Optimization and Visualization in Many-Objective Space Trajectory Design

Hernán Aguirre, Kiyoshi Tanaka, Tea Tušar and Bogdan Filipič

Abstract This work optimizes the thrusting profile of a low-thrust spacecraft propelled by an ion engine to raise from Earth's low orbit to the vicinity of the Moon. The orbital raising phase is divided uniformly into sixteen sections, of which the first six are set to full propagation to escape early from the radiation belts, and the profiles of the other ten sections are subject to optimization together with the propagation start date and the spacecraft's initial mass. Each section is defined by three variables. Thus, the optimization problem consists of thirty-two variables. Four objective functions are considered, namely the operation time of the ion engine system, time to reach the Moon, maximum eclipse time, and the initial mass of the spacecraft, subject to various constraints. We use the many-objective optimizer named Adaptive ε-Sampling and ε-Hood (AεSεH) to search for non-dominated solutions, analyze the trade-offs between variables and objectives, and use a method called visualization with prosections to gain insights into the problem and to analyze the dynamics of the optimization algorithm.

H. Aguirre (✉) · K. Tanaka
Faculty of Engineering, Shinshu University, 4-17-1 Wakasato, Nagano 380-8553, Japan
e-mail: ahernan@shinshu-u.ac.jp

K. Tanaka
e-mail: ktanaka@shinshu-u.ac.jp

T. Tušar · B. Filipič (✉)
Department of Intelligent Systems, Jožef Stefan Institute, Jamova cesta 39,
SI-1000 Ljubljana, Slovenia
e-mail: bogdan.filipic@ijs.si

T. Tušar
e-mail: tea.tusar@ijs.si

T. Bartz-Beielstein et al. (eds.), *High-Performance Simulation-Based Optimization*, Studies in Computational Intelligence 833,
https://doi.org/10.1007/978-3-030-18764-4_5

1 Introduction

The design of space exploration missions based on low-thrust spacecraft propelled by ion engines is increasingly attracting attention. Ion propulsion systems efficiently use fuel and electrical power to enable modern spacecraft to travel at higher velocity and lower costs than other propulsion technologies currently available. There are several in-space propulsion applications for ion engines. The best applications make use of the long mission interval when significant thrust is not needed.

DESTINY (Demonstration and Experiment of Space Technology for INterplanetary voYage) is a candidate mission of the Japan Aerospace Exploration Agency (JAXA) that aims to validate the advanced ion propulsion technology for future deep space missions [9]. The DESTINY spacecraft is equipped with ultra-lightweight solar panels and propelled by a low-thrust ion engine. The mission consists of several phases. Firstly, DESTINY is launched by an Epsilon rocket [11] and positioned into a low elliptical orbit around the Earth. Then, the spacecraft spirals up to the vicinity of the Moon using the ion propulsion system. We call this the propagation phase. DESTINY is subsequently injected into the L_2 halo orbit [5] of the Sun-Earth system by using the gravitational pull of the Moon. DESTINY conducts its engineering experiment and scientific observations at the L_2 halo orbit for at least half a year, after which it continues to its next destination if conditions permit. The mission is schematically shown in Fig. 1.

We focus on the optimization of the thrusting profile of the ion engine to propagate the DESTINY spacecraft from Earth's low orbit to the vicinity of the Moon. The formulation of the problem to optimize the thrusting profile of the ion engine has evolved as the designers have gained knowledge about the problem, resulting in a more detailed modeling of the orbital raising phase. Since the beginning, however, the problem was considered a many-objective optimization problem with up to six objective functions.

Initial formulations of the optimization problem considered a two-stage orbital raising approach for the propagation phase to the Moon, where the spacecraft would first raise only during the perigee and later switch to raise during the apogee. Thus, the initial formulations considered up to five variables, including propagation start date, switch date from the perigee raising to the apogee raising, range for apogee raising, range for perigee raising, and the initial mass of the spacecraft [7, 12]. Six objective functions were considered in [7], including the operation time of the ion engine system, time to reach the Moon, maximum eclipse time (for the spacecraft passing the shadow of the Earth), time to pass the inner radiation belt (at the altitude of 5,000 km), time to pass the outer radiation belt (at the altitute of 20,000 km), and the initial mass of the spacecraft. Similarly, four- and five-objective formulations were considered in [12], fixing the initial mass of the spacecraft to 400 kg, i.e., the first four or five objectives used in [7].

Latest formulations divide the orbital raising phase into several sections, each specified by three variables or parameters, seeking to optimize the engine thrusting profile for each section. In [18], the formulation divides the time of the orbital raising

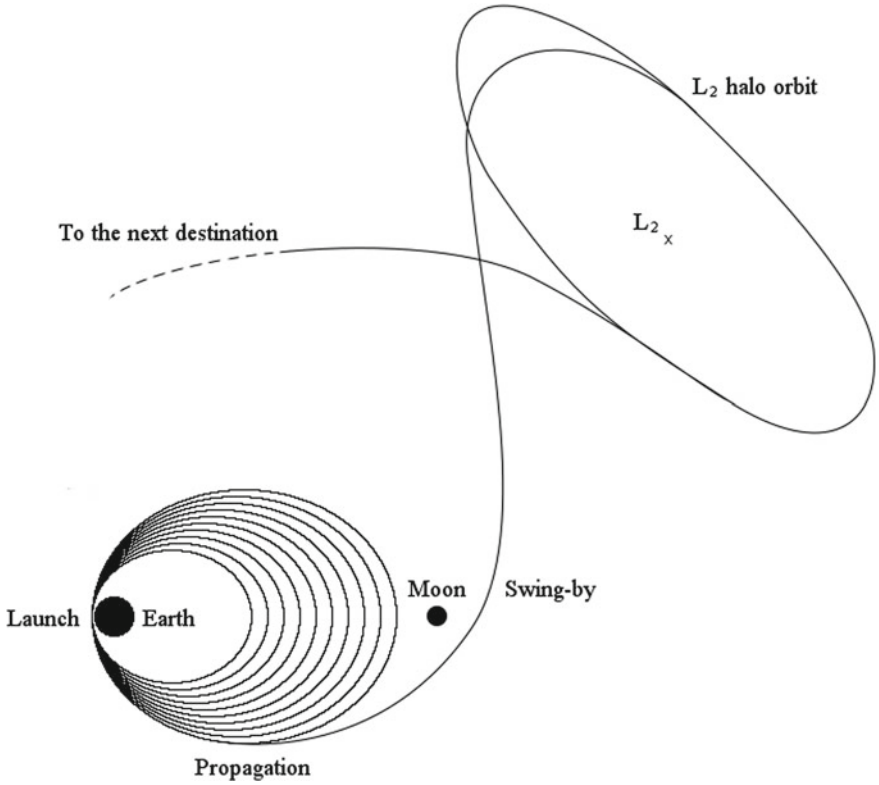


Fig. 1 A schematic view of the DESTINY mission (reproduced after [8])

phase into eight sections, optimizing their parameters while minimizing the time of flight, the operation time of the ion engine system, and the time to pass the radiation belt. The initial mass was fixed to 400 kg and maximum eclipse time was not considered for minimization. In [17], the orbital raising phase is divided into ten sections, setting the first three to full propagation to pass the radiation belts as soon as possible and optimizing the profiles of the other eight sections with four objective functions, minimizing the operation time of the ion engine system, time to reach the Moon, maximum eclipse time, and maximizing the initial mass of the spacecraft.

In this work, we use a formulation that divides the orbital raising phase into 16 sections, setting the first six to full propagation and optimizing the profiles of the other ten sections. Similarly to [17], we consider four objectives: the operation time of the ion engine system, time to reach the Moon, maximum eclipse time, and the initial mass of the spacecraft. We use AεSεH [1], a many-objective optimizer, to search for Pareto optimal sets of solutions and analyze the trade-offs between variables and objectives.

To gain insight into the workings of an algorithm as well as discover properties of the optimization problem at hand, visualization of algorithm results can be

used. In many-objective optimization, visualization efforts are usually aimed at representing solutions in the objective space. Because projections to lower-dimensional spaces generally cause loss of information, this is a nontrivial task when optimization problems have more than three objectives. In this work, we use a method called visualization with prosections [15] to visualize the optimization results and analyze the optimization algorithm performance.

This chapter is organized as follows. In Sect. 2 we present the formulation of the optimization problem, in Sect. 3 we describe the applied many-objective evolutionary algorithm, and in Sect. 4, we introduce visualization in many-objective optimization and, in particular, the method of visualization with prosections. In Sect. 5 we discuss in detail some key optimization results and illustrate the use of visualization with prosections in analyzing the outcome of the optimization and the dynamics of the algorithm. We conclude with Sect. 6, summarizing the work done and the findings, and presenting a plan for future work.

2 DESTINY Spacecraft Trajectory Design Problem

The DESTINY spacecraft will be launched in an Epsilon rocket, released into a low elliptical orbit, and start a propagation stage to spiral away from the Earth towards the Moon propelled by a low-thrust Ion Engine System (IES). The spacecraft must reach the orbit of the Moon in at most 1.5 years (approx. 550 days or 13,000 h), thus one evaluation criterion is the time to reach the Moon. The IES is solar-powered and requires the use of a battery when the spacecraft is under the shadow of the Earth. Thus, it is important to minimize the eclipse time in order to reduce the size and weight of the required battery. Based on previous knowledge, in this work we constrain the maximum eclipse time to 2 h. In addition, the IES operation time should be minimized to reduce fuel consumption. Another evaluation criterion is the initial mass of the spacecraft, which should be maximized to allow as much equipment for experiments as possible. Further, the spacecraft must escape as soon as possible from the inner (5,000 km) and outer (20,000 km) radiation belts surrounding the Earth, since radiation can damage the solar panels and digital equipment.

The problem we are solving in this work is formulated as follows:

$$
\begin{aligned}
&\text{optimize} \quad \boldsymbol{f}(\boldsymbol{v}) = (f_1(\boldsymbol{v}), f_2(\boldsymbol{v}), f_3(\boldsymbol{v}), f_4(\boldsymbol{v})),\\
&\text{subject to}\\
&\qquad \boldsymbol{v} = (d, \boldsymbol{p}, m),\\
&\qquad \boldsymbol{p} = ((x_1, y_1, z_1), \ldots, (x_{16}, y_{16}, z_{16})),\\
&\qquad d \in [0, 1],\\
&\qquad x_i, y_i, z_i \in [0, 1],\ 1 \leq i \leq 16,\\
&\qquad 400\ \text{kg} \leq m \leq 450\ \text{kg},\\
&\qquad f_2 \leq 1.5\ \text{years (approx. 550 days or 13,000 h)},\\
&\qquad f_3 \leq 2\ \text{h},\\
&\qquad f_4 = m,
\end{aligned}
\tag{1}
$$

where three objective functions (IES operation time f_1, time to reach the Moon f_2, and maximum eclipse time f_3) are to be minimized, whereas the fourth one (initial mass of the spacecraft f_4) is to be maximized. The decision variables are the propagation start date d, the IES operation profile during the raising phase to the Moon $\boldsymbol{p}$, and the initial mass of the spacecraft m. To compute the launch date, d is linearly mapped from the range [0,1] to date in days and time in seconds. The launch date is then given by 2019-07-01 00:00:00 + date + time in UTC. The 550 days (1.5 years) of orbital raising are uniformly divided into 16 sections. The first six sections $(x_1, y_1, z_1), \ldots, (x_6, y_6, z_6)$ are set to full throttle to escape from the radiation belts as quickly as possible.

In this work, we optimize the IES operation profile for the other ten sections $(x_7, y_7, z_7), \ldots, (x_{16}, y_{16}, z_{16})$, in addition to d and m. That is, the optimization problem consists of 32 variables. In each section, the IES operation profile is given by the apogee thrusting arc ΔL_a, the perigee thrusting arc ΔL_p, and the offset angle η. These three parameters are expressed in degrees. The arcs given by $180 + \eta - |\Delta L_\mathrm{a}| \leq \theta \leq 180 + \eta + |\Delta L_\mathrm{a}|$ and $\eta - |\Delta L_\mathrm{p}| \leq \theta \leq \eta + |\Delta L_\mathrm{p}|$ are set to operate the IES. Otherwise, the IES is coasting. When $\Delta L_\mathrm{a} > 0$ and/or $\Delta L_\mathrm{p} > 0$, the IES is thrusting in the corresponding arc(s). Similarly, when $\Delta L_\mathrm{a} < 0$ and/or $\Delta L_\mathrm{p} < 0$, the IES is thrusting in reverse in the corresponding arc(s). The parameters optimized by the evolutionary algorithm are expressed in a (x, y, z) coordinate system. Given a vector in (x, y, z), it is converted to the corresponding value $(\Delta L_\mathrm{a}, \Delta L_\mathrm{p}, \eta)$ to determine the IES profile. Further details about handling the coordinates can be found in [17].

The initial mass of the spacecraft is multiplied by -1, so all functions are minimized. To handle constraints and preferences, the fitness values of the solutions are penalized as follows:

$$f_i'(\boldsymbol{v}) = f_i(\boldsymbol{v}) + \alpha \max(f_i), \tag{2}$$

where $\alpha = 1$ if time to reach the Moon f_2 is larger than 1500 days, $\alpha = 2$ if maximum eclipse time f_3 is larger than 2 h, $\alpha = 3$ if the spacecraft does not reach the vicinity of the Moon, and $\max(f_i)$ is the maximum observed value for fitness function f_i.

3 The Adaptive ε-Sampling and ε-Hood Algorithm

Adaptive ε-Sampling and ε-Hood (AεSεH) [1] is an elitist evolutionary many-objective algorithm that applies ε-dominance principles for parent selection and survivor selection. In ϵ-dominance [10], the objective vector $\boldsymbol{f}(\boldsymbol{x})$ of a solution $\boldsymbol{x}$ is first mapped to another point $\boldsymbol{f}'(\boldsymbol{x})$ in the objective space and dominance is calculated using the mapped point. Let us consider, without loss of generality, a minimization multi-objective problem with M objectives $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x}))$. A solution $\boldsymbol{x}$ is said to ϵ-dominate another solution $\boldsymbol{y}$, denoted by $\boldsymbol{x} \prec^\epsilon \boldsymbol{y}$, if the following conditions are satisfied:

$$\begin{aligned} &\boldsymbol{f}(\boldsymbol{x}) \mapsto^{\epsilon} \boldsymbol{f}'(\boldsymbol{x}) \\ &\forall i \in \{1,\ldots,M\} \quad f_i'(\boldsymbol{x}) \leq f_i(\boldsymbol{y}) \quad \wedge \\ &\exists i \in \{1,\ldots,M\} \quad f_i'(\boldsymbol{x}) < f_i(\boldsymbol{y}), \end{aligned} \tag{3}$$

where $\boldsymbol{f}(\boldsymbol{x}) \mapsto^{\epsilon} \boldsymbol{f}'(\boldsymbol{x})$ is a mapping function that depends on parameter ϵ.

The general flow of A$\boldsymbol{\varepsilon}$S$\boldsymbol{\varepsilon}$H is illustrated in Algorithm 3. The main steps and characteristics of the algorithm are explained in further subsections.

Algorithm 3 A$\boldsymbol{\varepsilon}$S$\boldsymbol{\varepsilon}$H

Input: population size P_{size}, reference neighborhood size $H_{\text{size}}^{\text{Ref}}$, initial adaptation step Δ_0
Output: $\mathcal{F}_1$, set of non-dominated solutions

$N_{\text{h}}^{\text{Ref}} \leftarrow P_{\text{size}}/H_{\text{size}}^{\text{Ref}}$ // set reference number of neighborhoods
$\varepsilon_{\text{s}} \leftarrow 0$, $\Delta_{\text{s}} \leftarrow \Delta_0$ // set ε_{s}-dominance factor and its adaptation step
$\varepsilon_{\text{h}} \leftarrow 0$, $\Delta_{\text{h}} \leftarrow \Delta_0$ // set ε_{h}-dominance factor and its adaptation step
$\mathcal{P} \leftarrow random$, $\mathcal{Q} \leftarrow \emptyset$ // initial populations $\mathcal{P}$ and $\mathcal{Q}$, $|\mathcal{P}| = P_{\text{size}}$
evaluation($\mathcal{P}$)
non-dominated sorting($\mathcal{P}$)
repeat
  // Parent selection
  $\{\mathcal{H}, N_{\text{h}}\} \leftarrow \epsilon$-hood creation $(\mathcal{P}, \varepsilon_{\text{h}})$ // $\mathcal{H} = \{\mathcal{H}_j\},\ j = 1, 2, \ldots, N_{\text{h}}$
  $\{\varepsilon_{\text{h}}, \Delta_{\text{h}}\} \leftarrow$ adapt $(\varepsilon_{\text{h}}, \Delta_{\text{h}}, N_{\text{h}}^{\text{Ref}}, N_{\text{h}})$
  $\mathcal{P}' \leftarrow \epsilon$-hood mating($\mathcal{H}$, P_{size})
  // Offspring creation
  $\mathcal{Q} \leftarrow$ recombination and mutation($\mathcal{P}'$) // $|\mathcal{Q}| = |\mathcal{P}| = P_{\text{size}}$
  // Evaluation and front sorting
  evaluation($\mathcal{Q}$)
  $\mathcal{F} \leftarrow$ non-dominated sorting($\mathcal{P} \cup \mathcal{Q}$) // $\mathcal{F} = \{\mathcal{F}_i\},\ i = 1, 2, \ldots, N_F$
  // Survivor selection
  $\{\mathcal{P}, N_{\text{s}}\} \leftarrow \epsilon$-sampling truncation($\mathcal{F}$, ε_{s}, P_{size}) // N_{s}, number of samples
  $\{\varepsilon_{\text{s}}, \Delta_{\text{s}}\} \leftarrow$ adapt $(\varepsilon_{\text{s}}, \Delta_{\text{s}}, P_{\text{size}}, N_{\text{s}})$
until termination criterion is met **return** $\mathcal{F}_1$

3.1 Parent Selection

For parent selection the algorithm first uses a procedure called *ε-hood creation* to cluster solutions in the objective space. This procedure randomly selects an individual from the population and applies ε-dominance with parameter ε_{h}. A neighborhood is formed by the selected solution and its ε_{h}-dominated solutions. Neighborhood creation is repeated until all solutions in the population have been assigned to a neighborhood. Then, parents are selected by the *ε-hood mating* procedure, which sees neighborhoods as elements of a list that are visited one at the time in a round-robin schedule. The first two parents are selected randomly from the first neighborhood in the list. The next two parents are selected randomly from the second neighborhood in the list, and so on. When the end of the list is reached, parent selection continues with the first neighborhood in the list. Thus, all individuals have the same probability of

being selected within a specified neighborhood, but due to the round-robin schedule individuals belonging to neighborhoods with fewer members have more reproduction opportunities than those belonging to neighborhoods with more members.

The presented neighborhood creation and random selection within the same neighborhood works well when all solutions in the parent population are non-dominated, which is the common situation in many-objective optimization during most generations of the evolutionary process. In the case that there are dominated solutions in the population, *ε-hood creation* ensures that the solution sampled to create the neighborhood is non-dominated and tournament selection is used to select parents within the neighborhoods. This allows the algorithm to be used in multi- and many-objective optimization [2]. It should be noted that due to the random schedule in which solutions are selected to form the neighborhood, *ε-hood creation* would create different neighborhoods in the next generation even if the population has not changed.

3.2 Offspring Creation, Evaluation and Front Sorting

Once the pool of mates has been formed, recombination and mutation are applied to the selected parent individuals to create the offspring population $\mathcal{Q}_t$. The newly created offspring population is evaluated. Then, the current population $\mathcal{P}_t$ and its offspring $\mathcal{Q}_t$ are joined and divided into non-dominated fronts $\mathcal{F} = \{\mathcal{F}_i\}, i = 1, 2, \ldots, N_F$ using the non-dominated sorting procedure [4, 13].

3.3 Survivor Selection

Next, survivor selection is performed using *ϵ-sampling truncation* which applies two different procedures according to the number of non-dominated solutions. If the number of non-dominated solutions is smaller than the population size, $|\mathcal{F}_1| < P_{\text{size}}$, the sets of solutions $\mathcal{F}_i$ are copied iteratively to $\mathcal{P}_{t+1}$ until it is filled; if set $\mathcal{F}_i$, $i > 1$, overfills $\mathcal{P}_{t+1}$, the required number of solutions are chosen randomly from it. On the other hand, if $|\mathcal{F}_1| > P_{\text{size}}$, it calls *ε-sampling* with parameter ε_s. This procedure samples solutions randomly from the set $\mathcal{F}_1$, inserting the sample in $\mathcal{P}_{t+1}$ and eliminating from $\mathcal{F}_1$ the sample itself and the solutions ε-dominated by the sample. Sampling is repeated until there are no remaining solutions in $\mathcal{F}_1$. After sampling, if $\mathcal{P}_{t+1}$ is overfilled, the solutions are randomly eliminated from it. Otherwise, if there is still room in $\mathcal{P}_{t+1}$, the required number of solutions are randomly chosen from the initially ε-dominated solutions and added to $\mathcal{P}_{t+1}$. This guarantees that the size of $\mathcal{P}_{t+1}$ is exactly P_{size}. Note that due to the random schedule in which solutions are sampled, *ε-sampling* will also choose a different subset of solutions in the next generation even if the first front has not changed.

3.4 Epsilon Mapping Function

The mapping functions $\boldsymbol{f}(\boldsymbol{x}) \mapsto^{\epsilon} \boldsymbol{f}'(\boldsymbol{x})$ used for ε-dominance in *ε-sampling* and *ε-hood creation* determine the distribution of solutions the algorithm aims to find. In this work, we use the following mapping function for both *ε-hood creation* and *ε-sampling*:

$$f_i'(\boldsymbol{x}) = f_i(\boldsymbol{x}) - \varepsilon |\min f_i(\boldsymbol{x}) - \text{median} f_i(\boldsymbol{x})|, \tag{4}$$

where $\varepsilon_s \geq 0$ is used instead of ε in the case of *ε-sampling* and $\varepsilon_h \geq 0$ in the case of *ε-hood creation*. This kind of mapping function works well with functions of either similar or different scales. Note that a minimization problem is considered in the above mapping function.

3.5 Adaptation

Both epsilon parameters ε_s and ε_h used in survivor selection and neighborhood creation, respectively, are dynamically adapted during the run of the algorithm. The adaptation rule, similar for both procedures, is as follows. If $N > Ref$, it increases the adaptation step $\Delta \leftarrow \min(\Delta \times 2, \Delta_{\max})$ and $\epsilon \leftarrow \epsilon + \Delta$. Otherwise, if $N < Ref$, it decreases $\Delta \leftarrow \max(\Delta \times 0.5, \Delta_{\min})$ and $\epsilon \leftarrow \max(\epsilon - \Delta, 0.0)$. In the case of adapting the parameter ε_s used for truncation, the above rule is called with $\varepsilon = \varepsilon_s$, $\Delta = \Delta_s$, $N = N_s$ the number of ε sampled solutions, and $Ref = P_{\text{size}}$ the population size. On the other hand, in the case of the parameter ε_h used for neighborhood creation, the above rule is called with $\varepsilon = \varepsilon_h$, $\Delta = \Delta_h$, $N = N_h$ the number of created neighborhoods, and $Ref = N_h^{\text{Ref}}$ the reference number of neighborhoods.

4 Visualization

In this section, we briefly discuss the state of visualization in many-objective optimization and present visualization with prosections.

4.1 Visualization in Many-Objective Optimization

While many methods specifically designed for visualization in many-objective optimization are continuously being proposed [3, 15], *parallel coordinates* [6] and *scatter plots* (or *the scatter plot matrix*) remain the most often used visualization methods in this field. They are both easy to understand and can visualize the decision space in addition to the objective space, but have some limitations. For example, although

parallel coordinates are scalable to any number of objectives, they become increasingly difficult to interpret when visualizing a large number of solutions. Conversely, a scatter plot matrix can be used to visualize (moderately) large sets, but its scalability in the number of objectives is hindered by the large amount of resulting plots (the scatter plot matrix for M objectives contains $M(M-1)/2$ different plots). Moreover, both methods conceal some properties of solution sets that are important in multi-objective optimization, such as the shape of the front (indispensable for finding knees, i.e., regions with good trade-offs between objectives) and Pareto-dominance relations between solutions (needed when comparing multiple sets).

Contrary to these two methods, *visualization with prosections*, a visualization method introduced in [14] and detailed in [15], is able to partially preserve the front shape, distribution of solutions, and Pareto-dominance relations between solutions, but does so to the expense of visualizing the entire set. It is therefore complimentary to parallel coordinates and scatter plots and can be used in combination with scatter plots to provide additional information on algorithm performance.

In this work, we employ scatter plots and prosections to visualize results by AεSεH, while parallel coordinates are not used due to their disadvantages when visualizing large sets.

4.2 Visualization with Prosections

Prosection, a term describing the *pro*jection of a *section* [16], is a reduction in dimension that visualizes only one portion of solutions, i.e., those that lie in the chosen section. Assume that the objectives f_1 and f_2 have been normalized to $[0, 1]$. In visualization with prosections [15], the section on the plane f_1, f_2 is defined by the angle φ and width d with:

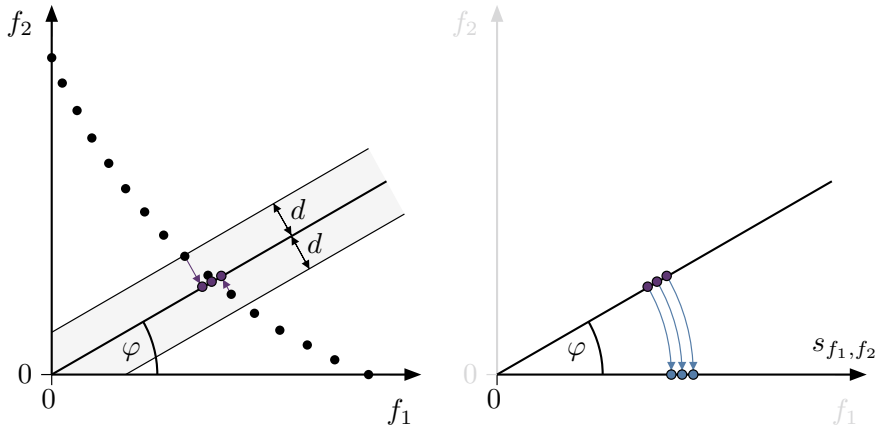
$$|f_1(\boldsymbol{x}) \sin\varphi - f_2(\boldsymbol{x}) \cos\varphi| \leq d. \tag{5}$$

That is, solutions $\boldsymbol{x}$ for which this inequality holds, have at most distance d in the objective space to the line originating in $(0, 0)$ and intersecting the plane f_1, f_2 under the angle φ. All solutions within this section are subject to the following reduction in dimension:

$$s_{f_1,f_2,\varphi,d}(\boldsymbol{x}) = f_1(\boldsymbol{x}) \cos\varphi + f_2(\boldsymbol{x}) \sin\varphi. \tag{6}$$

This is a composition of two functions, an orthogonal projection to the line starting in the origin and intersecting the plane f_1, f_2 under angle φ, and a rotation by $-\varphi$ around the origin as shown in Fig. 2. We will use the short notation s_{f_1,f_2} instead of $s_{f_1,f_2,\varphi,d}$ for readability reasons.

Prosections can be used to visualize solutions from a 4-D objective space in 3-D by applying the transformation from Eq. (6) to two of the objectives, for example f_1 and f_2, while leaving the other two objectives intact:



(a) Orthogonal projection (b) Rotation by $-\varphi$

Fig. 2 The two functions composing the transformation $s_{f_1,f_2,\varphi,d}$: the orthogonal projection of all solutions within the section to the line starting at the origin and intersecting the plane f_1, f_2 under angle φ (**a**), and the rotation by $-\varphi$ around the origin (**b**)

$$(f_1(\boldsymbol{x}),\, f_2(\boldsymbol{x}),\, f_3(\boldsymbol{x}),\, f_4(\boldsymbol{x})) \mapsto (s_{f_1,f_2}(\boldsymbol{x}),\, f_3(\boldsymbol{x}),\, f_4(\boldsymbol{x})). \tag{7}$$

In contrast to other visualization methods, in visualization with prosections we are able to assess the distortions brought by the dimension reduction, which amount to at most $2d \max\{\tan\varphi, \tan^{-1}\varphi\}$ [15]. Therefore, if the prosection of the solution $\boldsymbol{x}$ dominates the prosection of the solution $\boldsymbol{y}$, i.e., if

$$(s_{f_1,f_2}(\boldsymbol{x}),\, f_3(\boldsymbol{x}),\, f_4(\boldsymbol{x})) \prec (s_{f_1,f_2}(\boldsymbol{y}),\, f_3(\boldsymbol{y}),\, f_4(\boldsymbol{y})), \tag{8}$$

and the two solutions are *apart enough*, i.e.,

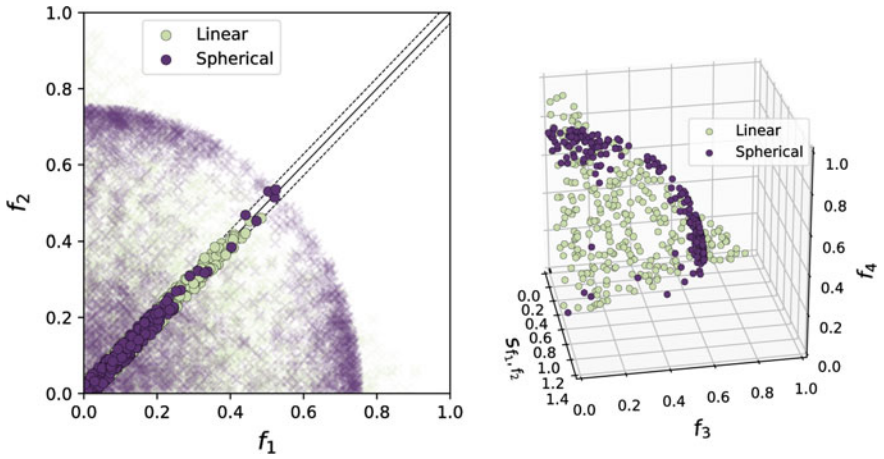
$$s_{f_1,f_2}(\boldsymbol{y}) - s_{f_1,f_2}(\boldsymbol{x}) \geq 2d \max\{\tan\varphi, \tan^{-1}\varphi\}, \tag{9}$$

then the original solution $\boldsymbol{x}$ actually dominates the original solution $\boldsymbol{y}$. This is important as it enables us to 'trust' the visualization to a certain (measurable) degree.

Furthermore, we can compute the original values of f_1 and f_2 from the value of s_{f_1,f_2} up to a precision dependent on d and φ:

$$f_1(x) = s_{f_1,f_2}(\boldsymbol{x}) \cos\varphi \pm d \sin\varphi, \tag{10}$$
$$f_2(x) = s_{f_1,f_2}(\boldsymbol{x}) \sin\varphi \pm d \cos\varphi. \tag{11}$$

We illustrate visualization with prosections using two sets of solutions from the 4-D objective space (each containing 3,000 solutions). The first is *linear*, with values ranging in $[0, 1]^4$ and a uniform random distribution of solutions. The second is



(a) Scatter plot (solutions in the section are emphasized)

(b) Prosection plot (only solutions in the section are shown)

Fig. 3 The scatter (**a**) and prosection (**b**) plots of the 4-D linear and spherical sets for the section on the plane f_1, f_2 with $\varphi = 45°$ and $d = 0.02$

spherical, with values in $[0, 0.75]^4$ and a non-uniform random distribution of solutions (only few solutions are located in the middle of the objective space, while most of them are near its four corners). The sets are intertwined, meaning that in one region, the linear dominates the spherical one, while in others, the spherical dominates the linear one. The two sets have very different characteristics and are therefore appropriate for analyzing the properties of visualization methods.

Figure 3 shows the scatter and projection plots of the linear and spherical sets for the section on the plane f_1, f_2 with $\varphi = 45°$ and $d = 0.02$. The two plots are supplementary in that the scatter plot depicts the two sets in their entirety, while the prosection plot presents a more detailed 3-D view of the solutions contained in the section. The chosen section cuts through two out of the four clusters of solutions from the spherical set. This cannot be seen from the scatter plot because the two clusters overlap when viewed in a projection on the plane f_1, f_2, but is immediately visible in the prosection plot. Moreover, it is clear that the shape and distribution of solutions from the sets are well-retained in the prosection plot. In addition, $2d \max\{\tan\varphi, \tan^{-1}\varphi\}$ from Eq. (9) equals 0.04 for the chosen section (depicted with the black line next to the origin $(0, 0, 0)$ in Fig. 3b), which means that almost all of the solutions that look like they dominate the solutions from the other set actually dominate them in the original 4-D sets. In other words, almost all Pareto dominance relations between the two sets are maintained after prosection in this case.

Since one prosection plot shows only a small part of the sets, different angles φ should be used to view the entire sets. Additionally, multiple planes could be chosen for dimension reduction, which is analogous to the choice of planes for a scatter plot.

5 Optimization Results and Their Analysis

In this work, AεSεH uses SBX crossover and polynomial mutation. The population size is $P_{\text{size}} = 680$, and the number of generations 100. The reference neighborhood size for AεSεH is $H_{\text{size}}^{\text{Ref}} = 20$. The algorithm archives the population at each generation. The fitness functions are computed using FABLE, a computer program that simulates the propagation phase based on an analytic averaging technique [18] that reduces computational time. It takes between one and two minutes to evaluate a solution with FABLE. Ten runs with different random seeds were performed. A large number of non-dominated solutions were obtained in each run, with small variance among runs. Here we report the results for a typical run of the algorithm.

In the following, we first examine the non-dominated solutions computed after joining all archived populations, which amounts to 3,496 solutions. Then, we analyze the convergence of the AεSεH algorithm by visualizing non-dominated solutions from population archives at 20, 40, 60, 80 and 100 generations.

5.1 Analyzing Non-dominated Solutions

Figure 4 shows the approximation for the Pareto optimal set found by the algorithm on the plane f_1, f_2, the IES operation time and time to reach the Moon, coloring solutions according to their value in function f_4, the initial mass of the spacecraft. Recall that the initial mass is to be maximized, while the remaining objectives are

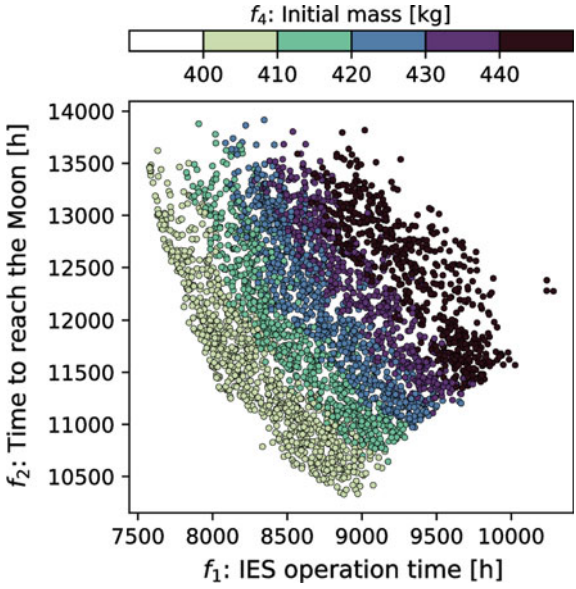


Fig. 4 f_1: Ion Engine System operation time and f_2: Time to reach the Moon, colored by f_4: Initial mass of the spacecraft

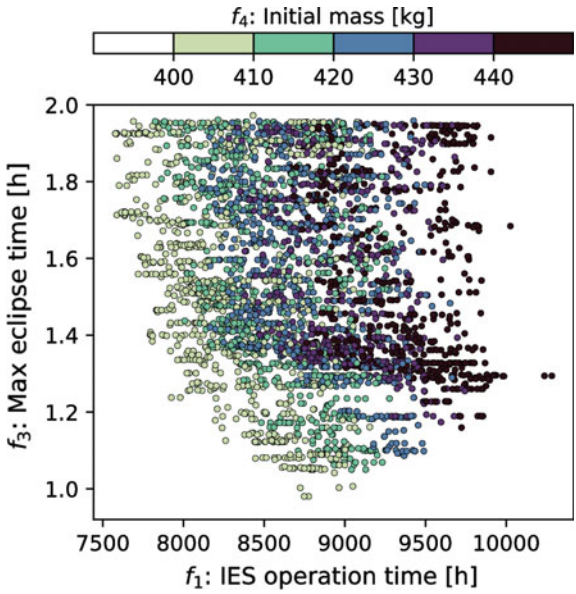


Fig. 5 f_1: Ion Engine System operation time and f_3: Maximum eclipse time, colored by f_4: Initial mass of the spacecraft

to be minimized. The figure clearly shows the trade-off between objectives f_1 and f_2 and illustrates the impact of increasing the initial mass. Note that for a given value of initial mass (f_4), the IES operation time (f_1) must be increased in order to reduce the time to reach the Moon (f_2). Increasing the initial mass implies a longer time to reach the Moon and also a longer IES operation time. The IES operation time is correlated to the amount of fuel required for the engine. This figure allows to estimate appropriate bounds for the required fuel according to the initial mass. As already mentioned, in the DESTINY mission, 1.5 years (around 13,000 h) is the maximum desired time to reach the Moon. Note that the algorithm finds a large number of solutions with the time to reach the Moon similar to or smaller than the maximum desired time.

Figure 5 shows the obtained approximation for the Pareto optimal set on the plane f_1, f_3, the IES operation time and the maximum eclipse time, also coloring solutions according to their value in function f_4, the initial mass of the spacecraft. In this problem, solutions with a maximum eclipse time of 2 h or less are desired to avoid larger batteries. Note that a large number of solutions within the desired range of maximum eclipse time can be found, irrespectively of the initial mass of the spacecraft.

Figures 6 and 7 show the propagation start date and hour (variable d) colored by f_2, time to reach the Moon, and by f_3, maximum eclipse time, respectively. Note that some of the solutions with time to reach the Moon $\leq$ 12,400 h have also a maximum eclipse time $\leq$ 1.4 h, satisfying the two main requirements for this problem formulation.

Next, we use prosections to visualize in more detail a section containing many solutions of interest, i.e., those with time to reach the Moon (f_2) lower than or equal

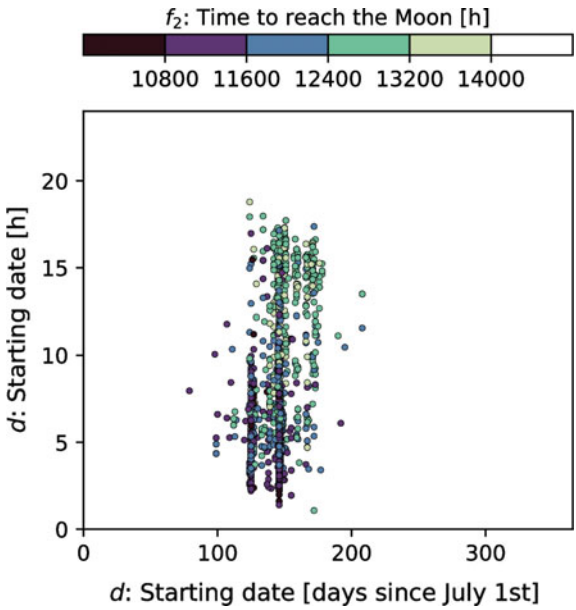


Fig. 6 Propagation starting date and hour, colored by f_2: Time to reach the Moon

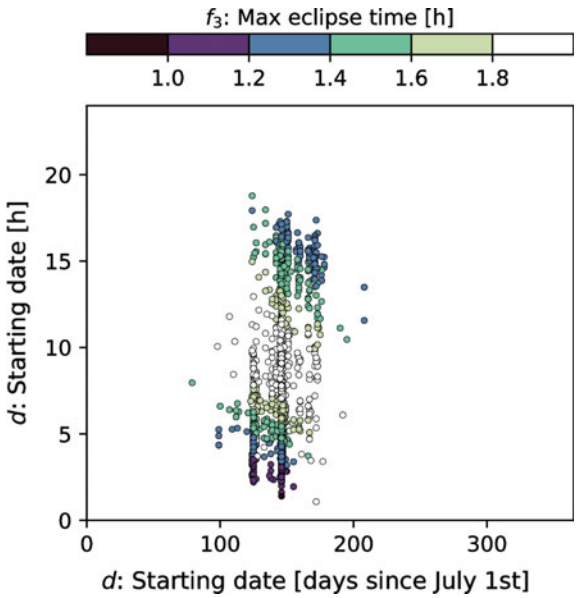
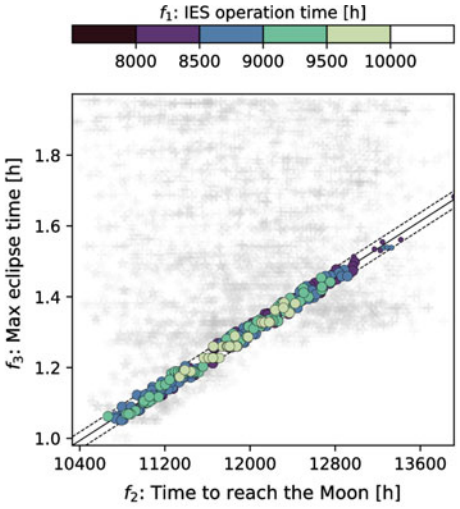
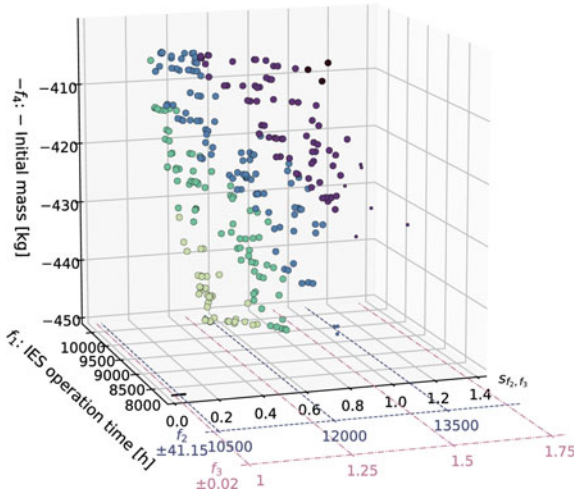


Fig. 7 Propagation starting date and hour, colored by f_3: Maximum eclipse time

to 13,000 h and maximum eclipse time (f_3) lower than or equal to 1.5 h. Figure 8 shows the scatter and prosection plots for the solutions within the section on the plane f_2, f_3 with $\varphi = 35°$ and $d = 0.02$. All solutions in the section are colored by



(a) Scatter plot



(b) Prosection plot

Fig. 8 The scatter (**a**) and prosection (**b**) plots for the section on the plane f_2, f_3 with $\varphi = 35°$ and $d = 0.02$, solutions are colored by f_1 (see text for more information)

the IES operation time (f_1). In addition, solutions outside of the 'region of interest' are depicted with smaller points.

The prosection plot in Fig. 8b is a 3-D scatter plot that shows the transformation s_{f_2,f_3} on the x axis, the objective f_1 on the y axis and the inverted objective $-f_4$ on the z axis (the inversion is done because prosection assumes minimization in all objectives). The original values of f_2 and f_3 are shown on the two additional x axes with dark and light dashed lines, together with their respective precisions as follows from Eqs. (10) and (11). We can see that due to the small section width,

the values of f_2 and f_3 can be restored with a high precision. The prosection plot shows well the seemingly linear trade-offs between objectives s_{f_2,f_3}, f_1 and f_4. A short IES operation time ($f_1 < 8{,}000$ h) is possible only for high initial mass (f_4) and (relatively) high, but still feasible, time to reach the Moon (f_2) and maximum eclipse time (f_3). On the other hand, if the IES operation time (f_1) is between 8,500 and 9,000 h, several trade-offs between the initial mass of the spacecraft (f_4) and time to reach the Moon (f_2) and maximum eclipse time (f_3) are possible.

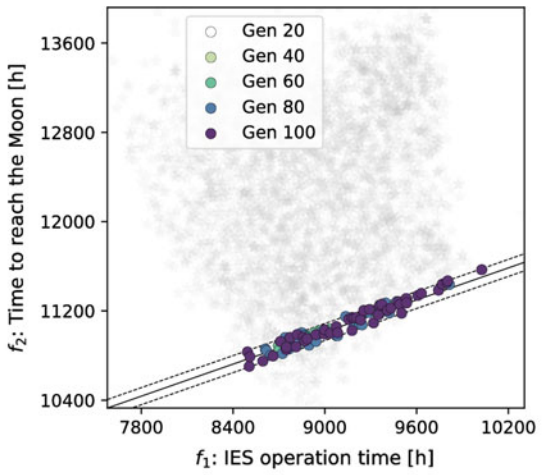
5.2 *Analyzing the Algorithm Dynamics*

So far, we have only shown the best solutions found by the AεSεH algorithm. Now we analyze the convergence of the algorithm with the help of visualization with prosections. To this end, we collect and visualize the non-dominated solutions of the current population of the AεSεH algorithm at 20, 40, 60, 80 and 100 generations.

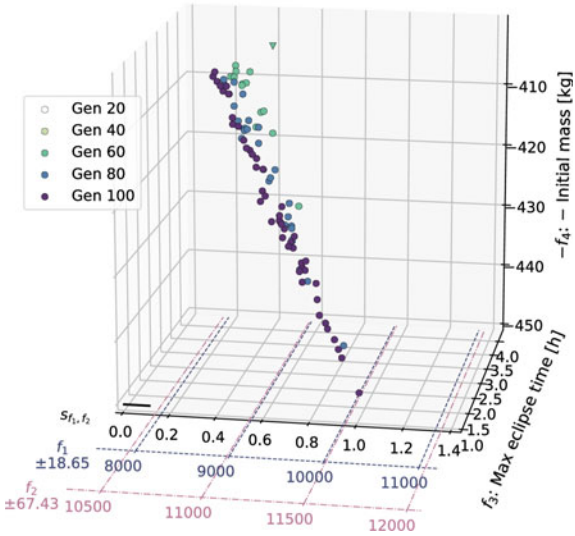
We show the results of the prosection on the plane f_1, f_2 under two different angles and section width $d = 0.02$ in Figs. 9 and 10. Solutions are colored according to the generation number. Furthermore, those for which maximum eclipse time (f_3) exceeds 2 h are denoted with different markers (triangles instead of circles) to ease the interpretation.

Plots in Fig. 9 of the section under the angle $\varphi = 20°$ show that no solutions were found by AεSεH in this part of the objective space in the first 40 generations. At 60 generations, some solutions are present, but only with low initial mass (f_4). Additional generations are needed to further widen the front in this part of the objective space.

Similar observations can be made by visualizing the solutions in the section under the angle $\varphi = 45°$ (see plots in Fig. 10). At generation 20, the solutions cover only a small part of the front and are far away from the best solutions. A lot of improvement can be observed at generations 40 and 60 in terms of convergence as well as the range of solutions. Contrary to this, the improvement between generations 80 and 100 is only minor, suggesting the search has converged to a (local) optimum. Note, however, that because of the loss of information due to projecting solutions to a low-dimensional space, not all solutions that appear to dominate other solutions in fact do so. This can be asserted by comparing the distances between solutions from different generations to the distance from Eq. (9) (shown in Figs. 9b and 10b with the black line starting at the plot origin $(0, 1, -450)$). We cannot be sure of the dominance relation between any two solutions that are closer in s_{f_1,f_2} than this distance (the other two coordinates are not affected by the projection). Note that the relations between solutions in a prosection plot can be explored much more easily on a computer where the 3-D plots can be rotated interactively, than by looking at the same plots on paper.



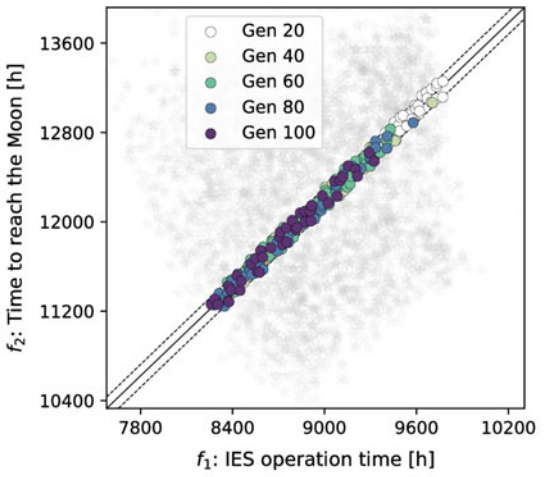
(a) Scatter plot



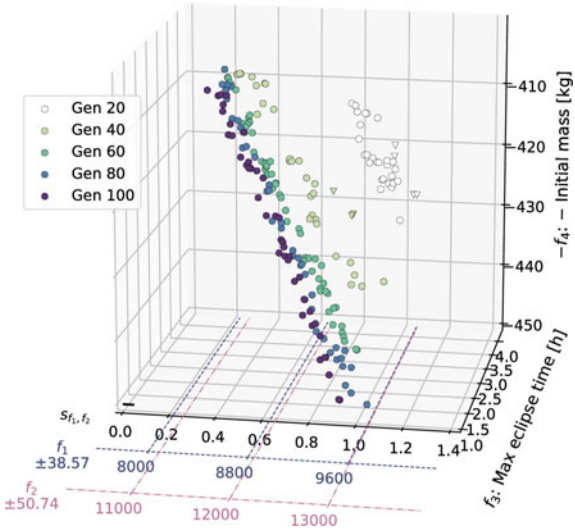
(b) Prosection plot

Fig. 9 The scatter (**a**) and prosection (**b**) plots for the section on the plane f_1, f_2 with $\varphi = 20°$ and $d = 0.02$, solutions are colored by generation number (see text for more information)

Finally, these plots also suggest that while early generations still contain some solutions with maximum eclipse time (f_3) of over 2 h, the penalization mechanism in AεSεH, as explained in Sect. 2, sucessfully navigates the search towards a smaller maximum eclipse time.



(a) Scatter plot



(b) Prosection plot

Fig. 10 The scatter (**a**) and prosection (**b**) plots for the section on the plane f_1, f_2 with $\varphi = 45°$ and $d = 0.02$, solutions are colored by generation number (see text for more information)

6 Conclusions

This work used the many-objective AεSεH evolutionary algorithm to find solutions for a four-objective formulation of the trajectory design problem of the JAXA's DESTINY mission. We analyzed the approximations of Pareto optimal sets of solutions found by the algorithm, focusing particularly on the trade-offs between the IES operation time, time to reach the Moon, maximum eclipse time, and the initial mass of the

spacecraft. We also analyzed the propagation starting date and the maximum eclipse time.

Visualization with prosections are helpful for analyzing in detail the outcome of the optimization and gaining insights into the problem. In addition, prosections allow to perform a fine-grained analysis of the algorithm dynamics and to verify whether the mechanisms it incorporates work as expected, as shown for the penalization of solutions when the eclipse time exceeds a maximum allowed value. Although in this work the analysis is a posteriori, prosections can provide on-line information from different sections in the objective space that could be fed back to a decision maker to guide the algorithm in an interactive way or to the algorithm itself to adapt its search.

In the near future we would like to study these applications of prosections. We would also like to analyze other many-objective formulations of the problem with five and six objectives. An important challenge here is how to use prosections for dimensions larger than four. Finally, the methodology used in this work for optimization, analysis and visualization is general and we would like to apply it to other real-world application domains.

Acknowledgements The Japanese authors acknowledge financial support from JSPS-MESS under the Japan-Slovenia Bilateral Joint Research Program. The Slovenian authors acknowledge financial support from the Slovenian Research Agency (project nos. BI-JP/16-18-003 and Z2-8177, and research core funding no. P2-0209). This work is also part of a project that has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 692286. The authors are grateful to Aljoša Vodopija for reproducing the DESTINY mission schematic view.

References

1. Aguirre, H., Oyama, A., Tanaka, K.: Adaptive ε-sampling and ε-hood for evolutionary many-objective optimization. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) Proceedings of the 7th International Conference on Evolutionary Multi-Criterion Optimization EMO 2013. Lecture Notes in Computer Science, vol. 7811, pp. 322–336. Springer, Berlin (2013)
2. Aguirre, H., Yazawa, Y., Oyama, A., Tanaka, K.: Extending AεSεH from many-objective to multi-objective optimization. In: Dick, G., Browne, W.N., Whigham, P.A., Zhang, M., Bui, L.T., Ishibuchi, H., Jin, Y., Li, X., Shi, Y., Singh, P., Tan, K.C., Tang K., (eds.) Proceedings of the 10th International Conference on Simulated Evolution and Learning, SEAL 2014. Lecture Notes in Computer Science, vol. 8886, pp. 239–250. Springer, Berlin (2014)
3. Filipič, B., Tušar, T.: Visualization in multiobjective optimization. In: Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2016, pp. 735–751. ACM (2016)
4. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)
5. Howell, K.C.: Three-dimensional, periodic, halo orbits. Celest. Mech. **32**(53), 53–71 (1984)
6. Inselberg, A.: Parallel Coordinates: Visual Multidimensional Geometry and its Applications. Springer, Berlin (2009)
7. Jaimes, A.L., Oyama, A., Fujii, K.: Space trajectory design: Analysis of a real-world many-objective optimization problem. In: Proceedings of the 2013 IEEE Congress on Evolutionary Computation, pp. 2809–2816 (2013)

8. Kawakatsu, Y., Funaki, I., Nishiyama, K., Oyama, A., Toyota, H., Yamamoto, T., Iwata, T.: DESTINY – A technology demonstrator for deep space exploration. In: 30th International Symposium on Space Technology and Science, ISTS 2015 (2015)
9. Kawakatsu, Y., Iwata, T.: Destiny mission overview: a small satellite mission for deep space exploration technology demonstration. Adv. Astronaut. Sci. **146**, 727–739 (2013)
10. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multiobjective optimization. Evol. Comput. **10**(3), 263–282 (2002)
11. Morita, Y., Imoto, T., Tokudome, S., Ohtsuka, H.: Epsilon rocket launcher and future solid rocket technologies. Trans. Jpn. Soc. Aeronaut. Space Sci. Aerosp. Technol. Jpn. **10**(28), 19–24 (2012)
12. Nishio, Y., Oyama, A., Akimoto, Y., Aguirre, H., Tanaka, K.: Many-objective optimization of trajectory design for DESTINY mission. In: Proceedings of the Learning and Intelligent Optimization Conference 2014 (LION 8), pp. 1–4 (2014)
13. Srinivas, N., Deb, K.: Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evol. Comput. **2**(3), 221–248 (1994)
14. Tušar, T., Filipič, B.: Visualizing 4D approximation sets of multiobjective optimizers with prosections. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2011, pp. 737–744. ACM (2011)
15. Tušar, T., Filipič, B.: Visualization of Pareto front approximations in evolutionary multiobjective optimization: a critical review and the prosection method. IEEE Trans. Evol. Comput. **19**(2), 225–245 (2015)
16. Tweedie, L., Spence, R., Dawkes, H., Su, H.: Externalising abstract mathematical models. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '96, pp. 406–412. ACM (1996)
17. Watanabe, T., Tatsukawa, T., Yamamoto, T., Oyama, A., Kawakatsu, Y.: Multi-objective optimization and data mining of a space trajectory for DESTINY. In: The 25th Workshop on JAXA Astrodynamics and Flight Mechanics, pp. 138–150 (2015)
18. Zuiani, F., Kawakatsu, Y., Vasile, M.: Multi-objective optimisation of many-revolution, low-thrust orbit raising for Destiny mission. In: Proceedings of the 23rd AAS/AIAA Space Flight Mechanics Conference, pp. 1–20 (2013)