. This form of cost function is a transition representation between fuel-optimal and energy-optimal problems, which has been introduced by Bertrand and Epenoy [28], and is used in many low-thrust trajectory optimization problems [29]. When $\epsilon = 1$, the problem is an energy-optimal problem. With ϵ gradually decreasing from 1 to 0, the problem will turn into a fuel-optimal problem. It is clear that for fuel-optimal transfers, $\mathcal{J}/\lambda$ represents the amount of fuel mass consumed by the spacecraft during the transfer $(m_{\mathcal{I}} - m(t_f))$ with respect to the dynamical equation of motion in Eq. 2. The normalization constant is calculated as $\lambda = 1/t_{imp}$ with t_{imp} being the total transfer time based on multi-impulsive orbit transfer for a given space mission. Successful orbital maneuver is subject to reach the desired orbit with an acceptable error margin. Hence, the orbital error vector is defined as

$$\mathcal{E}(t) = \Big[(\frac{a(t) - a_{\mathcal{D}}}{\sigma_a})^2 \;\; (\frac{e(t) - e_{\mathcal{D}}}{\sigma_e})^2 \;\; (\frac{i(t) - i_{\mathcal{D}}}{\sigma_i})^2 \;\; (\frac{\Omega(t) - \Omega_{\mathcal{D}}}{\sigma_\Omega})^2 \;\; (\frac{\omega(t) - \omega_{\mathcal{D}}}{\sigma_\omega})^2\Big] - 1 \quad (26)$$

where $\sigma_{(\cdot)}$ denote the maximum allowable differences between the final value and the desired value for each orbital element. This representation of constraint violation, better known as the global criterion method, has been vastly used in optimization of space trajectories [4]. Having the error vector as a function of time and the final time t_f, the constraint function can be defined as

$$\mathcal{G} = \begin{cases} \mathcal{E}(t_f), & \text{if } \mathcal{E}(t_f) \leq 0 \\ K_{\mathcal{E}} \dfrac{\mathcal{E}(t_f)}{\mathcal{E}_0} + \dfrac{1}{\mathcal{E}_0 t_f} \displaystyle\int_0^{t_f} \mathcal{E}(t) dt, & \text{otherwise} \end{cases} \quad (27)$$

where $\mathcal{E}_0$ is the error at the initial time. Note that both $\mathcal{E}_0$ and t_f are known for a unique solution. By this definition, the constraint function properly distinguishes the feasible and infeasible transfer trajectories with corresponding penalties. Clearly according to Eq. 26, $\mathcal{E}(t_f) \leq 0$ means that the orbital element by the end of transfer is within the acceptable error margin. So, the transfer is feasible and the constraint function returns a negative value, proportional to the difference from the desired orbital elements. However, if $\mathcal{E}(t_f) > 0$, the final conditions are not satisfied and the constraint violation is considered to be the summation of the scaled error by the end of transfer and the scaled integral of the error. Obviously, two terms of second condition in Eq. 27 have the following

boundaries

$$0 < \frac{\mathcal{E}(t_f)}{\mathcal{E}_0} \leq 1 \tag{28}$$

$$0 < \frac{1}{\mathcal{E}_0 t_f} \int_0^{t_f} \mathcal{E}(t) dt \leq 1 \tag{29}$$

Considering Eq. 27 with boundaries of Eq. 28 and Eq. 29, it can be observed that for transfers with same final error, the one with less integral of errors are preferred. The reason is that adaptive operators rely on the amount of constraint violation to find feasible regions in the search domain. Trajectories that are infeasible and have high constraint violation tend to disrupt the heuristic search process. Conversely, trajectories that are infeasible but have a lower integral of errors for orbital elements help adaptive operators to better identify feasible regions. Also, the final error has the coefficient $K_{\mathcal{E}}$, which increases the weight of final error. Since tuning this coefficient is not the purpose of this research, this coefficient has been set to $K_{\mathcal{E}} = 10$ in this research to adjust the priority of final error. Note that both $\mathcal{E}(t_f)$ and $\mathcal{G}$ are 1×5 vectors. The developed algorithm is aimed to finding a vector of decision variables in the form of Eq. 24 for minimization of the objective function described by Eq. 25, while satisfying the presented nonlinear constraints $\mathcal{G} \leq 0$ by Eq. 27.

3. AEDA optimization approach

To solve the formed problem, two new adaptive operators for EDA++ is developed and utilized. Before going through the details of the adaptive operators, it is necessary to introduce the overall structure of EDA++ further. Recalling the initial description about EDAs and EDA++ from Section I, EDA++ is an advanced evolutionary optimization algorithm, which has been recently developed for solving optimization problems with nonlinear constraints in continuous domain [19]. The compact pseudo code of this algorithm is illustrated in Alg. 1. This modern EDA is equipped with feasibility conserving mechanisms, aiming at rapid discovery of high quality feasible solutions in constrained continuous optimization problems, yet still treats the problem as a black-box with no adaptations. In this subsection, the overall workflow of the original EDA++ is briefly described. However, details of the mechanisms and the optimization process are beyond the scope of this article. Hence, the reader is recommended to refer to [19] for discussions

Algorithm 1: Simplified pseudo code of the non-adaptive constrained EDA [19]

Input: Objective function, Constraints function, Boundaries
Settings: Algorithm parameters
1 Invoke **SEEDING** mechanism `// Generating initial population`
2 Perform **EVALUATION** `// Evaluating the objective values`
3 **while** *stopping criteria are not met* **do**
4 Invoke **SELECTION** mechanism `// Selecting high quality solutions`
5 Invoke **LEARNING** mechanism `// Building probabilistic models`
6 Invoke **SAMPLING** mechanism `// Sampling new solutions`
7 Invoke **REPAIRING** mechanism `// Repairing out-of-bound solutions`
8 Invoke **MAPPING** mechanism `// Mapping infeasible solutions to feasible region`
9 Perform **EVALUATION** `// Evaluating the objective values`
10 Invoke **REPLACEMENT** mechanism `// Forming new population`
11 Perform **EXTRACTION** `// Extracting best solution and updating parameters`
Output: Best solution

about the development, performance evaluation, and the details of the original algorithm workflow prior to proceeding further.

EDA++ consists of several mechanisms based on probabilistic models, which are executed during the optimization process. The optimization process starts with the use of the SEEDING mechanism (Line 1 of Alg. 1) to create an initial feasible population. Subsequently, the optimization loop begins with the availability of initial feasible solutions and their respective objective values acquired from the EVALUATION step (Line 2 of Alg. 1). In each iteration, the truncation SELECTION method is employed by the algorithm to pick the most promising individuals from the present population (Line 4 of Alg. 1). After selecting a population and determining their objective values, a probability model is developed through a LEARNING mechanism (Line 5 of Alg. 1). This mechanism involves dividing the selected population into clusters based on their constraint violation. A set of models is created using a mixture of components, with each component corresponding to a specific cluster. This is done in order to increase the probability of selecting feasible solutions. After creating the mixture of models, the SAMPLING method is used to generate new solutions (Line 6 of Alg. 1). The REPAIRING method then fine-tunes these newly generated solutions based on the boundary vectors (Line 7 of Alg. 1). Thus far, the solutions that have been generated are more likely to fall within the feasible region due to the use of the mechanisms, which are designed to generate feasible solutions. Despite this, the algorithm still has a tendency to occasionally produce infeasible solutions. As a result, the MAPPING mechanism (Line 8 of Alg. 1) is utilized to ensure feasibility,

whereby all possible infeasible solutions are transformed and brought within the feasible region, resulting in the creation of a population that is entirely feasible. Following the EVALUATION process of the obtained solutions (Line 9 of Alg. 1), the new individuals are merged with the previous population, and the REPLACEMENT mechanism (Line 10 of Alg. 1) is employed to create a new population and corresponding objective values for the current iteration. Each aforementioned mechanism has several parameters, which control the behavior of the algorithm. Changing these parameters acts as balancing between exploration and exploitation capabilities of the algorithm while surfing the search domain. In this research, two of these parameters are aimed to be adapted by the features of the low-thrust orbit transfer problem.

3.1. Tuning Parameters for Efficient Search

One of the key algorithm parameters is the truncation factor. Represented by γ, this parameter belongs to the selection mechanism. Alg. 2 shows the implementation of selection mechanism via this parameter.

Algorithm 2: Pseudo code of the selection mechanism

Input: Population $\mathcal{X}_i$, Truncation factor γ

$\mathcal{J}_i \leftarrow$ Extract the objective values of the population
$\mathcal{S}_i^{\uparrow} \leftarrow$ Sort $\mathcal{X}_i$ based on the objective values $\mathcal{J}_i$
for $j \leftarrow 1$ **to** N_{pop} **do**
    if $j < \gamma N_{pop}$ **then**
        $\mathcal{S}_i^{sel}(j) \leftarrow \mathcal{S}_i^{\uparrow}(j)$
    else
        BREAK;

Output: $\mathcal{S}_i^{sel}$

With the boundary of $0 < \gamma < 1$, it controls the size of the selected population $\mathcal{S}_i^{sel}$ out of the current population $\mathcal{X}_i$ for creating the probabilistic model as

$$\mathcal{S}_i^{sel} = \left\{ \mathcal{S}_{i,j}^{\uparrow} | j \in \{1, ..., \gamma N_{pop}\} \right\} \tag{30}$$

where $\mathcal{S}_{i,j}^{\uparrow}$ is the jth individual of the sorted population $\mathcal{S}_i^{\uparrow}$ in the ith iteration of optimization according to the quality of the corresponding objective values $\mathcal{J}_i$, and N_{pop} is the size of the current population. Increasing the value for this parameter comes with more exploration of the search domain, but reduces the convergence rate.

Algorithm 3: Pseudo code of the seeding mechanism

Input: Constraints function $\mathcal{G}$, Constraints separation threshold κ

while *maximum seeding iteration is not reached* **do**
  if *first iteration* **then**
    $\mathcal{N}_i \leftarrow$Sample random population via uniform distribution
  else if *seeding regenration is switched* **then**
    $\mathcal{N}_i^{sel} \leftarrow$ Select top $N_{pop}\kappa$ solutions from the current population
    $\mathcal{U} \leftarrow$ Sample $N_{pop}(1-\kappa)$ solutions via uniform distribution
    $\mathcal{N}_i \leftarrow$ Merge $\mathcal{N}_i^{sel}$ and $\mathcal{U}$
  else
    Build mixture of probabilistic model from promising solutions
    $\mathcal{N}_i \leftarrow$ Sample new solutions using the mixture model
  Update seeding regeneration
  $\mathcal{J}_i \leftarrow$ Calculate constraint violation $\mathcal{G}(\mathcal{N}_i)$
  if *all* $\mathcal{J}_i$ *are feasible* **then**
    BREAK;

Output: $\mathcal{N}_i$

Another parameter is constraints separation threshold, which is associated with the seeding mechanism. It is represented by κ as in Alg. 3, acting as a threshold for separation of individuals based on their constraint violation within the seeding loop. Having $\mathcal{N}_i^{sel}$ as the selected population, the seeding mechanism forms a new population as

$$\mathcal{N}_i = \left\{ \{\mathcal{N}_{i,j}^{sel}, \mathcal{U}_k\} | j \in \{1, ..., \kappa N_{pop}\}, k \in \{1, ..., (1-\kappa)N_{pop}\} \right\} \tag{31}$$

where $\mathcal{U}_k$ is the sampled population based on the uniform distribution in the ith iteration of the seeding loop. Considering the boundaries of $0 < \kappa < 1$, with high values of this parameter, the algorithm tends to dedicate more process in exploring feasible solutions in exchange for reducing the priority for local search in objective function minimization.

The two aforementioned elements have been selected as the target parameters for adaptiveness. The main reason for targeting these parameters is that they are among the most crucial algorithm parameters which significantly affect the search process during the optimization. The interaction between the seeding and selection mechanisms makes these two parameters have determining trace in altering the exploration and exploitation capabilities of the algorithm.

3.2. Mission Characteristics

Due to the high complexity of the proposed orbit transfer problem, algorithm parameter selection is nontrivial. The two aforementioned parameters γ and κ are chosen to be adapted based on the features of the space mission for a given mission parameters

$\mathcal{Q}$ in Eq. 3. For each unique problem, there are thirteen parameters in $\mathcal{Q}$. However, it is more convenient to analyze the structure of the search domain with fewer parameters by considering the differences of the initial and desired orbital elements instead of their explicit absolute values. Moreover, the thrust to weight ratio has been taken into consideration instead of thrust and mass individually. Following this approach, five parameters referred to as problem identifiers are defined as follows

$$\mathscr{W}_s = \log\left(1 + \left(a_{\mathcal{D}}(1-e_{\mathcal{D}}^2) - a_{\mathcal{I}}(1-e_{\mathcal{I}}^2)\right)^2\right) \tag{32}$$

$$\mathscr{W}_i = \sin(|\frac{i_{\mathcal{D}} - i_{\mathcal{I}}}{2}|) \tag{33}$$

$$\mathscr{W}_o = \cos(\frac{\Omega_{\mathcal{D}} - \Omega_{\mathcal{I}}}{4}) + \cos(\frac{\omega_{\mathcal{D}} - \omega_{\mathcal{I}}}{4}) \tag{34}$$

$$\mathscr{W}_a = \frac{T_{\max}}{m_0} \tag{35}$$

$$\mathscr{W}_p = I_{\text{sp}} \tag{36}$$

where the orbital angles in $\mathscr{W}_i$ and $\mathscr{W}_o$ are in radians. As can be appreciated, the five parameters represent different aspects of the space mission. $\mathscr{W}_s$ represents the desired change in the shape of the space orbit, $\mathscr{W}_i$ and $\mathscr{W}_o$ represent the desired orientation of the space orbit, $\mathscr{W}_a$ represents the accessible acceleration for accomplishing the space mission, and $\mathscr{W}_p$ is the propellant characteristic. As can be observed, defining proper problem identifiers mainly depends on the significance of the chosen parameters and their physical meaning. In space orbit transfer missions, differences between parameters with the same units (such as initial and final semi-major axis) are important for meeting specific mission requirements, such as altering the shape of the orbit when transferring from one space orbit to another. The practicality of the chosen problem identifiers will be implicitly verified through experiments, which will show that combining adaptive operators based on the data acquired via these problem identifiers results in high-quality feasible solutions. On the other hand, using ineffective or improper problem identifiers does not enhance the performance of the adaptive approach over its non-adaptive version. Based on the presented mathematical model of the problem, space missions with high values for $\mathscr{W}_s$ and $\mathscr{W}_i$ and low values for $\mathscr{W}_o$, $\mathscr{W}_a$ generally produce optimization problems with high dimensions. $\mathscr{W}_p$ controls the sensitivity of the system dynamics, since it affects the rate of mass decrease due to fuel consumption. The variation of dimensions due to changes

in problem identifiers are nonlinear. Besides the problem identifiers, an augmented cost function is defined for evaluating the quality of given solutions as

$$\mathcal{H}(\mathcal{J},\mathcal{G}) = \begin{cases} \dfrac{\mathcal{J}}{m_{\mathcal{I}}}, & \text{if } \max(\mathcal{G}) \leq 0 \\ \sum \mathcal{G}, & \text{otherwise} \end{cases} \tag{37}$$

where $\mathcal{J}$ and $\mathcal{G}$ are functions for objective and constraints, as in Eq. 25 and Eq. 27 respectively. Clearly, the range of the augmented cost function is $0 < \mathcal{H} \leq 1$ for feasible trajectories and $1 < \mathcal{H}$ for infeasible trajectories. It should be clarified that the augmented cost function defined here is only used for landscape feature analysis, not the optimization. The optimization algorithm has its own internal mechanisms to deal with objectives and constraints [19].

3.3. Fitness Landscape Analysis

In order to develop proper adaptive operators, the complexity of the problem needs to be identified. The goal is to discover the effects of problem identifiers, proposed in Eq. 32 to Eq. 36, on the augmented cost function defined in Eq. 37, and try to match them with algorithm parameters γ in Eq. 30 and κ in Eq. 31 accordingly. The most common approach for achieving this goal is using FLA techniques for extracting the characteristics of the problem [20]. FLA methods propose various metrics for quantifying the problem characteristics [30, 31]. One of the most practical metrics for FLA is the *dispersion*, which gives useful information about the structure of the search domain. This metric, first introduced by Lunacek and Whitley [32], measures the average distance between pairs of individuals that are nominated as high quality solutions. A highly dispersed fitness landscape indicates that there are many possible search regions with different fitness values, while a low-dispersion landscape suggests that most search space regions have similar fitness values. Fitness landscape analysis often involves quantifying the dispersion of fitness values within a population, and exploring how this dispersion changes over time as evolution proceeds. This information can provide insights into the potential for evolutionary adaptation and the likelihood of populations becoming trapped in local optima. Dispersion has been originally introduced for continuous optimization problems, and has been used to study the search landscape of several problems [33, 34]. Dispersion extracts the level to which high quality solutions are concentrated in a given problem as

$$\mathcal{D} = \frac{1}{\delta n_{\mathcal{D}}(\delta n_{\mathcal{D}} - 1)} \sum_{i=1}^{\delta n_{\mathcal{D}}-1} \sum_{j=i+1}^{\delta n_{\mathcal{D}}} ||\mathcal{X}_i - \mathcal{X}_j|| \tag{38}$$

where $\mathcal{D}$ is the dispersion value, δ is the top percentage of the chosen high quality samples with respect to the augmented cost function $\mathcal{H}$, and $n_{\mathcal{D}}$ is the total number of selected population. Scaling the distances makes dispersion have a boundary of $0 < \mathcal{D} < 1$. Having the computational complexity of $\mathcal{O}(\delta {n_{\mathcal{D}}}^2)$, evolvability and deception of the search space in low-thrust orbit transfer problems can be identified with this metric. When samples approach the search space with most promising solutions (i.e., decrease in δ), the increase of the dispersion implies a weak global structure, which comes with a higher difficulty for solving the problem, hence more exploration is required. On the other hand, if lowering the threshold of promising solution results in low dispersion, more exploitation is needed for reaching the global optimal solution.

Following the proposed metric and problem identifiers, the landscape of the problem is analyzed in a data set of space missions. To generate a grid-like data set of space missions, first, 1000 different space missions are considered, which have been uniformly generated with respect to $6600\ km < a_{\mathcal{I}}, a_{\mathcal{D}} < 42164\ km, 0 < e_{\mathcal{I}}, e_{\mathcal{D}} < 0.8$ (subject to perigee altitude more than 200 km), $0 < i_{\mathcal{I}}, i_{\mathcal{D}} < \pi, 0 < \Omega_{\mathcal{I}}, \Omega_{\mathcal{D}} < 2\pi, 0 < \omega_{\mathcal{I}}, \omega_{\mathcal{D}} < 2\pi, 10^{-3}\ N < T_{\max} < 10\ N, 300\ kg < m_{\mathcal{I}} < 2000\ kg, 1500\ s < I_{\text{sp}} < 5000\ s$. Then, following the generated data for each mission parameter vector $\mathcal{Q}$ corresponding to every unique mission, additional mission sets are also generated and added to the data set by changing each of the mission parameter of Eq. 32 to Eq. 36 with 30 random values, while keeping the rest of the parameters fixed. Considering various thresholds, the dispersion is computed for each problem via 100 solutions uniformly distributed within the limits of $\mathcal{X}_{\min}$ and $\mathcal{X}_{\max}$ for each problem with respect to the augmented cost function $\mathcal{H}$ defined in Eq. 37. To evaluate the robustness of the feature, 50 different samples are taken for each problem and the dispersion is extracted with different thresholds within the range of $0.05 \leq \delta \leq 0.2$.

The bound-normalized dispersion for orbital shape identifier $\mathscr{W}_s$ is depicted in Fig. 2 along with its maximum and minimum values. According to the normalized dispersion for orbital shape, a relatively large decrease from the region with maximum value of 0.88372 to the region with minimum value of 0.37328 in dispersion can be observed as the problem identifier $\mathscr{W}_s$ increases. This decreasing behavior implies that high quality solutions are

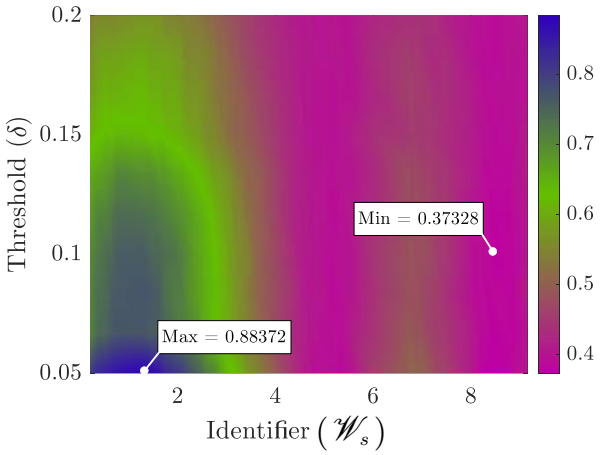

Figure 2: Normalized dispersion for orbital shape

increasingly closer together in orbit transfers with high changes in semi-major axis and eccentricity. It agrees with the fact that more revolutions, hence more thrust arcs and on-off thrust profiles, need to be discovered for orbit transfers when the desired change in orbital parameters becomes high. The bound-normalized dispersion values for orbital orientation identifier $\mathscr{W}_i$ has similar variations. Therefore, same insight can be inferred for this identifier as well. More changes in the orbital inclination (Δi) requires more thrust arcs and revolutions for a fixed thrust level.

Unlike the first two identifiers, the dispersion has an increasing variation for $\mathscr{W}_o$ and $\mathscr{W}_a$, which indicates that the solution domain is more chaotic for large changes in Ω and ω, and when thrust to mass ratio is low for a given orbit transfer problem. This increasing behavior for $\mathscr{W}_a$ is shown in Fig. 3, which illustrates overall dispersion increment from the region with minimum dispersion of 0.14499 associated with low values of $\mathscr{W}_a$ to the

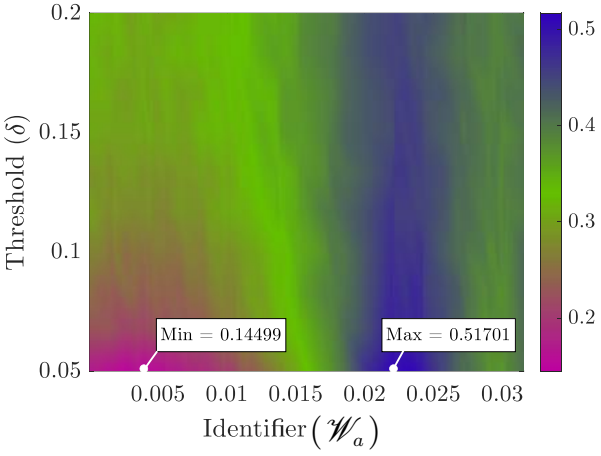

Figure 3: Normalized dispersion for accessible acceleration

region with maximum dispersion of 0.51701 associated with high values of $\mathscr{W}_a$.

As can be observed, the dispersion changes agree with the fact that more available acceleration comes with higher dispersion, and less number of required thrust arcs, which makes the optimization problem have lower decision variables and complexity. Dispersion change for $\mathscr{W}_o$ has similar behavior. Large changes in Ω and ω correspond to lower dispersion value for $\mathscr{W}_o$, as less thrust arcs are required for orbit transfers with low changes in Ω and ω, which corresponds to less complicated optimization problem with fewer decision variables. Between the proposed problem identifiers, $\mathscr{W}_p$ representing the specific impulse (I_{sp}), has shown to have the least effect on the dispersion value. The normalized dispersion for this identifier is shown in Fig. 4.

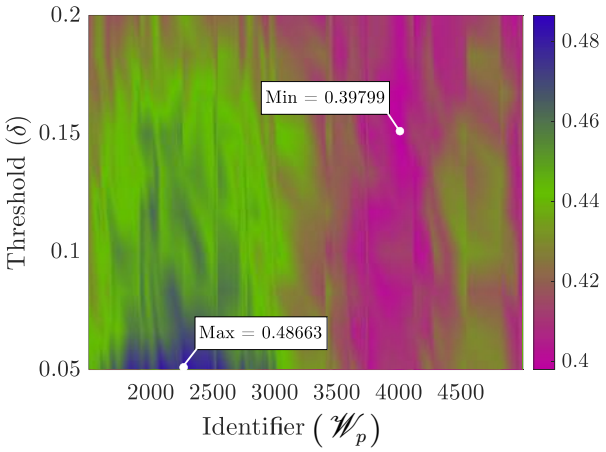

Figure 4: Normalized dispersion for propellant characteristic

Analysis of dispersion for $\mathscr{W}_p$ shows a very small range of changes of $0.39799 \leq \mathcal{D} \leq 0.48663$, which is significantly lower than the dispersion range associated with other problem identifiers. It implies that fewer changes of features are captured by dispersion metric regarding this problem identifier.

Analysis of dispersion evolution with respect to the changes of threshold shows noticeable variability of dispersion between selected samples for all problem identifiers. It indicates high discriminative ability of dispersion when orbital changes and thrust to mass ratio are not too high. Also, it can be observed that when threshold goes from low to high, the dispersion has less evolution for high values of identifiers, indicating an insignificant change in the landscape structure captured by dispersion for large changes in orbital elements and high thrust to mass ratios. Overall, analysis of the proposed metrics confirms relative reliability of dispersion in describing the complexity of the presented

finite-thrust trajectory optimization problem.

3.4. Adaptive EDA Operators

Given a space mission parameter vector $\mathcal{Q}$, the proposed landscape metric can be obtained via Eq. 38. The evolution of the dispersion value is monitored with decreasing δ within the boundary of $0.05 < \delta < 0.2$. The dispersion evolution $\Delta\mathcal{D}$ with respect to minimum and maximum threshold is used as an indicator to classify the orbit transfer complexity. A negative value of $\Delta\mathcal{D}$ implies that the best fitness values are localized in a small sub-region of the search space, which implies that more exploitation of the search domain is required to extract promising solutions. A $\Delta\mathcal{D}$ value around 0 corresponds to the fact that the best fitness values are spread over the entire search space. Also, higher values of $\Delta\mathcal{D}$ show localized promising solutions in distinct remote funnels, hence algorithms with more exploration capabilities are more preferred.

Following the dispersion evolution for a given problem, the adaptive operator for the truncation factor is defined as

$$\gamma = K_\gamma(\frac{1+\Delta\mathcal{D}}{2})^2 + C_\gamma \tag{39}$$

where K_γ and C_γ are the progressive truncation coefficients, which are updated within the optimization process as

$$K_\gamma = \frac{\sqrt{2}}{2} + \frac{\log(0.1+\rho)}{6\sqrt{3}} \tag{40}$$

$$C_\gamma = \frac{\sqrt{5}}{6}\cos(\frac{\rho\pi}{2}) \tag{41}$$

with ρ being the percentage of computational budget that has been used as the optimization goes on, representing the progress of the optimization process. The variation of the proposed truncation factor as the function of dispersion evolution is depicted in Fig. 5.

As can be observed, the truncation factor is adapted based on the dispersion evolution of the orbit transfer problem, which varies according to the complication of the search space. If the search domain for the trajectory optimization problem of the desired orbit transfer has high dispersion evolution, high values will be assigned to γ. This makes an increase in the population size within the selection mechanism, leading to increase exploration capability of the algorithm. On the other hand, γ is decreased for low dispersion evolution, leading the probabilistic model to be built based on small number of high quality solutions, i.e., more exploitation within the search process.

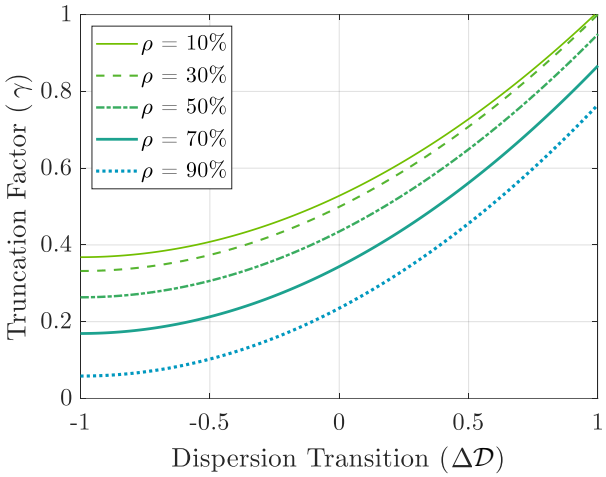

Figure 5: Adaptive truncation factor

Following the proposed operator, the adaptive truncation does not have a symmetric variation for $\Delta\mathcal{D} \geq 0$ and $\Delta\mathcal{D} \leq 0$. The coefficients defined in Eq. 40 and Eq. 41 show that higher dedication of individuals to the selected population is more prone to dispersion evolution. The main reason for such a consideration lies upon the natural behavior of the employed optimization framework, in which sharp variation of truncation factor for high dispersion evolution is more preferred for finding high quality solutions. Also, the truncation factor changes gradually toward dedicating less individuals to the selected populations as the optimization goes on (increase in ρ). This decrease in the truncation factor is more effective for $\Delta\mathcal{D} < 0$ than $\Delta\mathcal{D} > 0$ according to Fig. 5. This consideration is due to the fact that decreasing the truncation factor as the optimization goes on is relatively more desired when the need for high exploitation is critical. Statistically, the proposed variations have shown to be effective in finding optimal transfer trajectories while maintaining feasibility.

Besides the truncation factor, the adaptive operator for adjusting constraints separation threshold is defined as

$$\kappa = K_\kappa \sin\left(\frac{\pi(1+\Delta\mathcal{D})}{4}\right)^{P_\kappa} + C_\kappa \tag{42}$$

with K_κ, P_κ and C_κ being the coefficients for the threshold. The coefficients are calculated as

$$K_\kappa = 1 - \frac{\sqrt{\rho}}{2} \tag{43}$$

$$P_\kappa = 1 + 9\rho \tag{44}$$

$$C_\kappa = 0.4\rho^5 \tag{45}$$

which indicates the nonlinear dependency of κ on the optimization progress. The variation of the presented adaptive operator is illustrated in Fig. 6.

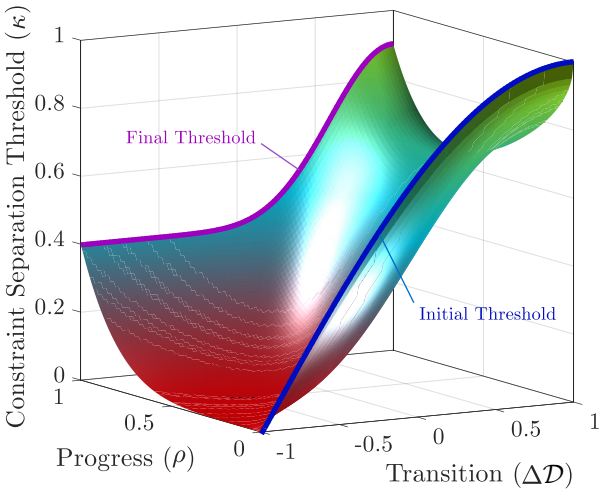

Figure 6: Adaptive constraints separation threshold

This representation shows overall gradual increase in the constraints separation threshold as the search space requires more exploration, i.e., high values of $\Delta\mathcal{D}$. The evolution of the separation threshold during the optimization process (from blue line to purple line in Fig. 6) illustrates the threshold variation for a variety of problem complexities. As can be observed, the separation threshold does not have much variation by the end of the optimization process when high exploitation is needed (neighborhood of $\rho = 1, \Delta\mathcal{D} < 0$). However, the increasing variation of threshold is noticeable at the beginning of optimization process regardless of the problem complexity ($\rho = 0$). This nonlinear variation has shown to be inline with more feasibility maintenance of the search process.

3.5. Estimation of Dispersion Evolution

Based on the proposed adaptive operators for truncation factor and constraints separation threshold, the algorithm is dynamically adapted if the dispersion evolution of a given problem is known. Recalling that the direct calculation of dispersion comes with the computational complexity of $\mathcal{O}(\zeta {n_\psi}^2)$, it is crucial to overcome the computational burden due to the fact that high computational burden can make the algorithm run slower or even make it infeasible to run on a particular system. In the other word, reducing computational burden is often a critical aspect of optimizing algorithms and improving performance. It is an important goal in developing an algorithm, as it can improve

	Average scores	
k_N	$\epsilon = 0$	$\epsilon = 1$
4	0.81295	0.84186
5	0.82054	0.76209
6	0.81923	0.81015
7	0.82880	0.87014
8	0.81825	0.78482
9	0.81623	0.75851
10	0.82007	0.73570

Table 1: Average scores of 8-fold CV for k-NN classifier

the efficiency, scalability, and accessibility of programs, while also making better use of computing resources. Following this insight, a k-NN with 8-fold cross validation (CV) is employed to estimate the dispersion evolution. Having the database of pre-calculated dispersion evolution for fitness landscape analysis in section 3.3, a k-NN classifier is implemented with problem identifiers $\mathscr{W}_s$, $\mathscr{W}_i$, $\mathscr{W}_o$, $\mathscr{W}_a$, and $\mathscr{W}_p$ as the features, and the dispersion evolution $\Delta\mathcal{D}$ as the output variable to be predicted. The size of the training data is considered as 80% and the remaining 20% of the data is used for testing. Euclidean distance is utilized, and the following score function is used for classification in every fold of test data-set.

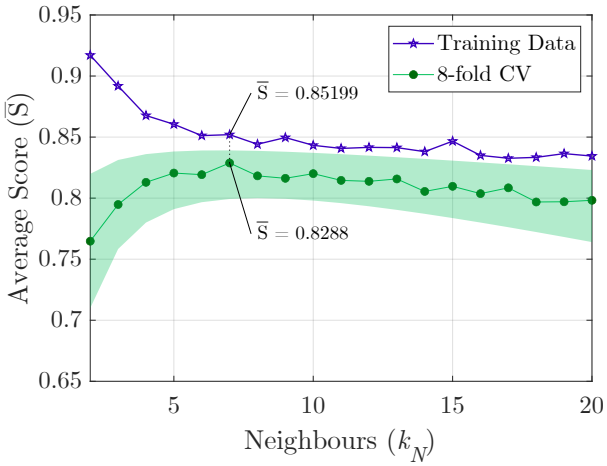

Figure 7: Estimation scores in k-NN classifier for fuel-optimal transfers

$$S_i = 1 - \frac{|\Delta\mathcal{D}_i - \Delta\hat{\mathcal{D}}_i|}{\max(|\Delta\mathcal{D}_i - \Delta\bar{\mathcal{D}}|)} \quad i \in \{1, ..., N_t\} \tag{46}$$

where N_t is the size of test data-set, $\Delta\mathcal{D}_i$ is the real dispersion evolution of the ith data, $\Delta\hat{\mathcal{D}}_i$ is the estimated dispersion evolution of the ith data, and $\Delta\bar{\mathcal{D}}$ is a vector of all dispersion evolutions of the training data-set. As can be appreciated, the defined score is 1 for perfectly estimated dispersion evolution, while the score of 0 corresponds to the worst prediction, which is associated with the maximum distance between the current test point from the training data-set. Having the score for each predicted value, the overall score for the test data-set is simply obtained via

$$\hat{S} = \frac{1}{N_t} \sum_{i=1}^{N_t} S_i \tag{47}$$

which is the average of scores for all data within the test data-set. This score is calculated for each fold of CV, and consequently the average of all scores associated with every fold will be the final score for a given number of neighbors k_N as

$$\bar{S} = \frac{1}{N_{CV}} \sum_{j=1}^{N_{CV}} \hat{S}_j \tag{48}$$

where N_{CV} is the total number of folds in CV. Fig. 7 shows the average score for 8-fold CV regarding different numbers of k_N in fuel-optimal problems. The tuning of k-NN is also done for energy-optimal problems as well. Details of the average scores for several values of k_N is provided in Table 1. As can be observed, the choice of $k_N = 7$ corresponds to the best prediction of dispersion evolution for both minimum fuel ($\epsilon = 0$) and minimum energy ($\epsilon = 1$) problems.

4. Numerical Results

The proposed framework is implemented using the presented approach to solve fuel-optimal and energy-optimal trajectory optimization problems. For a given problem of $\mathcal{Q}$, the problem identifiers ($\mathscr{W}_s$, $\mathscr{W}_i$, $\mathscr{W}_o$, $\mathscr{W}_a$, $\mathscr{W}_p$) are calculated initially. Then, the dispersion evolution $\Delta\mathcal{D}$ is estimated via the proposed k-NN classifier. Having the dispersion evolution as a representative of the problem complexity, EDA++ with the proposed adaptive operators for the truncation factor γ and the constraints separation threshold κ (i.e., AEDA) is used to solve the problem. The selection and the seeding mechanisms are

adapted with respect to the estimated dispersion evolution and also the progress of the optimization process. Following the procedure, the exploitation and exploration capabilities of the algorithm are adjusted with respect to the difficulty of the orbit transfer problem. The remaining parameters of the algorithm have been set according to their default values, as presented in [19]. Seeding reiteration coefficient is set to 100, which specifies the number of iterations for restarting the process of the seeding mechanism inside the algorithm. Smart cluster detection parameter, which acts as the threshold for separating smart clusters from the parent clusters is set to 0.01. The other key parameter of the learning mechanism as the outlier detection distance threshold is set to 1, which specifies the distance limit from the centroid of parent clusters for detecting smart clusters. Also, kernel density has been set to 0.5, which means that the algorithm samples equal number of populations for parent clusters and smart clusters. Moreover, the number of mapping steps is considered to be 10 within the mapping mechanism. This section provides the obtained results for applying the AEDA approach to fuel-optimal and energy-optimal transfers in two conducted experiments, where various analysis and comparisons are provided. The effort here is to bring different aspects of the results and observations to the attention in each experiment, while comparing the performance of the proposed approach with the non-adaptive version of the algorithm and other rival methods. It is noteworthy to remind that the two proposed operators are joint together within the optimization process of EDA. Adaptive truncation factor is associated with the selection mechanism, while the adaptive constraints separation threshold is associated with the seeding mechanism. The seeding mechanism maintains the feasibility of the initial population, while the selection mechanism adjusts the promising solutions based on their feasibility. The whole framework works effectively if and only if the two operators work together. With respect to this workflow, the non-adaptive version of the algorithm (EDA++) is included in the experiment setup to see the main effect of the whole technique, instead of testing each operator individually.

4.1. Fuel-optimal Transfer

The optimization of transfer trajectories for fuel-optimal ($\epsilon = 0$) low-thrust orbital maneuvers are taken into consideration in this experiment. The spacecraft is assumed to have the initial mass of $m_{\mathcal{I}} = 170kg$, and the specific impulse is assumed to be $I_{\mathrm{sp}} = 2200s$. The orbital elements of the initial orbit and the desired orbit are provided

in Table 2.

Table 2: Classical orbital elements of initial and desired orbits in fuel-optimal problem

	Initial orbit	Desired orbit
a	27500 km	34600 km
e	0.65	0.05
i	85°	50°
Ω	155°	315°
ω	25°	180°

The fuel-optimal low-thrust orbital maneuver problem is solved with respect to different thrust levels of $T_{\max} \in \{1.5, 2.5, 3.5, 4.5, 5.5\}N$. The allowable errors for the final orbital elements are considered as $\sigma_a = 5\,km, \sigma_e = 10^{-5}, \sigma_i = 10^{-3^\circ}, \sigma_\Omega = 10^{-2^\circ}, \sigma_\omega = 10^{-1^\circ}$. Fig. 8 shows the 3D visualization of space orbits with respect to the best obtained results for various thrust levels. On-off switches within the thrust arcs are also plotted for each case.

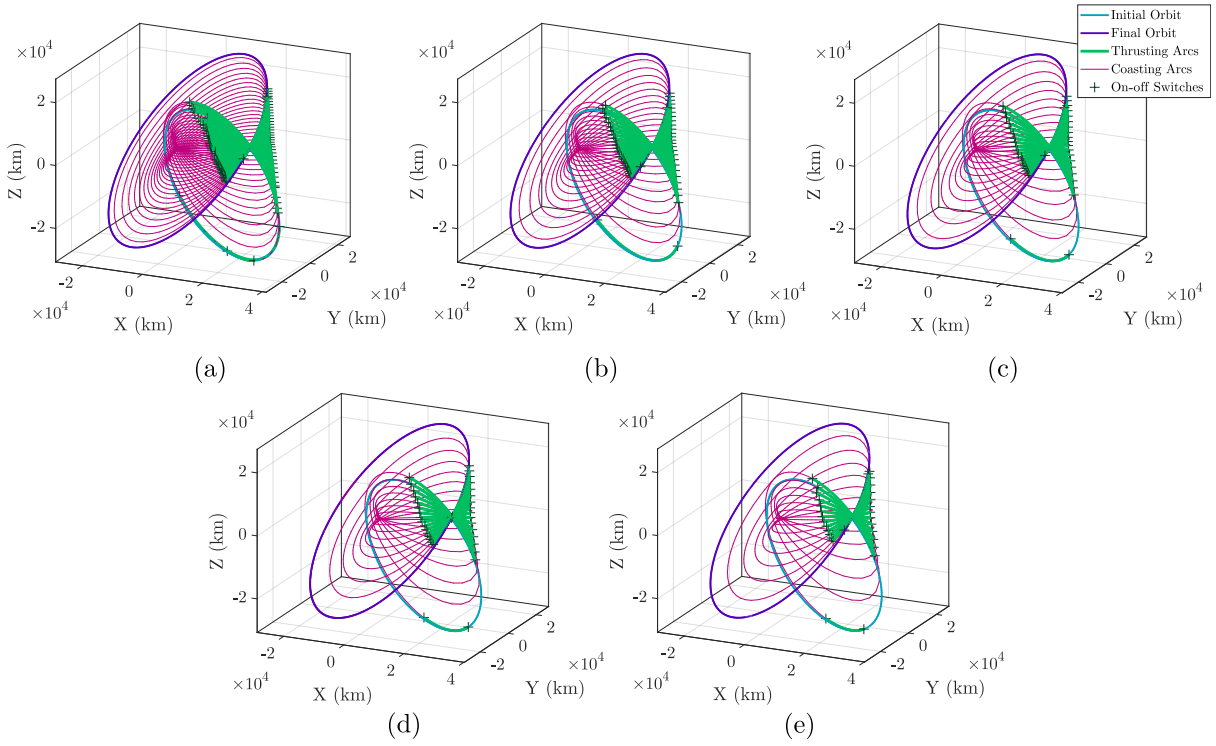

Figure 8: Visualization of transfer trajectories in fuel-optimal low-thrust orbital maneuver: (a) $T_{\max} = 1.5N$ (b) $T_{\max} = 2.5N$ (c) $T_{\max} = 3.5N$ (d) $T_{\max} = 4.5N$ (e) $T_{\max} = 5.5N$

As can be appreciated from the plots, the approach managed to find transfer trajec-

tories in various number of revolutions, while satisfying the initial and desired orbits. As the thrust level increases, lower number of thrust arcs are employed to reach the final orbit. Details are provided in Table 3, where the required number of revolutions along with the total transfer times are tabulated for each thrust level.

Table 3: Best obtained results for fuel-optimal transfers ($\epsilon = 0$)

$T_{\max}[N]$	Rev.	$t_f[m]$	$E_a[km]$	E_e	$E_i[deg]$	$E_\Omega[deg]$	$E_\omega[deg]$
5.5	15	8161.03	2.902e-02	9.106e-07	8.058e-05	7.323e-05	3.716e-03
4.5	18	10256.95	1.278e-01	1.099e-06	4.252e-04	5.811e-04	4.033e-03
3.5	21	11870.93	1.332e-01	1.604e-06	5.890e-04	7.175e-04	5.974e-03
2.5	27	15484.06	9.189e-01	4.435e-06	8.392e-04	2.868e-03	9.363e-02
1.5	41	24207.97	3.008e+00	8.148e-06	8.867e-04	4.333e-03	9.804e-02

According to the results, the shortest transfer time is 8161.03 minutes, associated with 15 revolutions for $T_{\max} = 5.5N$. On the other hand, when the maximum thrust level is set to $T_{\max} = 1.5N$, the desired orbit is reached after 41 revolutions in 24207.97 minutes. Table 3 also provides the error of orbital elements with respect to the final orbit for each thrust level. The variable $E_{(.)}$ represents the absolute difference between the desired orbital elements and the actual orbital elements of the final orbit. As can be seen, the proposed algorithm managed to satisfy the desired conditions based on the defined allowable errors. Results indicate that for space missions with shorter duration of the transfer time, the final error is smaller. This shows the fact that for less thrust levels, the problem becomes more complex with higher difficulty of detecting feasible region in the search space. Since lower thrust levels comes with optimization problems with higher decision variables, the difficulty of finding feasible transfer trajectories increases.

Having the best obtained solutions for each thrust level, it is possible to analyze many time-dependent variables and physical properties of the spacecraft. One comparison is illustrated in Fig. 9, in which the magnitude of state variables (position and velocity) are plotted for $T_{\max} = 1.5N$ and $T_{\max} = 5.5N$.

The variations show how the transfer trajectories are formed with respect to the internal acceleration acted on the spacecraft due to the dedicated thrust level of the propulsion system. The actual velocity increment (Δv) associated with thrust arcs can be extracted for each transfer trajectory as well. Fig. 10 shows the impulses for $T_{\max} = 5.5N$

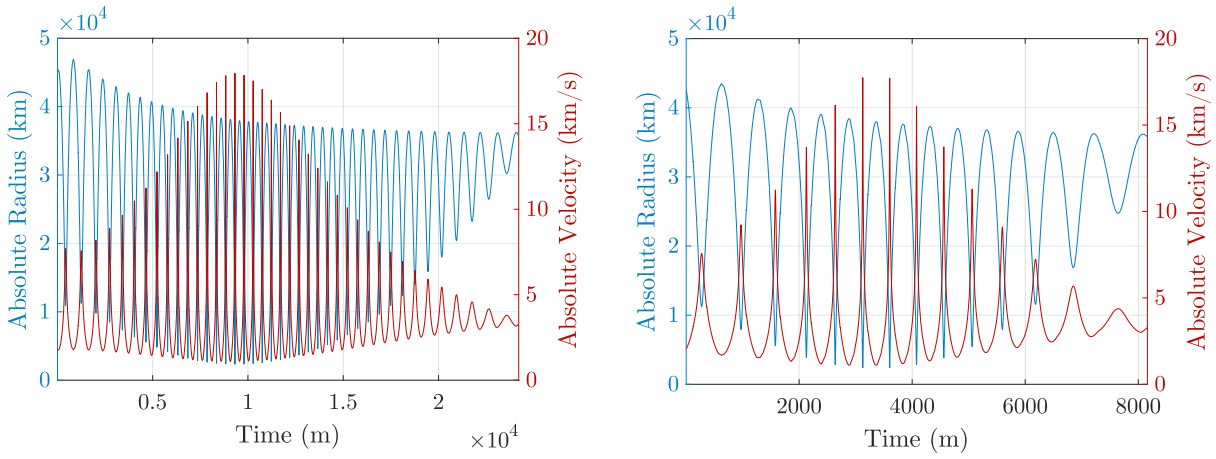

Figure 9: Magnitude of vectors for radius (blue) and velocity (red) of the spacecraft for fuel-optimal low-thrust orbital maneuver: $T_{\max} = 1.5N$ (left), $T_{\max} = 5.5N$ (right)

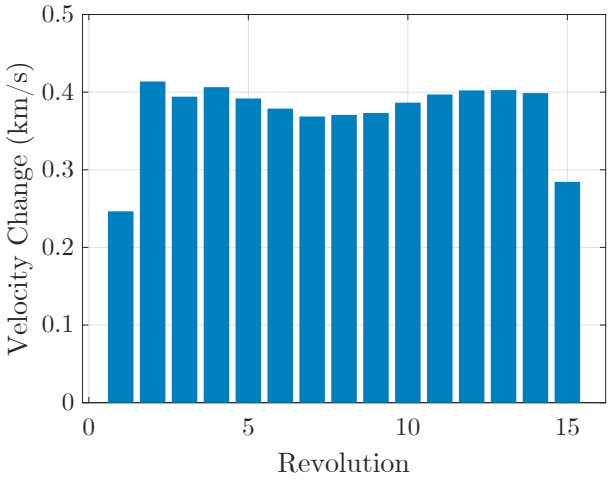

Figure 10: Distribution of effective velocity change (Δv) in 15 revolutions for fuel-optimal low-thrust orbital maneuver with $T_{\max} = 5.5N$

case, while the timings of acting thrust are illustrated in Fig. 11

As can be seen for $T_{\max} = 5.5N$, the minimum impulse is $\Delta v = 0.2465\, km/s$, which is acted on the spacecraft in the first revolution ($0 < t < 118.677$), while the second thrust arc includes the maximum impulse of $\Delta v = 0.4137\, km/s$ ($333.401 < t < 572.699$). The rest of the impulses due to the acting thrust arcs have small changes except the last impulse ($8049.52 < t < 8161.03$), where places the spacecraft in the final orbit with an impulse of $\Delta v = 0.2845\, km/s$.

Following the presented orbit transfer problem, the performance of the proposed approach is compared with the non-adaptive version of EDA (EDA++) and also other state-of-the-art algorithms for constrained continuous optimization including ARMOR-DE [24] and DC3 [25]. The chosen algorithms have proven to be the most competitive in terms

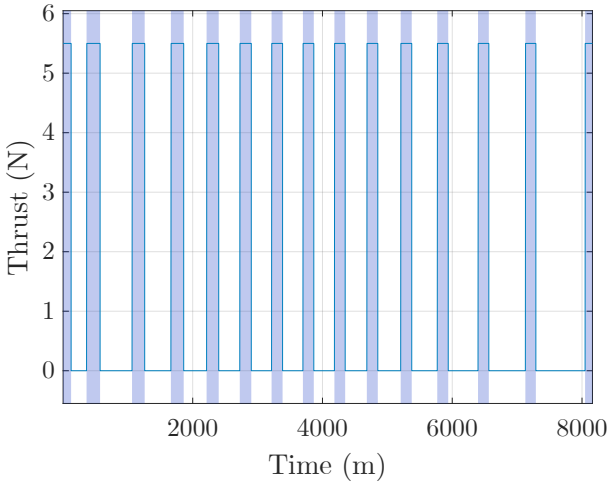

Figure 11: On-off switching thrust function for fuel-optimal low-thrust orbital maneuver with $T_{\max} = 5.5N$

of both the speed at which they execute and the quality of the solutions they provide, when compared to EDA++. Moreover, unlike the presented algorithm, these algorithms do not work based on probabilistic models. Considering the fact that the proposed algorithm relies on two notable components of parent clusters and smart clusters, it will be a promising comparison to validate the effectiveness of the two major components within the optimization process. For every case of $T_{\max}$, each algorithm is implemented and executed in 30 runs with same computational budget. Table 4 summarizes the achieved results.

Table 4: Algorithm performance comparison for fuel-optimal transfers ($\epsilon = 0$) in 30 runs

$T_{\max}$	AEDA		EDA++		ARMOR-DE		DC3	
	$\mathcal{J}_{\mathcal{G}}$	$\mathcal{J}_{\min}$	$\mathcal{J}_{\mathcal{G}}$	$\mathcal{J}_{\min}$	$\mathcal{J}_{\mathcal{G}}$	$\mathcal{J}_{\min}$	$\mathcal{J}_{\mathcal{G}}$	$\mathcal{J}_{\min}$
5.5	96.67	0.000e+00	90.00	1.170e-08	86.67	1.153e-16	96.67	5.558e-12
4.5	96.67	0.000e+00	76.67	1.766e-06	80.00	3.314e-11	33.33	2.663e-06
3.5	86.67	0.000e+00	56.67	5.852e-06	73.33	7.917e-08	30.00	1.509e-05
2.5	73.33	0.000e+00	56.67	9.328e-04	50.00	9.304e-03	23.33	6.471e-03
1.5	70.00	4.841e-02	23.33	0.000e+00	46.67	4.991e-01	06.67	1.160e-01

Two parameters for the measurement of algorithm performance are calculated. The first parameter is the feasibility ratio ($\mathcal{J}_{\mathcal{G}}$) of each algorithm, which simply shows the percentage of the total runs in which the algorithm succeeded in finding a feasible transfer trajectory (satisfying the constraints $\mathcal{G} \leq 0$ in Eq. 27) disregarding the amount of fuel

consumption (the objective function value $\mathcal{J}$ in Eq. 25). The other parameter is the relative best percentage ($\mathcal{J}_{\min}$), which is calculated as

$$\mathcal{J}_{\min} = \min(100 \times \frac{\vec{\mathcal{J}} - \mathcal{J}^*}{\mathcal{J}^*}) \tag{49}$$

where $\vec{\mathcal{J}}$ is the vector of objective values correspond to obtained feasible solutions by the algorithm, and $\mathcal{J}^*$ is the best obtained solution between all algorithms. Clearly, the $\mathcal{J}_{\min}$ of zero indicates that the algorithm managed to find the best possible solution between the rest of the algorithms, and any non-zero value represents relative difference of the best obtained solution with respect to the global best solution. In the interests of clarity, the mean values ($\bar{\mathcal{J}}$) and the variances of $\vec{\mathcal{J}}$ ($\sigma^2_{\mathcal{J}}$) are also provided in Table 5.

Table 5: The mean values and the variances of $\vec{\mathcal{J}}$ for fuel-optimal transfers ($\epsilon = 0$) in 30 runs

$T_{\max}$	AEDA		EDA++		ARMOR-DE		DC3	
	$\bar{\mathcal{J}}$	$\sigma^2_{\mathcal{J}}$	$\bar{\mathcal{J}}$	$\sigma^2_{\mathcal{J}}$	$\bar{\mathcal{J}}$	$\sigma^2_{\mathcal{J}}$	$\bar{\mathcal{J}}$	$\sigma^2_{\mathcal{J}}$
5.5	3.0269e-02	5.1402e-04	8.5469e-02	4.1639e-03	5.7588e-02	1.8515e-03	9.4246e-02	3.5001e-03
4.5	3.3457e-02	5.1343e-04	9.9312e-02	3.3476e-03	7.9764e-02	2.2510e-03	1.1523e-01	3.9336e-03
3.5	3.7038e-02	5.9883e-04	1.9155e-01	1.2860e-02	1.3760e-01	6.0348e-03	1.6548e-01	9.6135e-03
2.5	3.9981e-02	5.1868e-04	1.9470e-01	9.1872e-03	1.4432e-01	7.8948e-03	1.5984e-01	9.1285e-03
1.5	5.2689e-02	9.1511e-04	1.7078e-01	1.2468e-02	1.8017e-01	7.8536e-03	1.9413e-01	1.5032e-02

According to Table 4, the proposed adaptive approach managed to find feasible solutions in more than 90% of runs for $T_{\max} = 5.5N$ and $T_{\max} = 4.5N$. In this regard, the non-adaptive EDA and ARMOR-DE have shown to be competitive, yet with less feasibility ratios. Also, considering $\mathcal{J}_{\min}$ values, it can be verified that the best obtained solution for all cases belongs to the proposed adaptive approach except for $T_{\max} = 1.5N$, in which the best obtained solution via the adaptive approach is extremely close to the one obtained via EDA++ with $\mathcal{J}_{\min}$ in the order of 10^{-2}. The best obtained feasible solution belongs to EDA++ in this case ($\mathcal{J}_{\min} = 0$). However, the feasibility ratio of $\mathcal{J}_{\mathcal{G}} = 23.33\%$ versus $\mathcal{J}_{\mathcal{G}} = 70.00\%$ indicates more reliability of the adaptive approach in comparison to the non-adaptive method. Overall, it can be observed that the advantage of using the adaptive approach is justified for various amount of thrust levels.

4.2. Energy-optimal Transfer

The second experiment is dedicated to energy-optimal transfers ($\epsilon = 1$) and the orbital elements of initial and desired orbits are assumed as in Table 6.

Table 6: Classical orbital elements of initial and desired orbits in energy-optimal problem

	Initial orbit	Desired orbit
a	20800 km	36000 km
e	0.3	0.05
i	5°	20°
Ω	245°	320°
ω	240°	235°

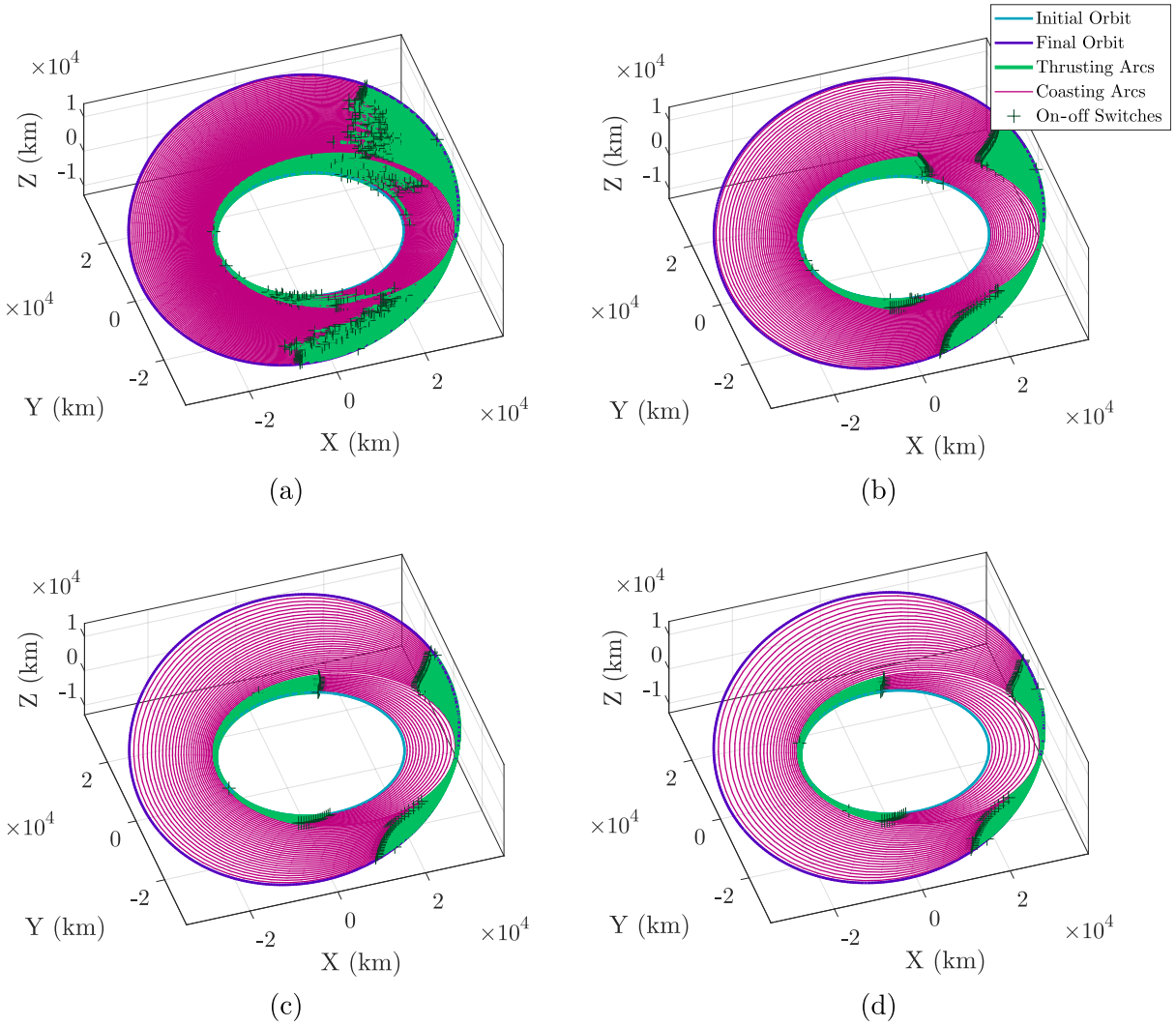

Figure 12: Visualization of transfer trajectories in energy-optimal low-thrust orbital maneuver: (a) $T_{\max} = 0.5N$ (b) $T_{\max} = 1.2N$ (c) $T_{\max} = 1.9N$ (d) $T_{\max} = 2.6N$

The initial spacecraft mass and the specific impulse are considered as $m_{\mathcal{I}} = 970kg$ and $I_{\text{sp}} = 4800s$ respectively. The problem is solved considering four different thrust levels of $T_{\max} \in \{0.5, 1.2, 1.9, 2.6\}N$, and the error margins for this experiment are considered as

$\sigma_a = 10\,km, \sigma_e = 10^{-6}, \sigma_i = 10^{-2^\circ}, \sigma_\Omega = 10^{-3^\circ}, \sigma_\omega = 10^{-2^\circ}$. The 3D visualization of best obtained transfer trajectories are illustrated in Fig. 12.

According to the obtained results, the algorithm achieved optimal transfers in all cases properly, while satisfying the terminal conditions. The increase of thrust arcs and revolutions for lower thrust levels can be identified from the visualization of transfer trajectories. Also, besides the increase in the number of on-off switches, it can be observed that the duration of thrust arcs are also higher in transfers with lower thrust levels. Details of the obtained transfer trajectories are tabulated in Table 7.

Table 7: Best obtained results for energy-optimal transfers ($\epsilon = 1$)

$T_{\max}[N]$	Rev.	$t_f[m]$	$E_a[km]$	E_e	$E_i[deg]$	$E_\Omega[deg]$	$E_\omega[deg]$
2.6	39	29619.26	7.831e-01	5.175e-08	9.809e-05	5.359e-05	1.113e-04
1.9	48	36703.44	1.798e+00	3.283e-07	9.552e-04	5.743e-05	4.458e-04
1.2	67	51508.83	2.079e+00	6.026e-07	1.071e-03	7.688e-05	4.626e-04
0.5	134	102487.38	8.860e+00	7.531e-07	1.286e-03	7.167e-04	4.653e-03

Based on the achieved transfer trajectories, the shortest transfer time is 29619.26 minutes, associated with 39 revolutions for $T_{\max} = 2.6N$. Also, for the thrust level of $T_{\max} = 0.5N$, the desired orbit is reached after 134 revolutions in 102487.38 minutes. The error of orbital elements with respect to the final orbit for each thrust level are also provided in Table 7. Similar to the previous experiment, solutions obtained via the proposed algorithm satisfy the desired conditions based on the error margins. Consequently, higher errors are observed for orbital maneuvers with longer duration of transfer time. Again, it justifies that the problem is more complex with higher difficulty of detecting feasible region in the search space when the thrust level is low.

The time-histories of orbital elements can be analyzed to verify the orbit evolution during the orbital maneuver. Fig. 13 and Fig. 14 show the time histories of five orbital elements correspond to the best obtained solution of the case $T_{\max} = 0.5N$ using the adaptive approach.

As can be appreciated, the obtained transfer trajectory includes a transition point at $t = 28315.5\,m$, where an intermediate orbit is achieved by the algorithm. The orbital elements at this point are as $a = 26433.5\,km, e = 0.4080, i = 3.9625^\circ, \Omega = 262.74^\circ, \omega = 245.88^\circ$. The shape of this intermediate orbit is visible in plot (a) of Fig. 12, where

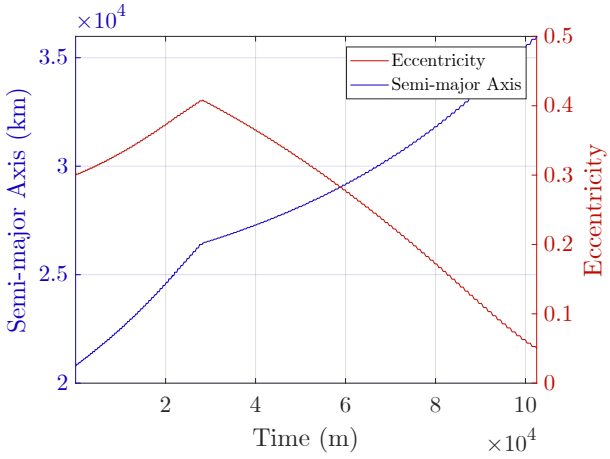

Figure 13: Variation of semi-major axis $a(t)$ and eccentricity $e(t)$ in energy-optimal low-thrust orbital maneuver for $T_{\max} = 0.5N$

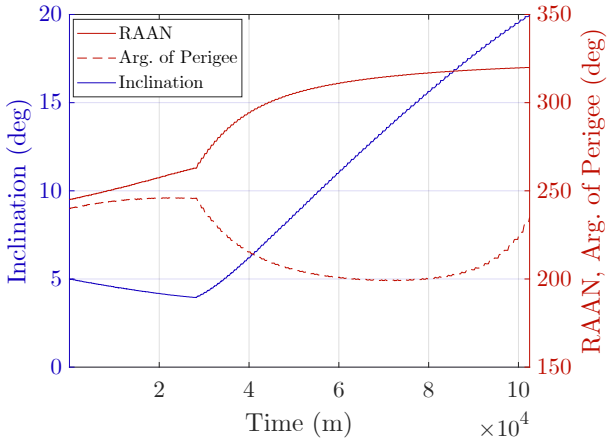

Figure 14: Variation of inclination $i(t)$, RAAN $\Omega(t)$, and argument of perigee $\omega(t)$ in energy-optimal low-thrust orbital maneuver for $T_{\max} = 0.5N$

the trajectory has an intersection with the initial orbit and another intersection with the desired orbit. This transition orbit corresponds to the impulsive solution, obtained via the employed method in [27], which is used for estimating the number of thrust arcs. It shows that the algorithm properly takes advantage of the multi-impulse transfer solution for achieving the optimal maneuver. It is noteworthy that the two-impulse solution of this transfer includes an intermediate orbit with $a = 26452.775\,km, e = 0.408174, i = 3.9488°, \Omega = 263.0025°$, and $\omega = 245.7072°$, which is very close to the one found by the proposed algorithm. Likewise, it implies that the employed multi-impulse approach is a reasonable choice for estimating the number of thrust arcs N_T, reformation of decision variables $(\tau_i^{\mathcal{I}}\ \tau_i^{\mathcal{F}})$, and setting the λ scaling factor.

Variation of thrust direction angles is another aspect of the transfer trajectory, which

gives useful insight regarding the evolution of orbit during the maneuver. Each thrust arc has unique variations for thrust direction angles, and it is possible to visualize the thrust vector acted on the spacecraft for each on-off thrust profile. As a result, 3D representation of thrust vector acting on the spacecraft is illustrated for one of the thrust arcs (revolution 21) in $T_{\max} = 0.5N$ case in Fig. 15.

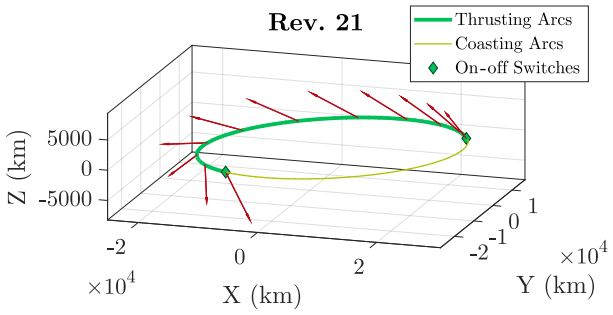

Figure 15: Direction of acting thrust within the revolution 21 of orbit transfer in energy-optimal low-thrust orbital maneuver for $T_{\max} = 0.5N$

Results show high variation of thrust components in this revolution, due to the parameterization nodes obtained by the proposed adaptive approach. The corresponding functions of thrust components are shown in Fig. 16, where the time-histories of direction angles for revolutions 18 to 22 are plotted. It can be observed how five nodes interpolation successfully parameterized the steering angles in each thrust arc. Although, it was possible to consider more discrete nodes to the model, achieving a feasible solution confirms the reasonable choice for the dedicated number of discrete nodes of steering angles parameterization.

Similar to the previous experiment, the performance of the proposed algorithm has been compared with ARMOR-DE and DC3 in an empirical test setup. In this experiment, each algorithm is implemented in 50 independent runs with the same computational budget for every case of $T_{\max}$. The feasibility ratio and relative best percentage of the results are provided in Table 8. As can be observed, unlike the previous experiment, AEDA managed to find feasible solutions in all runs. Also, the best objective values always belong to AEDA, while the performance of EDA++ has shown to be close to AEDA when $T_{\max} = 1.9N$.

Following the conducted experiment, the efficiency and the effectivity of the algorithms are also taken into consideration. Results are extracted, and the quality and the execution time of each run are stored. Two new metrics are used for comparison, including Γ,

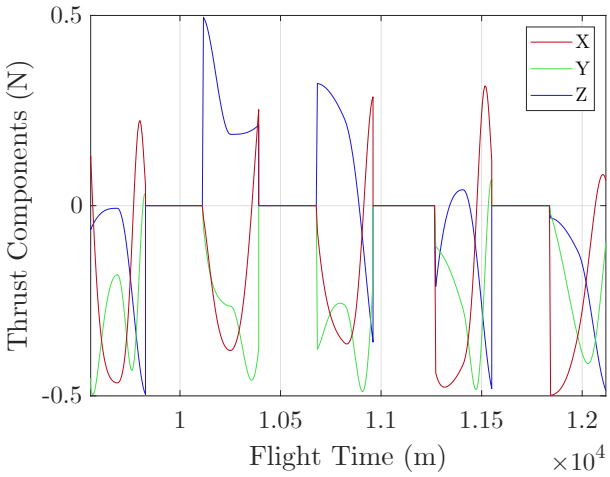

Figure 16: Thrust components during revolutions 18 to 22 of 134 ($9556 < t < 12117$) in energy-optimal low-thrust orbital maneuver for $T_{\max} = 0.5N$

Table 8: Algorithm performance comparison for energy-optimal transfers ($\epsilon = 1$) in 50 runs

$T_{\max}$	AEDA		EDA++		ARMOR-DE		DC3	
	$\mathcal{J}_{\mathcal{G}}$	$\mathcal{J}_{\min}$	$\mathcal{J}_{\mathcal{G}}$	$\mathcal{J}_{\min}$	$\mathcal{J}_{\mathcal{G}}$	$\mathcal{J}_{\min}$	$\mathcal{J}_{\mathcal{G}}$	$\mathcal{J}_{\min}$
2.6	100.00	0.000e+00	20.00	6.334e-05	40.00	8.956e-06	48.00	5.858e-07
1.9	100.00	0.000e+00	94.00	4.451e-05	10.00	2.517e-05	20.00	1.244e-04
1.2	100.00	0.000e+00	34.00	3.678e-05	10.00	2.739e-06	38.00	1.045e-04
0.5	100.00	0.000e+00	4.00	2.334e-04	54.00	4.646e-06	10.00	9.412e-05

representing the efficiency of the algorithm, and α denoting the scaled execution time. For every obtained solution of $\hat{\mathcal{X}}$, the Γ metric is defined as

$$\Gamma(\hat{\mathcal{X}}) = \begin{cases} 1 + \Gamma_c(\hat{\mathcal{X}}) & \Gamma_c(\hat{\mathcal{X}}) > 0 \\ \Gamma_f(\hat{\mathcal{X}}) & \Gamma_c(\hat{\mathcal{X}}) = 0 \end{cases} \tag{50}$$

where Γ_c and Γ_f are the scaled values of objective function and constraint violation as

$$\Gamma_f(\hat{\mathcal{X}}) = \frac{\mathcal{J}(\hat{\mathcal{X}}) - \mathcal{J}_{\min}}{\mathcal{J}_{\max} - \mathcal{J}_{\min}} \tag{51}$$

$$\Gamma_c(\hat{\mathcal{X}}) = \frac{\mathcal{G}(\hat{\mathcal{X}}) - \max(\mathcal{G}_{\min}, 0)}{\mathcal{G}_{\max} - \max(\mathcal{G}_{\min}, 0)} \tag{52}$$

where $\mathcal{J}_{min}$ and $\mathcal{J}_{max}$ are the minimum and maximum objective values found by any of the algorithms. Similarly, $\mathcal{G}_{min}$ and $\mathcal{G}_{max}$ are the lowest and highest constraint violation achieved by any algorithms for each thrust level. The defined parameters scale the objective score Γ_f and constraint score Γ_c in the interval of 0 and 1 for each solution. Having

these scores, the efficiency score Γ will be a score within the interval of $0 \leq \Gamma \leq 2$. In this regard, all feasible and infeasible solutions will be inside the interval of $0 \leq \Gamma \leq 1$ and $1 \leq \Gamma \leq 2$ respectively. Obviously, $\Gamma = 0$ means that the solution is feasible with the best objective value found. If $0 < \Gamma < 1$, it shows that the solution is feasible, but it is not the best solution found in terms of the objective value. If $\Gamma = 1$, it indicates that the solution is feasible (or almost feasible) with the worst objective value in comparison to other obtained feasible solutions. If $1 < \Gamma < 2$, it shows that the solution is infeasible with constraint violation less than the worst solution that has been found. Finally, $\Gamma = 2$ indicates that the solution is infeasible, and it has the highest amount of constraint violation. The scaled execution time $\alpha(\hat{\mathcal{X}})$ is also defined as

$$\alpha(\hat{\mathcal{X}}) = \frac{T_\alpha(\hat{\mathcal{X}}) - T_{\alpha\,\min}}{T_{\alpha\,\max} - T_{\alpha\,\min}} \tag{53}$$

where $T_\alpha(\hat{\mathcal{X}})$ represents the execution time in obtaining the solution $\hat{\mathcal{X}}$, and $T_{\alpha\,\min}$ and $T_{\alpha\,\max}$ are the lowest and highest execution times between all algorithms for the corresponding thrust level. Similarly, the execution time of any run is scaled within the range of $0 < \alpha < 1$, with $\alpha = 0$ representing the fastest run, while $\alpha = 1$ corresponds to the longest execution time. Having the Γ and α metrics for the efficiency of the algorithms in each run, the distribution of scores for these parameters represent the overall performance of the algorithms. Fig. 17 shows this representation for $T_{\max} = 0.5N$ case.

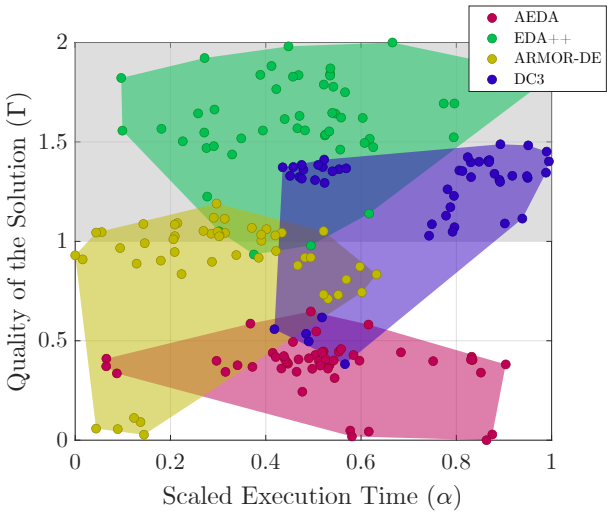

Figure 17: Algorithm efficiency for energy-optimal transfer ($\epsilon = 1$) with $T_{\max} = 0.5N$

According to the provided visualization, each point possesses the scaled execution time α and the scaled score Γ, representing the quality of the obtained solution. The

points for each algorithm are separated with a unique color, and the plotted polygons contains all the points for every individual algorithm. The relative position of polygons indicate the comparative performance of each algorithm. As the polygon is closer to $\alpha = 0$ and $\Gamma = 0$, the algorithm performs faster and better (in terms of the quality of the obtained solution) respectively. As a note, the $\Gamma = 1$ line separates the feasible and infeasible solutions.

According to Fig. 17, the non-adaptive EDA++ has very poor performance in terms of quality, with only 2 feasible solutions out of 50 runs. DC3 has better performance in terms of quality, yet with relatively longer execution times. ARMOR-DE with the fastest performance, gives relatively better results than DC3 and EDA++. However, the average quality of the solutions found by AEDA is generally better than ARMOR-DE. Although the fastest run belongs to ARMOR-DE in this scenario, the solution with the highest quality is achieved by AEDA. Comparing the relative position of polygons for EDA++ and AEDA confirms very high improvement of the adaptive strategy AEDA over the non-adaptive algorithm EDA++, which makes it outperform both rival algorithms with almost the same execution times as in EDA++. Similar analysis can be performed for $T_{\max} = 1.2N$ case, as illustrated in Fig. 18.

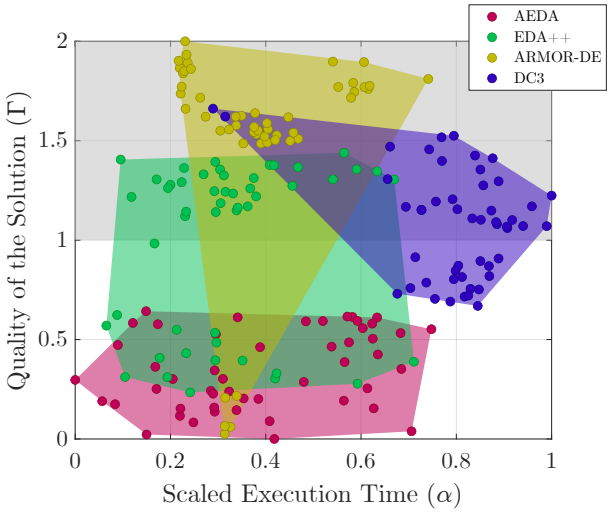

Figure 18: Algorithm efficiency for energy-optimal transfer ($\epsilon = 1$) with $T_{\max} = 1.2N$

The distribution of the execution times for AEDA, EDA++, and ARMOR-DE do not show any noticeable advantages, and the longest execution time belongs to DC3 in this case. However, the quality of the obtained solutions by EDA++ is already better than ARMOR-DE and DC3, since the number of obtained feasible solutions are higher.

As can be seen for ARMOR-DE, except for five very high quality solutions, the rest of the achieved solutions are infeasible. Positions of the polygons of AEDA and EDA++ indicates that the employment of AEDA has improved the performance of EDA++ even further, since all runs of AEDA ended up in feasible solutions. The performance of the algorithms for $T_{\max} = 1.9N$ is provided in Fig. 19.

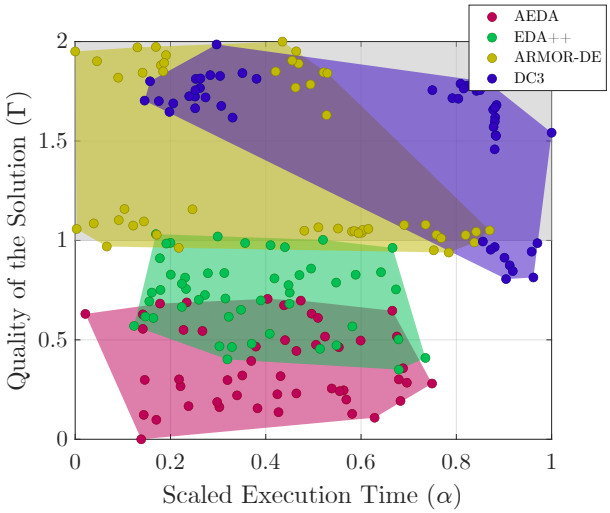

Figure 19: Algorithm efficiency for energy-optimal transfer ($\epsilon = 1$) with $T_{\max} = 1.9N$

In this case, it can be observed that the majority of the obtained solutions by ARMOR-DE and DC3 are infeasible, since the large portion of the corresponding polygons are placed in $\Gamma > 1$ region. Similar to the previous case, the EDA++ already has a better performance relative to ARMOR-DE and DC3 with only 3 infeasible solutions. However, using the adaptive approach, AEDA gives solutions with higher quality with almost the same scores for the execution time. Finally, Fig. 20 shows the relative performance of the algorithms for $T_{\max} = 2.6N$.

In this case, ARMOR-DE has shown to have the least execution times, while DC3 generally takes longer time for finding the optimal solutions. In terms of the quality of the solutions, both ARMOR-DE and DC3 algorithms outperform EDA++. However, the proposed adaptive approach significantly enhances the algorithm since the polygon of AEDA has the best position as it corresponds to the highest quality of solutions. Overall, the comparison of the performance of the algorithms shows that the proposed adaptive AEDA approach considerably improves the performance of the non-adaptive EDA++ in terms of quality without much execution time burden. Following the results provided in Fig. 17 to Fig. 20, AEDA is the only algorithm that achieves feasible solutions in all

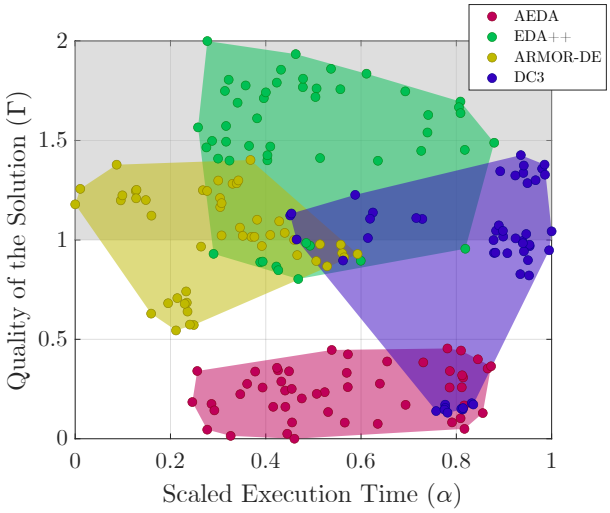

Figure 20: Algorithm efficiency for energy-optimal transfer ($\epsilon = 1$) with $T_{\max} = 2.6N$

runs. It can also be observed that without using the adaptive approach, the algorithm EDA++ does not always have better performance in terms of quality in comparison to ARMOR-DE and DC3 (specifically in cases $T_{\max} = 0.5N$ and $T_{\max} = 2.6N$). However, the improvement due to the proposed adaptive mechanisms makes AEDA competitive versus its rival methods, in terms of the quality of the obtained solutions.

Following the achieved results from the conducted experiments, several insights can be obtained regarding the performance of the proposed approach compared with other methods. These findings can be inferred by comparing the effectiveness and the efficiency of the algorithms in terms of the quality of the obtained solutions and the execution times. The first insight is the enhancement of EDA++ via the proposed adaptive operators, which makes the new algorithm (AEDA) provide solutions with higher quality in terms of the feasibility and the objective value. The main reason for this outcome lies upon the workflow of the proposed adaptive operators, in which the presented adaptive schemes for the truncation factor and the constraints separation threshold guide the optimization process in a promising manner to discover feasible transfer trajectories. Clearly the non-adaptive algorithm (EDA++) does not benefit from this workflow and therefore, the optimization process is less effective in efficient exploration of the search space. Another observation is that the adaptive approach has the potentiality of outperforming recently developed algorithms (ARMOR-DE and DC3) in the discovery of the optimal transfer trajectories. It is noteworthy that the two selected rival algorithms are constrained continuous optimization algorithms. However, the optimization process in these techniques

is not based on probabilistic models. The reason for relatively poor performance of these algorithms in comparison to AEDA can be justified via the fact that no adaptations are employed in these algorithms. In the other word, the search process does not take advantage of the information acquired regarding the complexity of the search domain. Due to the fact that the exploration and exploitation capabilities of these algorithms are not affected by the difficulty of the optimization problem, the probability of achieving high quality feasible solutions are less in comparison to the adaptive approach.

5. Conclusions

The proposed adaptive operators in this research have shown to be a reliable enhancement towards the implementation of EDA framework in spacecraft low-thrust trajectory optimization. The obtained results indicate that the presented approach based on adaptive behavior of the truncation factor and the constraints separation threshold manages to find high quality feasible solutions. In terms of the feasibility ratio of the obtained results, the proposed algorithm gives feasible solutions in more than %70 of the runs considering various thrust levels in the first experiment. The most competitive algorithm in this regard is the non-adaptive EDA++. The other two algorithms, ARMOR-DE and DC3, have also shown to be competitive, but only when the thrust level is high (i.e., problems with relatively lower number of decision variables). Analysis of the relative best scores indicate that except for the case of $T_{\max} = 1.5N$, the best solutions have been achieved by AEDA. Nevertheless, the obtained result for this case is extremely close to the global optimal solution that has been found by EDA++. Analysis of the results in the second experiment shows that the optimization process in the proposed algorithm never ends up in an infeasible solution as the feasibility ratio of AEDA is %100 of the runs considering various thrust levels. The question regarding the amount of improvement that AEDA possesses over the non-adaptive version has shown to be a problem-dependent inquiry. Following the distribution of the obtained results, it has been observed that the most noticeable improvement is achieved in the case of $T_{\max} = 2.6N$. The computational complexity of the proposed algorithm has shown to be the main barrier in this framework, which makes the optimization process take more time to adapt the operators. In this regard, ARMOR-DE is the most challenging algorithm as it has relatively smaller execution times in comparison to AEDA. However, observation on the quality of the obtained

solutions indicates that the longer execution time is a fair exchange for having higher feasibility ratio and solutions with higher optimality.

Many aspects of the proposed approach can be considered as a potential subject for future research. On of the key elements of the presented framework is the definition of the problem identifiers. The proposed identifiers in this research were mainly based on the desired changes in the shape and orientation of the orbit along with the accessible acceleration. Deep analysis of the performance of the algorithm by altering the current problem identifiers, or developing new identifiers is a promising subject for future research. Also, the adaptive approach in this research was aimed at the seeding mechanism and the selection mechanism of EDA++. The door has been left open to target other mechanisms of EDA++ in developing novel operators for optimization. The selection of the dispersion metric is another factor that can be further evaluated. As this is the first study to utilize FLA techniques in spacecraft trajectory optimization, it is unclear whether dispersion was the best option for developing adaptive operators within the proposed direct approach. Other FLA metrics, such as fitness distance correlation (FDC), length scale (LS), and fitness cloud (FL), may also be used to identify optimal transfer trajectories [30]. However, it is important to note that each metric has its own limitations and constraints. For instance, FDC necessitates the availability of the global optimal solution, which is usually unknown in the majority of spacecraft trajectory optimization problems, making it unsuitable.

Acknowledgments

This research is supported by the Basque Government through the BERC 2022-2025 program, the Ministry of Science, Innovation and Universities: BCAM Severo Ochoa accreditation SEV-2017-0718, BEAZ Bizkaia 3/12/DP/2021/00150, and SPRI Group through Ekintzaile Program EK-00112-2021.

References

[1] John E Prussing. Optimal four-impulse fixed-time rendezvous in the vicinity of a circular orbit. *AIAA Journal*, 7(5):928–935, 1969.

[2] J-P Marec. Optimal space trajectories. *NASA STI/Recon Technical Report A*, 80:48848, 1979.

[3] Theodore N. Edelbaum. Optimal space trajectories. Technical report, Analytical Mechanics Associates Inc, Jericho, NY, 1969.

[4] Abolfazl Shirazi, Josu Ceberio, and Jose A Lozano. Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions. *Progress in Aerospace Sciences*, 102:76–98, 2018.

[5] Xiaoyu Chu, Zongchuang Liang, and Yanyan Li. Trajectory optimization for rendezvous with bearing-only tracking. *Acta Astronautica*, 171:311–322, 2020.

[6] Zichen Fan, Mingying Huo, Naiming Qi, Ye Xu, and Zhiguo Song. Fast preliminary design of low-thrust trajectories for multi-asteroid exploration. *Aerospace Science and Technology*, 93:105295, 2019.

[7] Vishala Arya, Ehsan Taheri, and John L Junkins. A composite framework for co-optimization of spacecraft trajectory and propulsion system. *Acta Astronautica*, 178:773–782, 2021.

[8] Di Wu, Lin Cheng, Fanghua Jiang, and Junfeng Li. Rapid generation of low-thrust many-revolution earth-center trajectories based on analytical state-based control. *Acta Astronautica*, 187:338–347, 2021.

[9] Mingcheng Zuo, Guangming Dai, Lei Peng, Maocai Wang, Zhengquan Liu, and Changchun Chen. A case learning-based differential evolution algorithm for global optimization of interplanetary trajectory design. *Applied Soft Computing*, 94:106451, 2020.

[10] Sanaz Samsam and Robin Chhabra. Multi-impulse shape-based trajectory optimization for target chasing in on-orbit servicing missions. In *2021 IEEE Aerospace Conference (50100)*, pages 1–11. IEEE, 2021.

[11] Andrea Caruso, Alessandro A Quarta, Giovanni Mengali, and Matteo Ceriotti. Shape-based approach for solar sail trajectory optimization. *Aerospace Science and Technology*, 107:106363, 2020.

[12] Marco Pallone, Mauro Pontani, and Paolo Teofilatto. Performance evaluation methodology for multistage launch vehicles with high-fidelity modeling. *Acta Astronautica*, 151:522–531, 2018.

[13] Haiyang Li, Shiyu Chen, Dario Izzo, and Hexi Baoyin. Deep networks as approximators of optimal low-thrust and multi-impulse cost in multitarget missions. *Acta Astronautica*, 166:469–481, 2020.

[14] Ruida Xie and Andrew G Dempster. An on-line deep learning framework for low-thrust trajectory optimisation. *Aerospace Science and Technology*, 118:107002, 2021.

[15] Yang Zhou, Ye Yan, Xu Huang, and Linjie Kong. Mission planning optimization for multiple geosynchronous satellites refueling. *Advances in Space Research*, 56(11):2612–2625, 2015.

[16] Jose A Lozano, Pedro Larrañaga, Endika Bengoetxea, and Iñaki Inza. *Towards a new evolutionary computation: advances on estimation of distribution algorithms*, volume 192. Springer Science & Business Media, 2006.

[17] Edmondo Minisci and Giulio Avanzini. Comparative study on the application of evolutionary optimization techniques to orbit transfer manoeuvres. In *59th International Astronautical Congress*, pages 1–15, 2008.

[18] Francesco Marchetti, Edmondo Minisci, and Annalisa Riccardi. Single-stage to orbit ascent trajectory optimisation with reliable evolutionary initial guess. *Optimization and Engineering*, pages 1–26, 2021.

[19] Abolfazl Shirazi, Josu Ceberio, and Jose A Lozano. EDA++: Estimation of distribution algorithms with feasibility conserving mechanisms for constrained continuous optimization. *IEEE Transactions on Evolutionary Computation*, 2022.

[20] Feng Zou, Debao Chen, Hui Liu, Siyu Cao, Xuying Ji, and Yan Zhang. A survey of fitness landscape analysis for optimization. *Neurocomputing*, 503:129–139, 2022.

[21] Yaxin Li, Jing Liang, Kunjie Yu, Ke Chen, Yinan Guo, Caitong Yue, and Leiyu Zhang. Adaptive local landscape feature vector for problem classification and algorithm selection. *Applied Soft Computing*, page 109751, 2022.

[22] Jin Haeng Choi and Chandeok Park. Spacecraft trajectory optimizations: Metrics for fitness landscape analysis. In *AIAA SCITECH 2022 Forum*, page 1891, 2022.

[23] D Izzo, T Vinkó, and M del Rey Zapatero. Gtop database: Global optimisation trajectory problems and solutions, 2010.

[24] Wenyin Gong, Zhihua Cai, and Dingwen Liang. Adaptive ranking mutation operator based differential evolution for constrained optimization. *IEEE transactions on cybernetics*, 45(4):716–727, 2014.

[25] Priya Donti, David Rolnick, and J Zico Kolter. Dc3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations*, 2021.

[26] Dilcia Pérez and Yamilet Quintana. A survey on the weierstrass approximation theorem. *arXiv preprint math/0611038*, 2006.

[27] Abolfazl Shirazi, Josu Ceberio, and Jose A Lozano. An evolutionary discretized lambert approach for optimal long-range rendezvous considering impulse limit. *Aerospace Science and Technology*, 94:105400, 2019.

[28] Régis Bertrand and Richard Epenoy. New smoothing techniques for solving bang–bang optimal control problems—numerical results and statistical interpretation. *Optimal Control Applications and Methods*, 23(4):171–197, 2002.

[29] Fanghua Jiang, Hexi Baoyin, and Junfeng Li. Practical techniques for low-thrust trajectory optimization with homotopic approach. *Journal of guidance, control, and dynamics*, 35(1):245–258, 2012.

[30] Yuan Sun, Saman K Halgamuge, Michael Kirley, and Mario A Munoz. On the selection of fitness landscape analysis metrics for continuous optimization problems. In *7th International Conference on Information and Automation for Sustainability*, pages 1–6. IEEE, 2014.

[31] Zhiping Tan and Kangshun Li. Differential evolution with mixed mutation strategy based on deep reinforcement learning. *Applied Soft Computing*, 111:107678, 2021.

[32] Monte Lunacek and Darrell Whitley. The dispersion metric and the cma evolution strategy. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 477–484, 2006.

[33] Augusto Dantas and Aurora Pozo. On the use of fitness landscape features in meta-learning based algorithm selection for the quadratic assignment problem. *Theoretical Computer Science*, 805:62–75, 2020.

[34] Rachael Morgan and Marcus Gallagher. Sampling techniques and distance metrics in high dimensional continuous landscape analysis: Limitations and improvements. *IEEE Transactions on Evolutionary Computation*, 18(3):456–461, 2013.