Complex & Intelligent Systems
https://doi.org/10.1007/s40747-023-01123-2

ORIGINAL ARTICLE

Gravity assist space pruning and global optimization of spacecraft trajectories for solar system boundary exploration

Yuqi Song[1] · Weiren Wu[1,2] · Hang Hu[1] · Mingpei Lin[3] · Hui Wang[3] · Jinxiu Zhang[3]

Received: 22 December 2022 / Accepted: 21 April 2023

Abstract
The solar system boundary exploration mission has the characteristics of long flight time, high fuel consumption, complex gravity-assist sequence and strict constraints. Therefore, the number of decision variables and the search space of the transfer trajectory are very large, resulting in poor convergence and efficiency of the global search of the metaheuristic algorithm. Moreover, the existing gravity assist space pruning algorithm is no longer applicable for solar system boundary exploration. To effectively reduce the search space and improve the effect of trajectory optimization, an improved gravity assist space pruning algorithm is proposed. In this algorithm, a unique pruning procedure is used to effectively prune the search space, a shape of solution space box bounds combining rectangle and rhombus is adopted, and a method to automatically determine the solution space box bounds is presented. To verify the effectiveness of the improved gravity assist space pruning algorithm, the sensitivity of pruning effect to parameters is analyzed and the optimization effects of three typical metaheuristics are compared. The optimization results of 50 repeated runs of the differential evolution algorithm in the entire search space and the solution space box bounds are compared. Simulation results show that the performance of differential evolution algorithm is better than bat algorithm and firefly algorithm. And the improved pruning algorithm can increase the efficiency of subsequent optimization by more than eleven times and the convergence probability of the objective function by fifty of times. The applicability and efficiency of the proposed method for the solar system boundary exploration are demonstrated.

Keywords Solar system boundary exploration · Global optimization of transfer trajectory · Differential evolution algorithm · Gravity assist space pruning · Solution space box bounds

Introduction

The future of space exploration includes a focus on the solar system boundary, which is located 80-150AU from the sun at the edge of the heliosphere [1]. Since the 1970s, although some spacecraft have passed through the heliosphere and reached or are about to reach interstellar space, such as the Pioneer 10/11 [2–4], Voyager 1/2 [5–7] and New horizon missions [8], the solar system boundary has not been the primary target of exploration, but rather as extensions of planetary exploration missions. Since the exploration of the solar system boundary can not only obtain major scientific discoveries, but also promote the leap-forward development of cutting-edge space technologies, China proposes to implement the solar system boundary exploration plans, in which two almost identical spacecraft will fly to the nose and tail of the heliosphere respectively and reach 100AU before October 1, 2049 [9]. Strict flight time and distance constraints mean that the spacecraft will have an average speed of 4.5 AU/year, which is greater than the average speed of the Pioneer and Voyager missions. Moreover, the gravity-assist sequences will be different and the optimization of interplanetary transfer trajectories will be more complicated.

The optimization of interplanetary transfer trajectory is essentially a typical NP-hard problem [10]. Due to the complexity and nonlinearity of the problem, it is difficult for traditional deterministic methods to solve the problem. As a stochastic algorithm that does not require the gradient and

✉ Jinxiu Zhang
zhangjinxiu@sysu.edu.cn

1 School of Physics and Astronomy, Sun Yat-sen University, Zhuhai 519082, China

2 Lunar Exploration and Space Engineering Center, Beijing 100037, China

3 School of Aeronautics and Astronautics, Sun Yat-sen University, Shenzhen 518107, China

convexity of the problem, metaheuristics have been identified as very effective global optimizers and used in a wide range of practical NP-hard challenges [11]. Firstly, the metaheuristic algorithm is applied to the tuning of the controller parameters. Stojanovic et al. [12] applied the bat algorithm (BA) to optimize the cascade control parameters of the robot platform, and showed significant performance improvement compared with the traditional parameter tuning method. Nedic et al. [13] proposed a parameter optimization method of cascade controller based on the firefly algorithm (FA), and compared the best results with the results obtained by other metaheuristic algorithms to show the advantages of FA. Fang et al. [14] designed a new online policy iteration algorithm to obtain an adaptive optimal controller for a class of nonlinear Markovian jump systems. Secondly, metaheuristics are widely used in the field of machine learning. Tuba et al. [15] proposes a handwritten digit recognition algorithm which uses projection histogram for feature extraction and support vector machine for classification. To effectively extract local and global features from handwritten word images, Malakar et al. [16] implement a global search strategy based on genetic algorithm (GA) in the wrapper method to select features. Finally, metaheuristics are also widely used to find the optimal value of the interplanetary transfer trajectory to minimize fuel consumption. Wall et al. [17] applied the genetic algorithm to the trajectory optimization of the asteroid patrol mission, and a near-optimal trajectory is obtained by combining the outer loop genetic algorithm for the sequence optimization and the inner loop genetic algorithm for trajectory optimization. Rosa et al. [18] proposed a hybrid evolutionary algorithm which synergistically exploits differential evolution, genetic algorithms and particle swarm optimization, and applied it to spacecraft trajectory optimization. Vasile et al. [19] tested and compared the global search performance of metaheuristics such as genetic algorithm, differential evolution algorithm, and particle swarm optimization on the space trajectory design problem.

The biggest challenge in the application of metaheuristics is the No Free Lunch (NFL) theorem, that is, there is no general algorithm that can achieve the best results for all optimization problems [11]. Moreover, there are a large number of local optimal solutions in the interplanetary transfer trajectory optimization problem, so the convergence and computational efficiency of the metaheuristic algorithm cannot be guaranteed. An effective approach to this problem is to prune the search space of the problem before applying the metaheuristic. Myatt et al. developed an efficient pruning method for the multiple gravity assists (MGA) problem and named it gravity assist space pruning (GASP) Algorithm [20]. Considering the sequential nature of the problem, the GASP algorithm prunes the search space on a phase-by-phase basis. It results in significant computational savings with search space reductions greater than six orders of magnitude, thus simplifying significantly the subsequent optimization. The pruning method has been shown to have polynomial time and space complexity, so that it remains tractable as the number of decision variables increases [21, 22]. Subsequently, Izzo et al. summarized the research status of GASP algorithm and pointed out that some attempts had been made to improve GASP algorithm [23]. Armellin et al. provided a mathematical proof on the global optimality of the found solution using GASP algorithm based on differential algebra [24]. Schütze et al. proposed a method to design optimal low-thrust gravity-assist trajectories using space pruning and a multi-objective approach [25]. Yang et al. introduced an image tool into the gravity assist space pruning algorithm to determine the boundary of the solution space [26]. In addition, the GASP algorithm was extended to the multiple gravity assists with one deep space maneuver (MGA-1DSM) problem [27–29].

However, both the original GASP algorithm and the corresponding improved algorithm are only suitable for the traditional planetary exploration mission, but not for the solar system boundary exploration mission. The gravity-assist sequence of the solar system boundary exploration is more complex than the planetary exploration, the flight time and distance are longer and the constraints of the mission are more stringent. As a result, the dimensions of decision variables and search space for trajectory optimization are extremely complex, and the previous gravity assist space pruning algorithm has been unable to apply to such a situation. All these situations require us to develop an efficient gravity assist space pruning algorithm for the solar system boundary exploration mission and to effectively reduce the search space and improve the convergence and efficiency of the metaheuristic algorithm for trajectory optimization.

In this study, the search space of the transfer trajectory optimization is determined according to the requirements of the solar system boundary exploration mission in the direction of the heliosphere's tail. An improved gravity assist space pruning algorithm is then proposed to better prune the search space, and the improvement of the algorithm is mainly reflected in two aspects. On the one hand, the forward and backward constraining are used to further prune the search space after each gravity assist pruning, on the other hand, the shape of solution space box bounds combining rectangle and rhombus is used and the several sets of reduced box bounds are automatically determined according to the discontinuity of the pruned solution space. Then the sensitivity of pruning effect to the gravity assist maximum thrust constraint is analyzed and the trajectory optimization effects of differential evolution algorithm, firefly algorithm and bat algorithm are compared. The differential evolution algorithm is used to optimize the trajectories in the entire search space and the pruned solution space box bounds, the convergence probability, convergence efficiency and the performance of the optimal solution are compared.

The main contributions in this paper are listed as the following facts:

It is the first time to carry out a multi-gravity assist trajectory study on the heliosphere tail exploration mission, and to establish a transfer trajectory optimization model for solar system boundary exploration. The optimization effects of differential evolution algorithm, bat algorithm and firefly algorithm are compared and analyzed.

An improved gravity assist space pruning algorithm is proposed to solve the problem of low efficiency and poor convergence of the optimization algorithm. A new pruning procedure and a method for determining the solution space box bounds are proposed. The sensitivity of the pruning algorithm to the gravity assist maximum thrust constraint is studied.

Differential evolution algorithm is used to optimize the trajectories in the entire and pruned search space respectively, by comparing the convergence probability and convergence efficiency of the optimization, the effectiveness of the proposed gravity assist space pruning algorithm is verified.

The rest of the paper is arranged as follows. "Gravity assist model" section describes the gravity-assist models, including MGA and MGA-1DSM. "Optimization and space pruning" section proposes the optimization method of transfer trajectory for the solar system boundary exploration and illustrates the improved gravity assist space pruning algorithm. "Simulation and results" section shows the simulation results of the space pruning as well as the trajectory optimization. "Conclusion" section summarizes the conclusions.

Gravity assist model

There are roughly two types of gravity-assist models. The MGA model is to apply an impulsive maneuver only at the periapsis of the hyperbolic trajectory of the spacecraft relative to the gravity-assist planet. The MGA-1DSM model is to apply a deep space maneuver at some point during the transfer between two adjacent gravity assists.

MGA model

Figure 1 shows the process of gravity assist in the MGA model. Since the gravity-assist time is small relative to the overall mission transfer time, it can be considered that the gravity assist is instantaneous, and the heliocentric position vector of the spacecraft is consistent with that of the gravity-assist planet. By solving Lambert's problem, we can determine the spacecraft's incoming and outgoing velocity vectors $\mathbf{v}_{\text{s/c-in}}$ and $\mathbf{v}_{\text{s/c-out}}$ in the heliocentric frame for each gravity assist. Defining $\mathbf{v}_{\text{p}}$ as the velocity vector of the gravity-assist planet, the incoming and outgoing velocities relative to the planet (hyperbolic residual velocities) are [30]

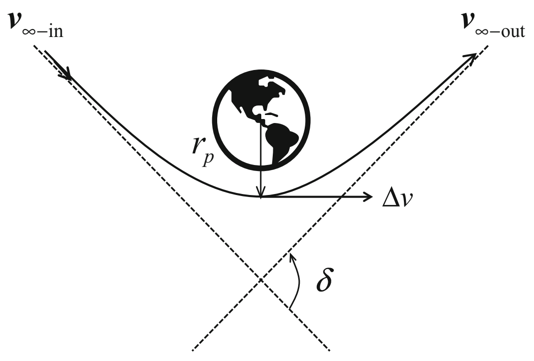


Fig. 1 Schematic diagram of the periapsis maneuver in the MGA model

$$\begin{aligned} \mathbf{v}_{\infty-\text{in}} &= \mathbf{v}_{\text{s/c-in}} - \mathbf{v}_{\text{p}} \\ \mathbf{v}_{\infty-\text{out}} &= \mathbf{v}_{\text{s/c-out}} - \mathbf{v}_{\text{p}} \end{aligned} \tag{1}$$

These velocity vectors are along the asymptotic directions of the incoming and outgoing hyperbolic orbits about the planet, respectively. The gravity-assist transfer angle δ is

$$\delta = \langle \mathbf{v}_{\infty-\text{in}}, \mathbf{v}_{\infty-\text{out}} \rangle, \tag{2}$$

where the symbol $\langle\rangle$ denotes the angle between two vectors. Then, the gravity-assist periapsis radius r_{p} is obtained by solving the following equation

$$\sin^{-1}(1/e_{\text{in}}) + \sin^{-1}(1/e_{\text{out}}) = \delta, \tag{3}$$

where e_{in} and e_{out} are the eccentricities of incoming and outgoing hyperbolic orbits given by

$$\begin{aligned} e_{\text{in}} &= 1 + \frac{r_{\text{p}}}{\mu_{\text{p}}} v_{\infty-\text{in}}^2 \\ e_{\text{out}} &= 1 + \frac{r_{\text{p}}}{\mu_{\text{p}}} v_{\infty-\text{out}}^2, \end{aligned} \tag{4}$$

where μ_{p} is the gravitational constant of the gravity-assist planet, $v_{\infty-\text{in}}$ and $v_{\infty-\text{out}}$ are the magnitudes of vectors $\mathbf{v}_{\infty-\text{in}}$ and $\mathbf{v}_{\infty-\text{out}}$, respectively. The magnitude of the maneuver at periapsis can be found by

$$\Delta v = \left| \sqrt{v_{\infty\text{ - in}}^2 + \frac{2\mu_{\text{p}}}{r_{\text{p}}}} - \sqrt{v_{\infty\text{ - out}}^2 + \frac{2\mu_{\text{p}}}{r_{\text{p}}}} \right| \tag{5}$$

MGA-1DSM model

As shown in Fig. 2, the gravity assist with deep space maneu-

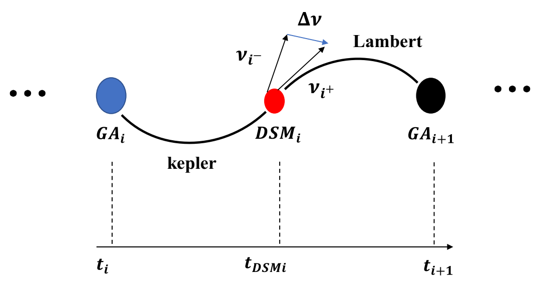


Fig. 2 Schematic diagram of deep space maneuver in the MGA-1DSM model

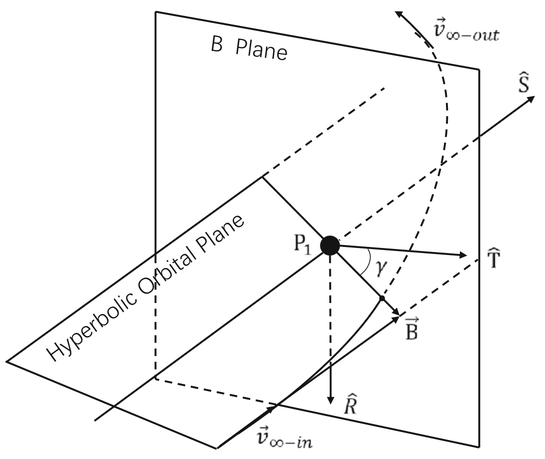


Fig. 3 Schematic representation of the parameters of the B-plane

ver can split the transfer trajectory between two adjacent gravity assists into two legs according to the position where the deep space maneuver is applied. The first leg is obtained by Kepler orbital propagation according to the departure velocity, and the second leg is obtained by solving the Lambert problem. As shown in Fig. 3, the B-plane model is used to describe the gravity-assist process, where r_p is the periapsis radius of the gravity assist and γ is the B-plane angle.

Because the flyby is unpowered during the gravity assist, the spacecraft will follow a hyperbolic path about the planet. Therefore, the magnitudes of the incoming and outgoing velocity vectors relative to the planet are equal, that is

$$v_{\infty-\text{in}} = v_{\infty-\text{out}} = v_\infty, \tag{6}$$

where v_∞ is the magnitude of the hyperbolic residual velocity.

Next, we need to find the outgoing velocity vector. Firstly, the eccentricity of the hyperbola is found by

$$e = 1 + \frac{r_\text{p} v_\infty^2}{\mu_\text{p}} \tag{7}$$

Then, the gravity-assist transfer angle may be found by

$$\delta = 2\sin^{-1}(1/e) \tag{8}$$

Finally, the outgoing velocity vector is obtained by [31]:

$$\mathbf{v}_{\infty\text{ - out}} = v_\infty(\cos\delta\hat{\mathbf{S}} + \cos\gamma\sin\delta\hat{\mathbf{T}} + \sin\gamma\sin\delta\hat{\mathbf{R}}), \tag{9}$$

where $\hat{\mathbf{S}}$, $\hat{\mathbf{T}}$ and $\hat{\mathbf{R}}$ is a unit vector defined by

$$\begin{aligned} \hat{\mathbf{S}} &= \frac{\mathbf{v}_{\infty\text{ - in}}}{v_{\infty\text{ - in}}} \\ \hat{\mathbf{T}} &= \frac{\hat{\mathbf{S}}\times\mathbf{v}_\text{p}}{\left\|\hat{\mathbf{S}}\times\mathbf{v}_\text{p}\right\|} \\ \hat{\mathbf{R}} &= \hat{\mathbf{S}}\times\hat{\mathbf{T}} \end{aligned} \tag{10}$$

The outgoing heliocentric velocity vector can then be found as

$$\mathbf{v}_{\text{s/c - out}} = \mathbf{v}_{\infty\text{ - out}} + \mathbf{v}_\text{p} \tag{11}$$

Optimization and space pruning

In this section, we first transform the transfer trajectory optimization problem for the solar system boundary exploration into a nonlinear programming problem with multiple constraints. The standard form of the optimization model and the optimization method we adopt are presented. Then we introduce the improved gravity assist space pruning algorithm in detail, including the introduction of basic pruning criteria and the description of the unique pruning procedure, as well as the explanation of the method to determine box bounds of the solution space after pruning.

Optimization model and method

Traditional planetary exploration missions require the spacecraft to eventually be inserted into an elliptical orbit around the target planet, so the braking maneuver is indispensable. While the solar system boundary exploration mission requires the spacecraft to fly out of the solar system along a hyperbolic orbit and no braking maneuvers are required. Therefore, the last gravity assist is described by the B-plane parameters of the MGA-1DSM model, and meanwhile the MGA model is used in the 1st to the $(N-1)$-th gravity assist. Where N denotes the total number of gravity assists performed. After the N-th gravity assist, the spacecraft travels in a Kepler orbit for a while, and then applies a deep space maneuver, and finally flies to the heliosphere's tail in a Kepler orbit again. On October 1, 2049, the magnitude of the heliocentric position vector of the spacecraft is r_{2049}, and the angle between the spacecraft's heliocentric position vector and the tail's position vector is θ_{2049}.

The decision vector of the trajectory optimization for the solar system boundary exploration is $\mathbf{x} = [t_0, T_1, T_2, \ldots T_i \ldots T_N, r_{pN}, \gamma, \eta, (\Delta v)_{\text{dsm}}]$, where t_0 represents the departure date, T_i represents the time interval between two adjacent gravity assists, r_{pN} and γ represent the periapsis radius and the B-plane angle of the N-th gravity assist, respectively. η represents the deep space maneuver time coefficient and the time to apply the deep space maneuver is given by

$$\begin{aligned} t_{\text{dsm}} &= t_N + \eta \times (t_f - t_N) \\ t_N &= t_0 + \sum_{i=1}^{N} T_i \\ t_f &= 2049,\ 10,\ 1, \end{aligned} \tag{12}$$

where t_f denotes the end time of the mission, t_N denotes the time of the N-th gravity assist. $(\Delta v)_{\text{dsm}}$ represents the magnitude of the deep space maneuver applied at t_{dsm}.

Here, the velocity vectors of the spacecraft before and after maneuver at t_{dsm} are defined as $\mathbf{v}_{\text{sc}}^0$ and $\mathbf{v}_{\text{sc}}^1$, respectively, and the direction of the vector $\mathbf{v}_{\text{sc}}^0$ is defined as $\mathbf{n}_{sc}$. The deep space maneuver is applied along $\mathbf{n}_{\text{sc}}$, so $\mathbf{v}_{\text{sc}}^1$ can be calculated by $\mathbf{v}_{\text{sc}}^1 = \mathbf{v}_{\text{sc}}^0 + (\Delta \mathbf{v})_{\text{dsm}}$, where $(\Delta \mathbf{v})_{\text{dsm}}$ is the deep space maneuver vector determined by $(\Delta \mathbf{v})_{\text{dsm}} = (\Delta v)_{\text{dsm}} \times \mathbf{n}_{\text{sc}}$.

The optimization variables are assumed to be in the following range:

$$\begin{aligned} & t_0 \in \Gamma(t_0) \\ & T_i \in \Gamma(T_i), \quad i = 1, \ldots N \\ & r_{pN} \in \Gamma\big(r_{pN}\big) \\ & \gamma \in \Gamma(\gamma) \\ & \eta \in \Gamma(\eta) \\ & \Delta v_{\text{dsm}} \in \Gamma(\Delta v_{\text{dsm}}), \end{aligned} \tag{13}$$

where $\Gamma()$ represents a one-dimensional search space formed by value ranges of the variable in parentheses. The entire search space Γ can be expressed as:

$$\begin{aligned} \Gamma := {} & \Gamma(t_0) \times \cdots \Gamma(T_i) \cdots \times \Gamma(T_N) \\ & \times \Gamma\big(r_{pN}\big) \times \Gamma(\gamma) \times \Gamma(\eta) \times \Gamma(\Delta v_{\text{dsm}}) \end{aligned} \tag{14}$$

The optimization model can be described as

$$\begin{aligned} & \text{find: } \mathbf{x} \in \Gamma \\ & \text{to minimise: } J(\mathbf{x}) = \sum\nolimits_{i=1}^{N-1} (\Delta V)_i(\mathbf{x}) + (\Delta v)_{\text{dsm}}(\mathbf{x}) + k \\ & k = k_1 + k_2 + k_3 + k_4 + k_5 \\ & k_1 = c_1 \max\big(0,\ C_3(\mathbf{x}) - C_3^{\max}\big) \\ & k_2 = \sum\nolimits_{i=1}^{N-1} c_{2i} \max(0,\ r_{pi}^{\min} - r_{pi}(\mathbf{x})) \\ & k_3 = c_3 \max(0,\ 100\text{AU} - r_{2049}(\mathbf{x})) \\ & k_4 = c_4 \max(0,\ \theta_{2049}(\mathbf{x}) - 45^\circ) \\ & k_5 = \sum\nolimits_{i=1}^{N-1} c_{5i} \max\big(0,\ (\Delta V)_i(\mathbf{x}) - (\Delta V)_i^{\max}\big) \end{aligned} \tag{15}$$

The objective of the optimization is to minimize the total maneuver subject to the constraints, so the objective function J consists of three terms, which are the sum of the periapsis maneuvers $\sum_{i=1}^{N-1} (\Delta V)_i$, the deep space maneuver $(\Delta v)_{\text{dsm}}$ and penalty term k. The penalty term k consists of five sub-terms k_1, k_2, k_3, k_4 and k_5, which represent the constraints on C_3, periapsis radius, position, flight direction and periapsis maneuver, respectively. Where C_3 is equal to the square of the hyperbolic residual velocity at launch, which is usually used to measure the carrying capacity of a rocket. c_1, c_{2i}, c_3, c_4 and c_{5i} represent the penalty coefficients, respectively, and are usually taken to be a large constant. When the constraint conditions are not satisfied, the penalty term k will become the dominant term of the objective function value, so C_3 should be less than the given value $C_3^{\max}$, the periapsis radius r_{pi} of each gravity assist should be greater than the given value $r_{pi}^{\min}$, r_{2049} should be greater than 100AU, θ_{2049} should be less than 45° and each periapsis maneuver $(\Delta V)_i$ should be less than a given threshold $(\Delta V)_i^{\max}$.

Generally speaking, a variety of optimization algorithms such as genetic algorithm, particle swarm optimization, differential evolution algorithm, simulated annealing algorithm and so on can be used for trajectory optimization [32–34], but for each specific problem, these algorithms show different performance. It is generally believed that differential evolution algorithm has good performance in dealing with various trajectory optimization problems [35–37]. In this paper, we mainly use adaptive differential evolution algorithm to search for the optimal solution on the entire search space and the pruned solution space to compare the efficiency of optimization and the effect of space pruning. Differential evolution algorithm is used in all optimizations with the consistent parameters set as follows: the crossover probability is $0.5 \times (1 + \text{rand}(0, 1))$, the mutation probability is $F = 0.5 \times 2^{\lambda}$, $\lambda = e^{1 - G/(G+1-G_{\max})}$ and mutation operator is

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{i,G} + 0.35 \times (\mathbf{x}_{\text{best},G} - \mathbf{x}_{i,G}) + F \times (\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}), \tag{16}$$

where G is the current generation number, $G_{\max}$ is the maximum number of the generation, $\mathbf{x}_{i,G}$ represents the individual before mutation, $\mathbf{v}_{i,G+1}$ represents the individual after mutation, $\mathbf{x}_{r1,G}$ and $\mathbf{x}_{r2,G}$ denote the randomly selected individuals from the population, and $x_{\text{best},G}$ means the current best individual.

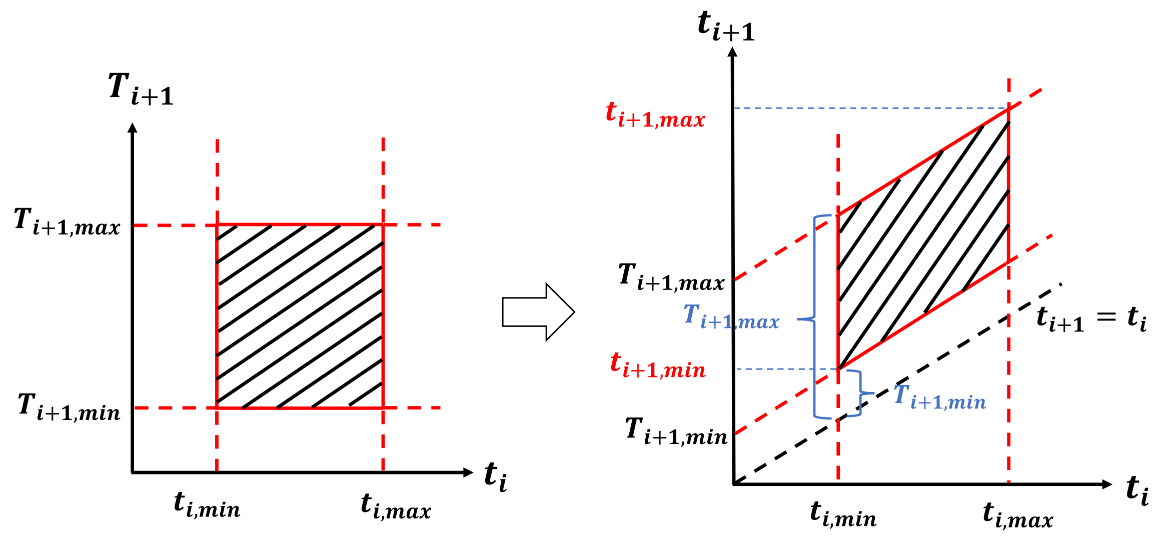


Fig. 4 The search space is converted from rectangle to rhomb

Improved gravity assist space pruning algorithm

The first step of the improved gravity assist space pruning algorithm is to transform the search space of the problem, which is similar to the original GASP algorithm. Applying the transformation defined by the simple relation $t_i = t_0 + \sum_{j=1}^{i} T_j$, $i = 0, \ldots, N$ to the search space Γ, we obtain a new search space that we denote with $\Gamma^* = f(\Gamma)$. As shown in Fig. 4, the mapping f realizes the transformation of the rectangular search space consisting of t_i and T_i into the rhombic search space consisting of t_i and t_{i+1}. The $(i + 1)$-th Lambert arc can be solved in terms of t_i and t_{i+1}, regardless of the other arc. Thus, the high-dimensional search space is decoupled into a cascade of multiple 2-dimensional search spaces.

The decision vector of the new search space is then [21, 25]:

$$f : \mathbf{x} = [t_0, T_1, \ldots, T_N] \rightarrow \mathbf{X} = [t_0, t_1, \ldots, t_N] \quad (17)$$

A new statement of the optimization problem in terms of absolute times t_i can be formulated as shown in Eq. (18)

find: $\mathbf{X} \in \Gamma^*$

to minimise: $J(\mathbf{X}) = \sum_{i=1}^{N-1} (\Delta V)_i(\mathbf{X}) + (\Delta v)_{\mathrm{dsm}}(\mathbf{X}) + k$

$$k = k_1 + k_2 + k_3 + k_4 + k_5$$
$$k_1 = c_1 \max(0,\ C_3(\mathbf{X}) - C_3^{\max})$$
$$k_2 = \sum_{i=1}^{N-1} c_{2i} \max(0,\ r_{pi}^{\min} - r_{pi}(\mathbf{X}))$$
$$k_3 = c_3 \max(0,\ 100\mathrm{AU} - r_{2049}(\mathbf{X}))$$
$$k_4 = c_4 \max(0,\ \theta_{2049}(\mathbf{X}) - 45^\circ)$$
$$k_5 = \sum_{i=1}^{N-1} c_{5i} \max\left(0,\ (\Delta V_i)(\mathbf{X}) - (\Delta V)_i^{\max}\right) \quad (18)$$

The 2-dimensional search space is then sampled at an appropriate resolution with possible departure dates in the horizontal axis and prospective arrival dates in the vertical axis. Because of this, the number of Lambert problem evaluations is vastly reduced. The grid sampling will require many less ephemeris calculations as the same positions/velocities need not be recalculated for a given departure or arrival time [20, 21].

The improved gravity assisted space pruning algorithm needs to consider the following pruning criteria, namely C_3 constraining, gravity assist maximum thrust constraint, gravity assist angular constraint, forward constraining and backward constraining. The details of these pruning criteria are as follow [21, 23]:

1. C_3 constraining: The maximum allowable C_3 works on the sampled space $\Gamma^*(t_0) \times \Gamma^*(t_1)$, pruning out all those points where corresponding trajectories have unfeasible C_3.
2. Gravity assist maximum thrust constraint: If the difference between incoming and outgoing velocities during a gravity assist is larger than a threshold T_{v}, the corresponding trajectories will be pruned out. Through the threshold T_{v} and the incoming velocity, we can determine the reasonable range of outgoing velocity, so as to prune out the trajectories with infeasible outgoing velocity. Similarly, according to the threshold T_{v} and the outgoing velocity after pruning, the reasonable range of the incoming

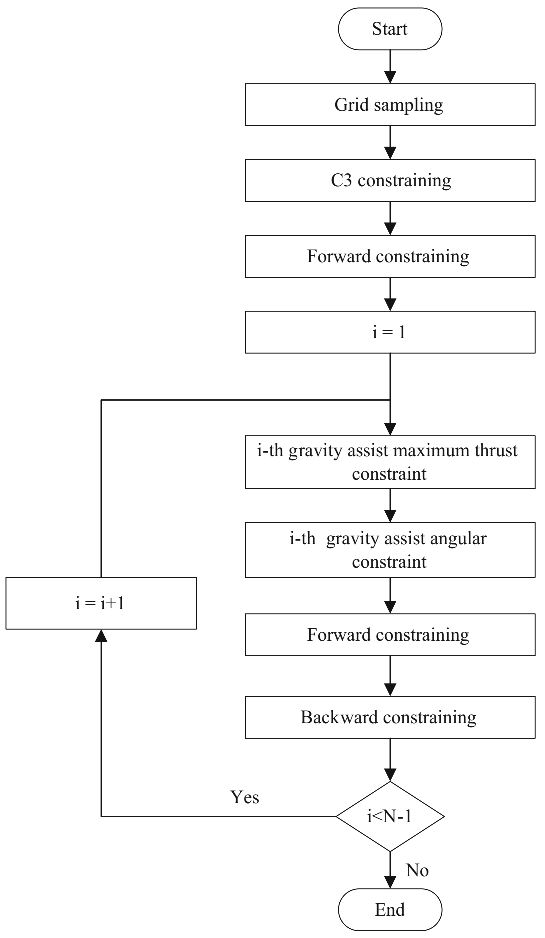


Fig. 5 Pruning procedure of the improved GASP algorithm

Table 1 Pruning steps for gravity assist maximum thrust constraint

Steps: gravity assist maximum thrust constraint
1. Calculate the bounds on the possible incoming velocity $v^{\text{in}}_{\min}$ and $v^{\text{in}}_{\max}$
2. Invalidate any outgoing trajectories that do not have outgoing velocities in the range $\left[v^{\text{in}}_{\min} - T_{\text{v}},\ v^{\text{in}}_{\max} + T_{\text{v}}\right]$
3. Calculate the modified bounds on outgoing velocity $v^{\text{out}}_{\min}$ and $v^{\text{out}}_{\max}$
4. Invalidate any incoming trajectories with velocities outside the range $\left[v^{\text{out}}_{\min} - T_{\text{v}},\ v^{\text{out}}_{\max} + T_{\text{v}}\right]$

Table 2 Pruning steps for gravity assist angular constraint

Steps: gravity assist angular constraint
1. For all i incoming trajectories
2. For all j outgoing trajectories
3. If the periapsis height of gravity assist is valid for the current incoming and outgoing trajectory
4. Mark both incoming and outgoing trajectory as valid
5. End If
6. End For
7. End For
8. Invalidate all trajectories not marked as valid

velocity can be determined, so that the incoming velocity not in the range can be pruned. The detailed pruning steps are shown in Table 1.

3. Gravity assist angular constraint: The gravity assist angular constraint prunes out infeasible trajectories with periapsis height less than the minimum safe distance for a given planet. Assuming i valid incoming trajectories and j valid outgoing trajectories, the pruning steps for gravity assist angular constraint are shown in Table 2.
4. Forward constraining: If the arrival date in $\Gamma^*(t_{i-1}) \times \Gamma^*(t_i)$ becomes unfeasible because of pruning, the relative departure date in $\Gamma^*(t_i) \times \Gamma^*(t_{i+1})$ has to be pruned out.
5. Backward constraining: If the departure date in $\Gamma^*(t_i) \times \Gamma^*(t_{i+1})$ becomes unfeasible because of pruning, also the relative arrival date in $\Gamma^*(t_{i-1}) \times \Gamma^*(t_i)$ has to be pruned out.

The original version of GASP algorithm only uses forward and backward constraining after the application of the C_3 constraining and the braking maneuver constraint. However, in this study, to ensure that the impact of each pruning operation in the current phase can be transmitted to all other phases and better play the effect of the pruning, a pruning procedure different from the basic pruning algorithm is formed. The pruning criteria of the forward and backward constraining are not only used after the C_3 constraining, but also after each gravity assist pruning (gravity assist maximum thrust constraint and gravity assist angular constraint). In this way, the search space can be effectively reduced and the arrival and departure dates of the preceding and following phases can be matched. The pruning procedure of the improved gravity assist space pruning algorithm are shown in Fig. 5.

The box bounds of the solution space after pruning needs to be determined to obtain the range of the decision variables t_0 and T_i. Since infeasible parts of the space are pruned, the solution space is usually divided into several independent spaces and several sets of box bounds with smaller volume are left. As shown in Fig. 6, the pruned solutions are shown as red dots, and due to grid sampling, each solution actually represents a solution space box shown as a small black box centered on the dot. Because the envelope of the solution space box is an irregular shape, it is difficult to define the

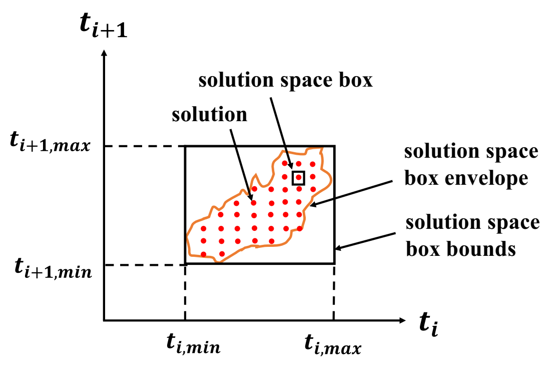


Fig. 6 Solution space box and solution space box bounds

value range of the decision variables corresponding to it completely, the solution space box envelope is therefore replaced by the solution space box bounds that happens to completely contain the solution space and is regular in shape, which in effect results in an inevitable enlargement of the solution space. But this enlargement is limited and tolerable because the box bounds are tangent to the solution space envelope.

The existing GASP algorithms all use a single rectangle or rhombus to determine the box bounds, but the rectangle or rhombus has some shortcomings. As shown in Fig. 7, when the solution space is close to the boundary of the original search space, the solution space box bounds determined by the rectangle is easy to exceed the boundary of the original search space, so that solutions that do not belong to the search space are included, and even solutions with $T_{i+1} \leq 0$ may be obtained, which is inconsistent with the actual situation. Although the rhombic solution space box bounds can determine the value range of T_{i+1} accurately enough, the corresponding t_{i+1} is always beyond the range determined by the rectangular box bounds, resulting in non-correspondence with the next two-dimensional search space on t_{i+1}.

In this paper, we use a combination of rectangle and rhombus to determine the solution space box bounds. Firstly, the value range $[t_{i,\min}, t_{i,\max}]$ of t_i is determined according to the overlapping part of the rectangular and rhombic box bounds, and then the value range $[T^*_{i+1,\min}, T^*_{i+1,\max}]$ of T_{i+1} is determined according to the intercept of the extension line of the upper and lower bounds of the rhombic box bounds on the t_{i+1} axis. According to the value ranges $[t_{i,\min}, t_{i,\max}]$ and $[T^*_{i+1,\min}, T^*_{i+1,\max}]$, the value range $[t^*_{i+1,\min}, t^*_{i+1,\max}]$ of t_{i+1} is obtained using the mapping f in Eq. (17), and finally the value range $[t^{\text{end}}_{i,\min}, t^{\text{end}}_{i,\max}]$ of t_{i+1} is modified according to Eq. (19).

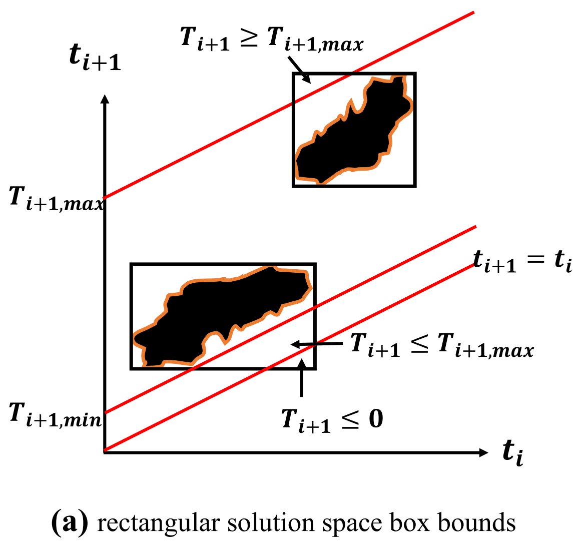
$$
\begin{aligned}
t^{\text{end}}_{i+1,\min} &= \begin{cases} t_{i+1,\min}, & \text{if } (t_{i+1,\min} \geq t^*_{i+1,\min}) \\ t^*_{i+1,\min}, & \text{else} \end{cases} \\
t^{\text{end}}_{i+1,\max} &= \begin{cases} t_{i+1,\max}, & \text{if } (t_{i+1,\max} \leq t^*_{i+1,\max}) \\ t^*_{i+1,\max}, & \text{else} \end{cases},
\end{aligned}
\tag{19}
$$

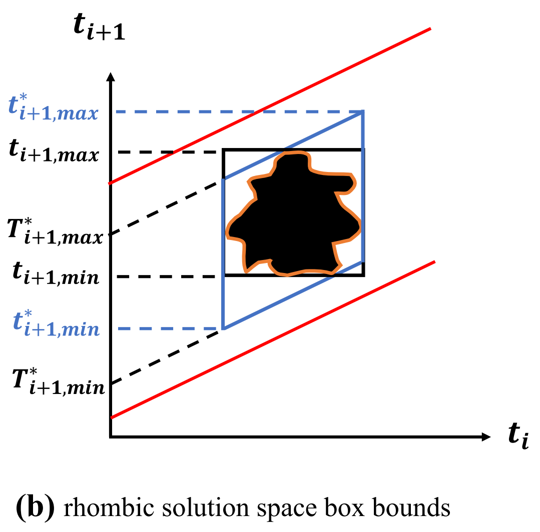
where $[t_{i+1,\min}, t_{i+1,\max}]$ is the range of t_{i+1} determined from the rectangular solution space box bounds. The ranges $[t_{i,\min}, t_{i,\max}]$, $[T^*_{i+1,\min}, T^*_{i+1,\max}]$ and $[t^{\text{end}}_{i,\min}, t^{\text{end}}_{i,\max}]$ together determine the solution space box bounds for the combination of the rectangle and rhombic. When the decision variables t_i and T_{i+1} are subsequently optimized on the solution space box bounds, Eq. (19) is also used to correct the decision variable values of the newly generated individuals after the mutation and crossover operations of each generation of the differential evolution algorithm (it has the effect of boundary absorption).

In addition, in the matrix corresponding to the pruned search space, the positions of solutions belonging to the same box bounds in the matrix are adjacent, while the positions of solutions not belonging to the same box bounds are discontinuous. Based on this feature, we implement a method to automatically determine every rectangular solution space box bound, and the detailed steps are shown in Table 3. Using the proposed pruning method, the visualization of the high-dimensional search space and the subsequent effective optimization in the reduced search range can be realized.

Simulation and results

To better test and show the effect of the pruning algorithm, we choose the flight sequence E-V-V-E-J-N-Tail with as many times of gravity assists as possible. Although this sequence may not be the optimal sequence for the solar system boundary exploration, it is beneficial to highlight the superiority of the pruning algorithm. With the flight sequence E-V-V-E-J-N-Tail, the spacecraft departs from the Earth, passes through two Venus gravity assists (VGA), one Earth gravity assist (EGA), one Jupiter gravity assist (JGA) and one Neptune gravity assist (NGA), finally flies to the tail of the heliosphere. The value range of the decision vector is shown in Eq. (20), which defines a complete and complicated search space. According to the search space of the transfer trajectory given by Eq. (20), the ranges of t_0, T_1, T_2, T_3, T_4 and T_5 are [9132, 11322], [30, 400], [230, 470], [30, 400], [400, 1000] and [1000, 3000] respectively. k_{Ti} and $k_{ti} (i = 0, 1, 2, 3, 4, 5)$ are defined as the number of bins after the ranges of T_i and t_i are discretized, respectively. With a sampling resolution of 5 days, the number of bins obtained is $k_{t0} = 439$, $k_{T1} = 74$, $k_{T2} = 48$, $k_{T3} = 74$,





(a) rectangular solution space box bounds **(b)** rhombic solution space box bounds

Fig. 7 **a**, **b** Shortcomings of rectangle and rhombus in determining the solution space box bounds, respectively

$k_{T4} = 120$ and $k_{T5} = 400$, respectively. According to the relation $t_{i+1} = t_i + T_{i+1}$, we can obtain the number of bins $k_{t1} = 513, k_{t2} = 561, k_{t3} = 635, k_{t4} = 755$ and $k_{t5} = 1155$. The number of Lambert problems to be solved at stage $t_i - t_{i+1}$ using the gravity assist space pruning algorithm is $N_{t_i t_{i+1}} = k_{t_i} * k_{t_{i+1}}$. The total number of Lambert problems to be solved is $N_{\text{gasp}} = \sum_{i=0}^{4} N_{t_i t_{i+1}} = 2, 220, 685$. However, if the high-dimensional search space is directly discretized using the traverse search with the same sampling resolution, the total number of Lambert problems to be solved is $N_{\text{ts}} = \prod_{i=0}^{4} k_{t_i} = 6.996 \times 10^{16}$. It can be seen that the gravity assist space pruning algorithm reduces the number of Lambert problems by more than $N_{\text{ts}}/N_{\text{gasp}} = 3.15 \times 10^{10}$ times, which is efficient in calculation and implementation.

$$\begin{aligned}
&t_0 \in [9132,\ 11322]\,\text{MJD2000}\\
&T_1 \in [30,\ 400]\,\text{days}\\
&T_2 \in [230,\ 470]\,\text{days}\\
&T_3 \in [30,\ 400]\,\text{days}\\
&T_4 \in [400,\ 1000]\,\text{days}\\
&T_5 \in [1000,\ 3000]\,\text{days}\\
&r_{pN} \in [1.1,\ 300] R_{pN}\\
&\gamma \in [-\pi, \pi]\,\text{rad}\\
&\eta \in [0.01, 0.99]\\
&\Delta V_{\text{dsm}} \in [0, 3]\,\text{km/s}
\end{aligned} \tag{20}$$

Table 3 Steps to get the rectangular solution space box bounds

```
Steps: get rectangular solution space box bounds

1. Obtain the solution space matrix $M_s$ after pruning ($m$ rows, $n$ columns)
2.  For the $k$th box bounds
3.   row(1) = {1:$m$}, $i$ = 1
4.   For $j$ = 1:$n$
5.    If ($M_s$ (row{$i$},$j$)~ = NaN
6.     row($i$ + 1) = {find(~is NaN ($M_s$(row($i$),$j$)))}
7.     If $M_s$(row{$i$ + 1},$j$ + 1) is NaN
8.     Break;
9.     End If
10.     $i = i + 1$;
11.    End If
12.   End
13.   Extract $M_s$ (all elements in row, 1 : $j$) elements to get box bounds $k$
14.   Delete columns whose entire elements is NaN in box bounds $k$
15.  End
```

The simulation results of pruning the solution space using the proposed gravity assist space pruning algorithm are shown in "Gravity assist space pruning" section, while the simulation results of trajectory optimization in the entire search space are shown in "Trajectory optimization in the entire search space" section. The optimization results on the solution space box bounds after pruning and the comparison of trajectory optimization results before and after pruning are shown in "Trajectory optimization in the solution space box bounds" section.

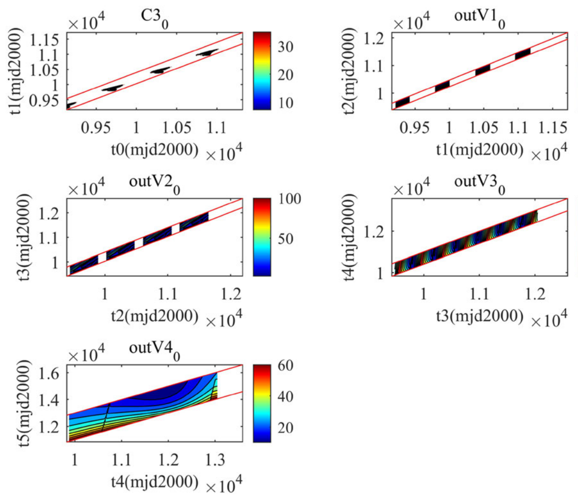


Fig. 8 C3 constraining pruning

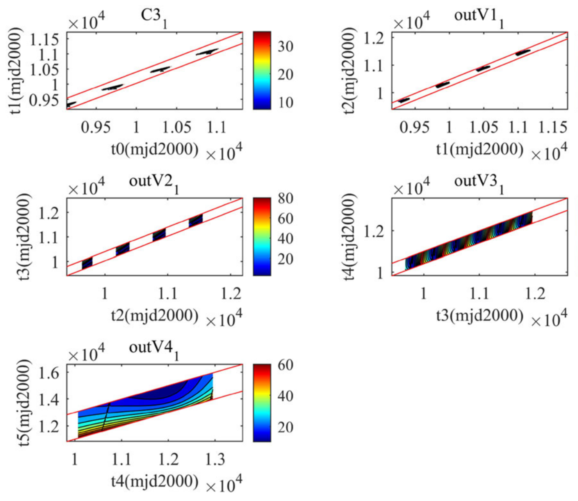


Fig. 9 1st gravity assist constraint pruning

Gravity assist space pruning

A sampling resolution of 5 days is used in both axes to yield the search space. C_3 constraint is 36 km^2/s^2, each gravity assist maximum thrust constraint T_v is 5 km/s. Gravity assist angular constraint for inner solar system planets is $r_{pi}^{\min} = 1.05R_{pi}$, while the corresponding value for outer solar system planets is $r_{pi}^{\min} = 1.1R_{pi}$, where R_{pi} is the mean radius of the gravity-assist planet.

The execution time of the improved GASP algorithm is 17.97 s on a 3.4 GHz Intel(R) Xeon Duo, and the pruning algorithm only needs to be executed once in total. The process of gradually pruning the solution space with the addition of different constraints is shown in Figs. 8, 9, 10, 11 and 12, where each figure shows the results of five two-dimensional (2D) search spaces ($\Gamma^*(t_0)\times\Gamma^*(t_1)$, $\Gamma^*(t_1)\times\Gamma^*(t_2)$, $\Gamma^*(t_2)\times\Gamma^*(t_3)$, $\Gamma^*(t_3)\times\Gamma^*(t_4)$ and $\Gamma^*(t_4)\times\Gamma^*(t_5)$) pruned under

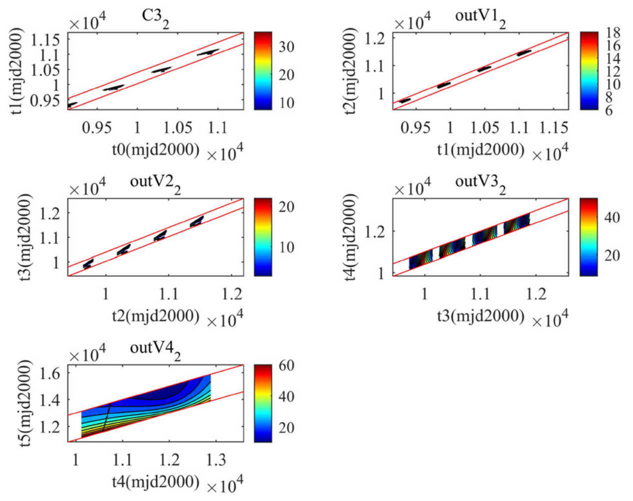


Fig. 10 2nd Gravity assist constraint pruning

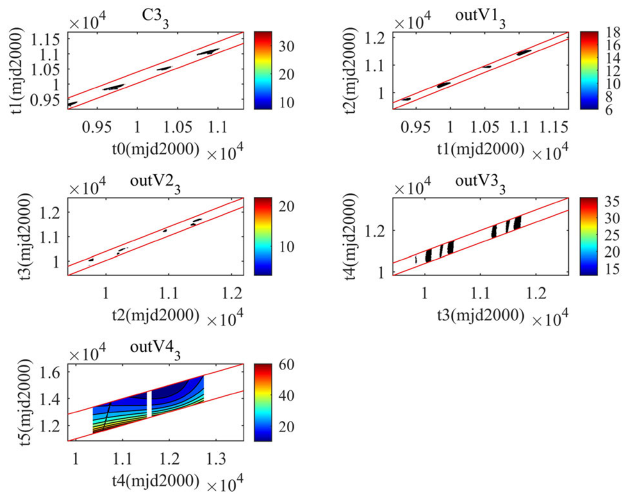


Fig. 11 3rd gravity assist constraint pruning

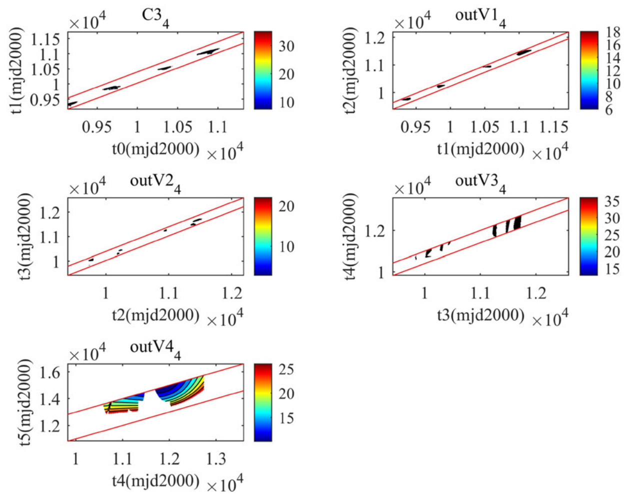


Fig. 12 4th gravity assist constraint pruning

the current constraints. It can be seen that the C_3 constraining pruning and the first and second gravity assist pruning have the most obvious pruning effect on the search spaces $\Gamma^*(t_0) \times \Gamma^*(t_1)$, $\Gamma^*(t_1) \times \Gamma^*(t_2)$ and $\Gamma^*(t_2) \times \Gamma^*(t_3)$, while the third and the fourth gravity assist pruning has the most obvious pruning effect on the search space $\Gamma^*(t_3) \times \Gamma^*(t_4)$ and $\Gamma^*(t_4) \times \Gamma^*(t_5)$, respectively.

After applying the fourth gravity assist constraint pruning, final five 2D search spaces C3$_4$(t_0, t_1),outV1$_4$(t_1, t_2),outV2$_4$(t_2, t_3),outV3$_4$(t_3, t_4) and outV4$_4$(t_4, t_5) are obtained and shown in Fig. 13a–e, respectively, where outVi_j represents the magnitude of the hyperbolic residual velocity when the spacecraft leaves the i-th planet after the j-th gravity assist constraint pruning, and C3$_j$ represents the C_3 after the j-th gravity assist constraint pruning. Starting from the above five different 2D search spaces, different numbers of solution spaces box bounds can be determined respectively. Table 4 shows the number of solution space box bounds that can be obtained starting from the 2D search space (a), (b), (c), (d) and (e) respectively. It can be seen that the number of solution space box bounds determined starting from space (d) is the largest, so the solution space can be divided in the most detail. At the same time, the minimum number of solution spaces box bounds can be determined from space (e), although the number of subsequent optimizations can be reduced, the solution space for each optimization is still too large. In summary, we choose to determine the solution space box bounds starting from the space (b) to balance the fineness of the space partition and the ease of the optimization process.

The rectangular solution space box bounds named box 1–4 determined from space (b) are shown in Fig. 13b. According to the relationship that the arrival time of the previous phase is the departure time of the next phase, the rectangular box bounds corresponding to the other four 2D search spaces can be obtained as shown in Fig. 13a, c, d, e, respectively.

Table 5 shows the value range of the decision variables corresponding to the box1-box4. The value ranges of t_0, T_1, T_2, T_3, T_4 and T_5 constitute the rhombic box bounds, and the value ranges of t_0, t_1, t_2, t_3, t_4 and t_5 constitute the rectangular box bounds. The intersection of these two box bounds is the solution space after pruning. Therefore, in the subsequent optimization, the range of rhombic box bounds is used as the range of optimization variables, and then the range of rectangular box bounds is used to constrain and correct the value of the corresponding t_i.

To better illustrate the effect of the gravity assist space pruning algorithm, we study the sensitivity of the simulation results to the value of the gravity assist maximum thrust constraint T_v. Fixing the sampling resolution to 5 days, we let T_v vary from zero to 15.00 km/s and take a value every 0.10 km/s. In this way, we need to test the effect of gravity assist space pruning under 150 different T_v values (note that in principle, the gravity assist maximum thrust constraint can take different values according to the different planet at each gravity assist, but in this study, for convenience, a unified value T_v is taken for all gravity assists). To evaluate the effect of gravity assist space pruning under different values of T_v, we count the number of elements in the solution space matrix after pruning as a standard to measure the pruning effect, and draw the curves of the number of elements with the change of T_v in final five 2D search spaces C3$_4$, outV1$_4$, outV2$_4$, outV3$_4$ and outV4$_4$. As shown in Fig. 14, the pruning effect of 2D spaces C3$_4$, outV1$_4$ and outV2$_4$ is less sensitive to T_v, and the number of elements is almost unchanged with the increase of T_v. This is because the required periapsis maneuver itself is relatively small for Venus gravity assist, and it is easy to satisfy the constraint of T_v. However, since the required periapsis maneuver is relatively large for Earth gravity assist and Jupiter gravity assist, the 2D search spaces outV3$_4$ and outV4$_4$ are very sensitive to T_v. When the value of T_v is small, the constraint of periapsis maneuver cannot be satisfied, and the number of elements in the solution space decreases rapidly due to pruning. The actual trajectory optimization results also show that the magnitude of the two periapsis maneuvers for Venus gravity assist are very small, which are 0.72 km/s and 0.00 km/s, respectively, while the periapsis maneuvers for Earth gravity assist and Jupiter gravity assist are larger, which are 5.00 km/s and 3.62 km/s, respectively. This is consistent with the results of our analysis. In addition, in the 2D search space outV4$_4$, the number of elements increases rapidly with the increase of T_v when T_v is less than 5.00 km/s, and the increasing trend slows down when T_v is greater than 5.00 km/s. When T_v is equal to 5.00 km/s, it becomes an inflection point, which is also the reason why T_v is taken as 5.00 km/s in the gravity assist space pruning in this paper.

Trajectory optimization in the entire search space

The global optimization is carried out in the entire search space defined as Eq. (20) using differential evolution (DE) [35], bat algorithm (BA) [38] and firefly algorithm (FA) [39], respectively. The population size is NP $=$ 200 and the maximum number of the generation is $G_{\max} = 3000$ in all three algorithms. The simulation was repeatedly run 50 times due to inconsistent convergence of the algorithm each time. To compare the results of computational intelligence experiments, a single-problem analysis is usually performed, dealing with the results obtained over several runs of the algorithms over a given problem [40]. Here we count the minimum, maximum, average, and standard deviation of the total maneuver in 50 runs of each algorithm to compare the performance of different algorithms. As shown in Table 6, the value of DE is smaller than the other two algorithms in terms of minimum, maximum, average and standard deviation, so

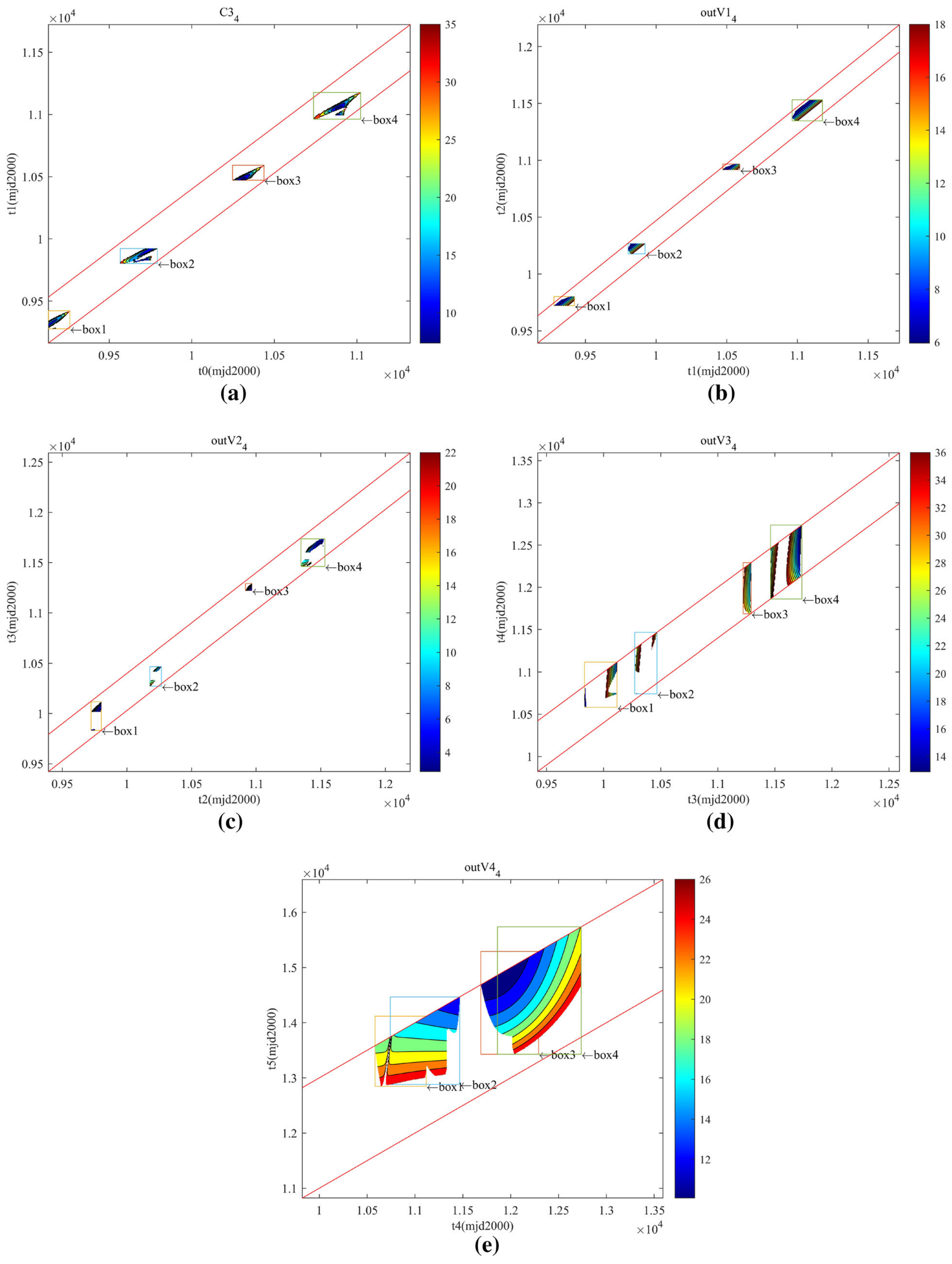


Fig. 13 Solution space after pruning can be divided into four box bounds starting from 2D space (**b**). Where (**a**) shows box1–4 in C3$_4$(t_0, t_1), (**b**) shows box1–4 in outV1$_4$(t_1, t_2), (**c**) shows box1–4 in outV2$_4$(t_2, t_3), (**d**) shows box1–4 in outV3$_4$(t_3, t_4) and (**e**) shows box1–4 in outV4$_4$(t_4, t_5)

Table 4 Number of box bounds starting from different space

Space	(a)	(b)	(c)	(d)	(e)
Number of box bounds	5	4	7	8	2

Table 5 The decision variables range of the pruned solution space box bounds

Variables	box1	box2	box3	box4
t_0 (mjd2000)	[9132,9262]	[9567,9792]	[10247,10437]	[10737,11022]
T_1 (days)	[100,210]	[75,245]	[130,225]	[70,235]
t_1 (mjd2000)	[9277,9422]	[9802,9922]	[10472,10592]	[10962,11177]
T_2 (days)	[345,445]	[325,445]	[345,445]	[345,445]
t_2 (mjd2000)	[9722,9802]	[10177,10267]	[10917,10967]	[11347,11532]
T_3 (days)	[85,315]	[90,230]	[255,325]	[60,245]
t_3 (mjd2000)	[9832,10117]	[10272,10467]	[11222,11292]	[11462,11737]
T_4 (days)	[625,1000]	[450,1000]	[400,1000]	[400,1000]
t_4 (mjd2000)	[10582,11117]	[10742,11467]	[11687,12292]	[11862,12737]
T_5 (days)	[1900,3000]	[1745,3000]	[1390,3000]	[1390,3000]
t_5 (mjd2000)	[12847,14117]	[12882,14467]	[13427,15292]	[13427,15737]

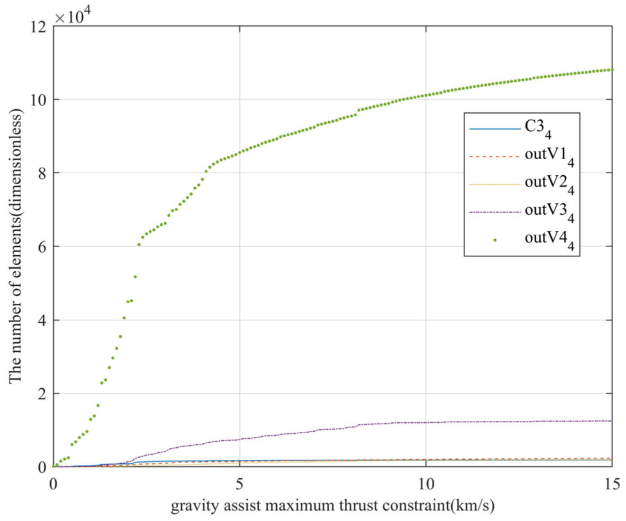


Fig. 14 Sensitivity of solution space pruning results to the value of the gravity assist maximum thrust constraint

Table 6 Results of 50 repeated runs in the entire search space

Optimization algorithm	Total maneuver (km/s)			
	Minimum	Maximum	Average	Standard deviation
DE	9.34	19.25	14.19	2.10
BA	13.36	47.18	19.71	6.84
FA	16.17	59.78	25.08	9.02

DE has better performance. The optimal performances of the three algorithms are ranked as follows: DE is greater than BA, BA is greater than FA.

The convergence processes of the objective functions of DE, BA and FA are shown in Figs. 15, 16 and 17, respectively. It can be seen that after 3000 iterations, the objective function of DE can all be reduced to below 500, and more than half of the objective function of BA can be reduced to below 500, while the objective function of FA can be reduced below 500 only in less than a quarter of the cases. The actual statistical results from Table 7 show that the number of times that the objective functions of DE, BA and FA converge below 500 in 50 repeated runs is 50, 36 and 7, respectively. Therefore, from the point of view of iterative convergence, there are also conclusions that the performance of DE is greater than that of BA, and the performance of BA is greater than that of FA. In view of this, in the following content of this paper, we only discuss the optimization results of differential evolution algorithm.

Although the performance of DE is the best among the three algorithms, its standard deviation of 50 runs is still large, reaching 2.10 km/s. This indicates that the difference between each two runs of DE is large, which makes the results of a single optimization not repeatable and makes it difficult to determine whether the results of the next run will be better than those of the current run. In addition, only one of the 50 runs converged to a minimum of 9.34 km/s, only four converged below 10.00 km/s, and the remaining 46 converged above 11.44 km/s. The probability of convergence to the minimum is only 2%. It can also be seen from Fig. 15 that the minimum number of generations required for all objective functions to converge below 500 is 516, which is relatively large, indicating that the convergence efficiency is low. In

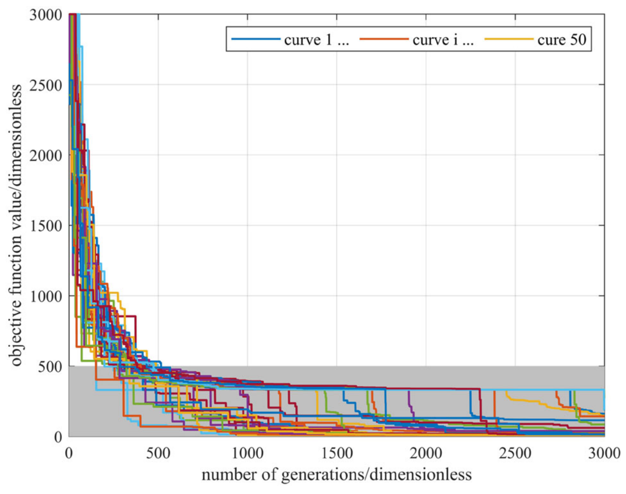


Fig. 15 Convergence process of differential evolution algorithm in the entire search space

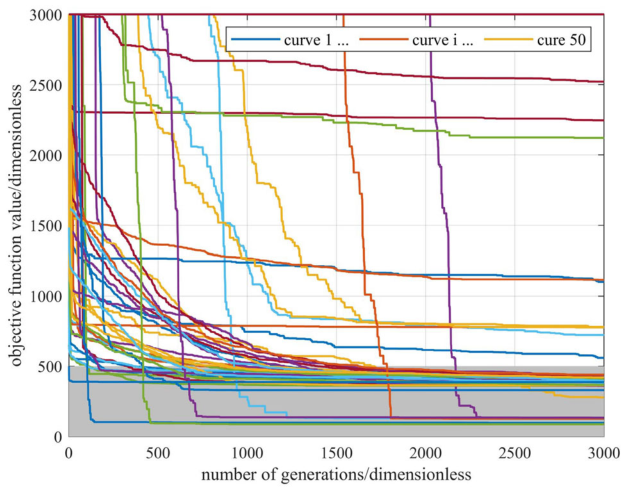


Fig. 16 Convergence process of bat algorithm in the entire search space

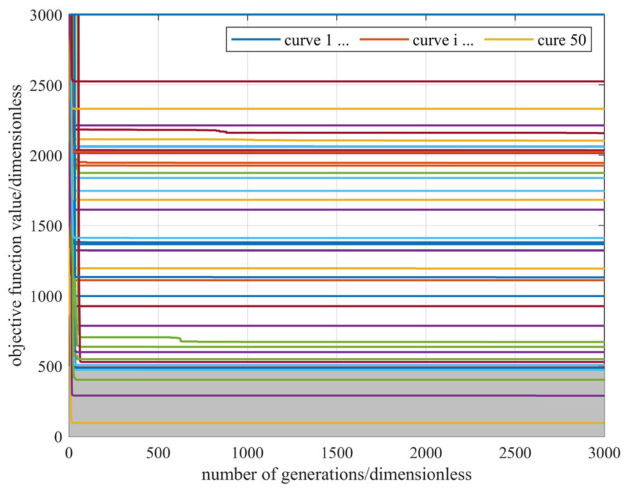


Fig. 17 Convergence process of firefly algorithm in the entire search space

Table 7 The number of times the objective function converges below 500 in 50 runs

Optimization algorithm	DE	BA	FA
The number of times	50	36	7

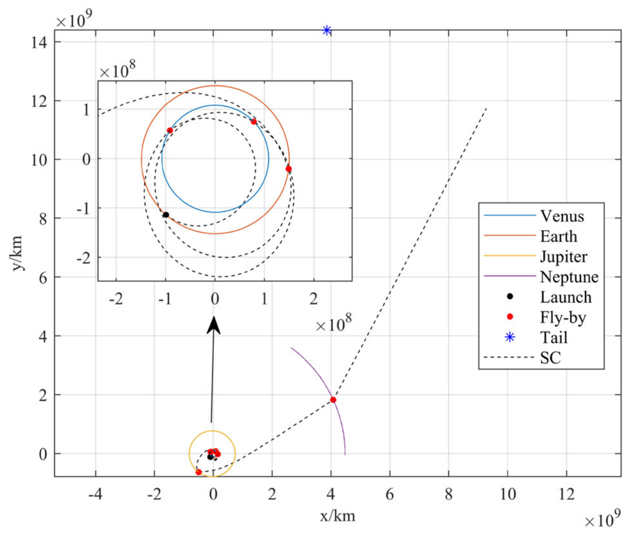


Fig. 18 The transfer trajectory of the optimal solution in the entire search space

conclusion, without any pruning of the search space, the convergence probability and efficiency of DE are both low.

From the results of 50 runs, it is clear that the solution with a total maneuver equal to 9.34 km/s is the optimal solution. The transfer trajectory of the optimal solution is shown in Fig. 18, and the relevant parameters of the optimal solution are shown in Table 8. The C_3 is 33.67 km^2/s^2 and the constraint $C_3 \leq 36\,\mathrm{km}^2/\mathrm{s}^2$ is satisfied. The position of spacecraft reaches 100.00 AU in an angle of 23.50 degrees with tail on October 1, 2049, so the constraints $r_{2049} \geq 100\mathrm{AU}$ and $\theta_{2049} \leq 45\,\mathrm{deg}$ is satisfied. The periapsis radius for the two VGA, EGA, JGA and NGA are $1.12R_{p1}$,$2.50R_{p2}$,$1.05R_{p3}$,$1.10R_{p4}$ and $1.29R_{p5}$ respectively, satisfying the constraints $r_{pi} \geq 1.05R_{pi}(i \leq 3)$ and $r_{pi} \geq 1.1R_{pi}(i \geq 4)$. The magnitude of all periapsis maneuvers is less than or equal to 5.00 km/s, and the magnitude of deep space maneuvers is zero. Therefore, the optimal solution satisfies all the constraints.

Trajectory optimization in the solution space box bounds

Since the analysis in the previous section has shown that the performance of DE is better than that of BA and FA, we only use DE for trajectory optimization in each solution space box bound after pruning, and compare the optimization results

Table 8 The value of the decision vector for the optimal solution in the entire search space

Event	Decision vector value		Maneuver magnitude or C_3(km^2/s^2)	Periapsis radius (R_{pi})
	Variables	Value		
Launch (Earth)	t_0 (y-m-d)	2025-05-08	33.67	0.00
1st gravity assist (VGA)	T_1 (days)	156.08	0.72	1.12
2nd gravity assist (VGA)	T_2 (days)	384.71	0.00	2.50
3rd gravity assist (EGA)	T_3 (days)	317.84	5.00	1.05
4th gravity assist (JGA)	T_4 (days)	958.34	3.62	1.10
5th gravity assist (NGA)	T_5 (days)	2250.54	0.00	1.29
	r_{pN} (R_{pN})	1.29		
	γ (rad)	− 1.58		
Deep-space maneuver	η	0.13		
	ΔV_{dsm} (km/s)	0.00		

Table 9 Results of 50 repeated runs in the solution space box bounds

Solution space	Total maneuver (km/s)			
	Minimum	Maximum	Average	Standard deviation
box1	12.72	15.79	14.20	0.72
box2	24.59	24.68	24.66	0.01
box3	10.54	10.54	10.54	0.00
box4	15.65	21.99	15.79	0.89
Entire search space	9.34	19.25	14.19	2.10

with those in the previous section. The population size of the algorithm is still NP = 200 and the maximum number of generations is $G_{\max}$ = 3000. Table 9 shows the results of 50 repeated runs in box1–4. It can be seen that whether the minimum, maximum, average or standard deviation, the optimization results of box3 are completely better than those of box1, box2 and box4. Therefore, we only discuss the optimization results in box3 in the following.

According to Table 9, although the minimum of the total maneuver in box3 is slightly larger than that in the entire search space, the maximum, the average and the standard deviation are all much smaller than those in the entire search space. The standard deviation reaches zero, that is, the magnitude of the total maneuver in each run converges to 10.54 km/s, with a convergence probability of 100%.

The convergence of the objective function for 50 runs in the solution space box3 is shown in Fig. 19. It can be seen that all the objective functions converge to less than 500 after 45 generations, which is much better than the result that the objective functions converge to less than 500 after 516 generations in the entire search space. It shows that the

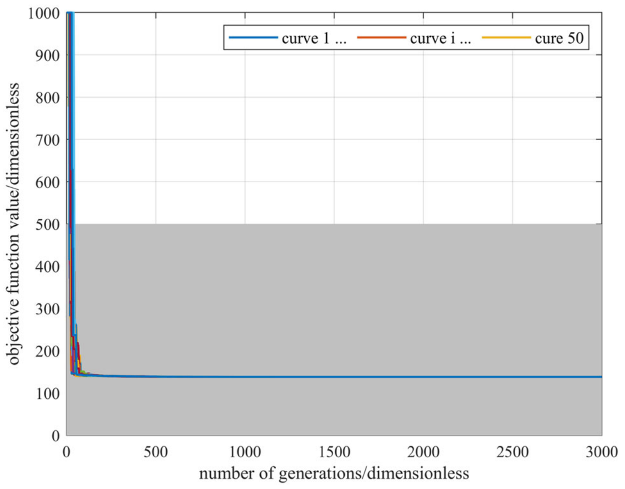


Fig. 19 Convergence process of differential evolution algorithm in the box3

optimization efficiency in the pruned solution space box3 is more than 11 times higher than that in the entire search space.

The transfer trajectory of the optimal solution in the solution space box3 is shown in Fig. 20, the relevant parameters

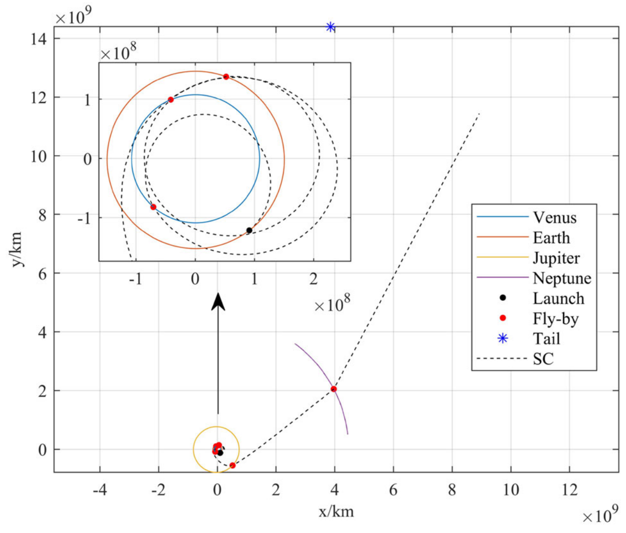


Fig. 20 The transfer trajectory of the optimal solution in the solution space box3

Table 11 Comparison of the optimization results

Parameter	The entire search space	Solution space box3
Generations of convergence to 500	516	45
Convergence probability	2%	100%
Total maneuver of optimal solution (km/s)	9.34	10.54
Periapsis maneuver of optimal solution (km/s)	9.34	9.33
Deep-space maneuver of optimal solution (km/s)	0.00	1.21

of the optimal solution are shown in Table 10. The C_3 can be calculated as 35.17 km^2/s^2. The position of spacecraft reaches 100.00 AU in an angle of 22.88 degrees with tail on October 1, 2049. The ratio of the periapsis radius to the planet radius is 1.05, 4.02, 1.05, 1.1, and 1.45 for VGA, VGA, EGA, JGA, and NGA, respectively. All these results satisfy the constraints.

Table 11 gives a comparison of the optimization results in the entire search space and in the solution space box3. The convergence efficiency of DE in the solution space box3 is more than 11 times higher than that in the entire search space. The probability that the objective function converges to the optimal solution in the solution space box3 is 50 times higher than that in the entire search space. Although the optimal total maneuver in the solution space box3 is slightly larger than that in the entire search space, the periapsis maneuver is actually superior to the latter.

Therefore, we can conclude that the improved gravity assist space pruning algorithm proposed in this paper not only guarantees the approximate optimal solution, but also greatly improves the convergence probability and convergence efficiency of trajectory optimization. These results illustrate the effectiveness of the improved gravity assist space pruning algorithm in the transfer trajectory optimization for solar system boundary exploration.

Table 10 The value of the decision vector for the optimal solution in the solution space box3

Event	Decision variables value		Maneuver magnitude or C_3(km^2/s^2)	Periapsis radius (R_{pi})
	Variables	Value		
Launch (Earth)	t_0 (y-m-d)	2028-07-28	5.93	0
1st gravity assist (VGA)	T_1 (days)	152.60	0.42	1.05
2nd gravity assist (VGA)	T_2 (days)	377.40	0.61	4.02
3rd gravity assist (EGA)	T_3 (days)	325.00	5.64	1.05
4th gravity assist (JGA)	T_4 (days)	781.91	2.67	1.1
5th gravity assist (NGA)	T_5 (days)	1776.90	0.00	1.45
	r_{pN} (R_{pN})	1.45		
	γ (rad)	− 1.58		
Deep-space maneuver	η	0.01		
	ΔV_{dsm} (km/s)	1.21		

Conclusion

In this paper, we systematically study the transfer trajectory optimization for China's solar system boundary exploration mission and application of the proposed gravity assist space pruning algorithm to the E-V-V-E-J-N-Tail flight sequence for the first time. According to the characteristics of the mission, the transfer trajectory optimization model for the solar system boundary exploration is established, and then the improved gravity assist space pruning algorithm is proposed and discussed in detail. The gravity assist pruning criterion and a new pruning procedure are presented. A new solution space box bounds shape combining rectangular and rhombic is proposed to determine the range of decision variables. The method of determining the number of solution space box bounds and automatically determining the solution space box bounds is given. The optimization performance of differential evolution algorithm, bat algorithm and firefly algorithm for the given problem is compared. And the differential evolution algorithm is used to optimize the decision vector both in the entire search space and solution space box bounds. The simulation results show that the differential evolution algorithm is more efficient than bat algorithm and firefly algorithm. The proposed gravity assist space pruning algorithm is effective in the solar system boundary exploration mission, which is conducive to improving the efficiency of subsequent optimization to more than 11 times, and improving the convergence probability of the objective function by 50 of times.

In this study, not only the flight distance and direction requirements of the solar system boundary exploration mission are met, but also the parameters in line with the actual engineering constraints are adopted. Therefore, the results of this study are realizable in practical engineering, and the relevant conclusions can provide a reference for practical engineering missions. In addition, this study shows that the pruning results of Venus gravity assist are less sensitive to the maximum thrust constraint than those of Earth and Jupiter gravity assist, so the periapsis maneuver constraint of Venus gravity assist can be more stringent, while the periapsis maneuver constraint of Jupiter gravity assist should be relaxed.

The transfer trajectory optimization model established in this paper for solar system boundary exploration adopts $N - 1$ times of periapsis maneuver and one time of deep space maneuver, and the proposed gravity assist space pruning algorithm is mainly aimed at the situation of periapsis maneuver. However, in practical engineering applications, it is difficult to accurately locate the position of the maneuver as the periapsis in a strict sense, which limits the application scenarios of the improved gravity assist space pruning algorithm proposed in this paper. Therefore, in further research, the transfer trajectory optimization model for solar system boundary exploration using N times of deep space maneuvers and none periapsis maneuvers should be considered. In this case, the dimensions of the decision variables and the volume of the search space are further increased, so a new gravity assist space pruning algorithm for deep space maneuvers should be developed. In addition, the total maneuver magnitude of the optimal transfer trajectory obtained in this paper reaches the order of 10.00 km/s. Because the specific impulse of the pulse propulsion system used is small, the fuel consumption corresponding to the total maneuver of 10.00 km/s is large. To further reduce the fuel consumption, the transfer trajectory optimization of solar system boundary exploration in the case of low-thrust propulsion system should be considered in the follow-up study.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Grant No. 12150007). The authors would like to thank Lang Lu and Lantian Chang for helpful discussions on topics related to this work.

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Dialynas K, Krimigis S, Mitchell D, Decker R et al (2017) The bubble-like shape of the heliosphere observed by Voyager and Cassini. Nat Astron 1:1–7. https://doi.org/10.1038/s41550-017-0115
2. Hall C (1974) Pioneer 10. Science 183:301–302. https://doi.org/10.1126/science.183.4122.301
3. Acuña M, Ness N (1980) The magnetic field of Saturn: Pioneer 11 observations. Science 207:444–446. https://doi.org/10.1126/science.207.4429.444
4. Dyal P (1993) Future operations of Pioneer 10 and 11. Adv Space Res 13:267–274. https://doi.org/10.1016/0273-1177(93)90419-C
5. Stone E, Cummings A, McDonald F et al (2005) Voyager 1 explores the termination shock region and the Heliosheath Beyond. Science 309:2017–2020. https://doi.org/10.1126/science.1117684

6. Richardson J, Belcher J, Garcia-Galindo P, Burlaga L (2019) Voyager 2 plasma observations of the heliopause and interstellar medium. Nat Astron 3:1019–1023. https://doi.org/10.1038/s41550-019-0929-2
7. Hosteaux S, Rodiguez L, Poedts S (2022) Analysis of Voyager 1 and Voyager 2 in situ CME observations. Adv Space Res 70:1684–1719. https://doi.org/10.1016/j.asr.2022.03.005
8. Guo Y, Farquhar R (2007) New horizons mission design. Space Sci Rev 140:49–74. https://doi.org/10.1007/s11214-007-9242-y
9. Wu W, Yu D, Huang J et al (2019) Exploring the solar system boundary. Sci Sin Inform 49:1–16. https://doi.org/10.1360/n112018-00273
10. Ceriotti M, Vasile M (2010) MGA trajectory planning with an ACO-inspired algorithm. Acta Astronaut 67:1202–1217. https://doi.org/10.1016/j.actaastro.2010.07.001
11. Jovanovic L, Jovanovic G, Perisic M et al (2023) The explainable potential of coupling metaheuristics-optimized-XGBoost and SHAP in revealing VOCs' environmental fate. Atmosphere 14:109. https://doi.org/10.3390/atmos14010109
12. Stojanovic V, Nedic NA (2016) Nature inspired parameter tuning approach to cascade control for hydraulically driven parallel robot platform. J Optim Theory Appl 168:332–347. https://doi.org/10.1007/s10957-015-0706-z
13. Nedic N, Stojanovic V, Djordjevic V (2015) Optimal control of hydraulically driven parallel robot platform based on firefly algorithm. Nonlinear Dyn 82:1457–1473. https://doi.org/10.1007/s11071-015-2252-5
14. Fang H, Zhu G, Stojanovic V et al (2021) Adaptive optimization algorithm for nonlinear Markov jump systems with partial unknown dynamics. Int J Robust Nonlinear Control 31:2126–2140. https://doi.org/10.1002/rnc.5350
15. Tuba E, Bacanin N (2015) An algorithm for handwritten digit recognition using projection histograms and SVM classifier. In: 23rd telecommunications forum Telfor (TELFOR). Belgrade, Serbia, pp 464–467. https://doi.org/10.1109/TELFOR.2015.7377507
16. Malakar S, Ghosh M, Bhowmik S et al (2020) A GA based hierarchical feature selection approach for handwritten word recognition. Neural Comput Appl 32:2533–2552. https://doi.org/10.1007/s00521-018-3937-8
17. Wall B, Conway B (2008) Genetic algorithms applied to the solution of hybrid optimal control problems in astrodynamics. J Glob Optim 44:493–508. https://doi.org/10.1007/s10898-008-9352-4
18. Rosa S, Casalino L (2009) Cooperative evolutionary algorithm for space trajectory optimization. Celest Mech Dyn Astron 105:211–227. https://doi.org/10.1007/s10569-009-9223-4
19. Vasile M, Minisci E, Locatelli M (2010) Analysis of some global optimization algorithms for space trajectory design. J Spacecr Rocket 47:334–344. https://doi.org/10.2514/1.45742
20. Myatt D, Becerra V, Nasuto S, Bishop J (2004) Advanced global optimisation tools for mission analysis and design. In: Tech. Rep. 03-4101a. The Advanced Concepts Team, European Space Agency, pp 1–154. Retrieved 14 July 2022. https://www.esa.int/gsp/ACT/doc/ARI/ARI%20Study%20Report/ACT-RPT-MAD-ARI-03-4101a-GlobalOptimisation-Reading.pdf
21. Izzo D, Becerra V, Myatt D et al (2006) Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories. J Glob Optim 38:283–296. https://doi.org/10.1007/s10898-006-9106-0
22. Becerra V, Nasuto S, Anderson J et al (2007) Search space pruning and global optimization of multiple gravity assist trajectories with deep space manoeuvres. In: 2007 congress on evolutionary computation. IEEE, Singapore, pp 957–964. https://doi.org/10.1109/cec.2007.4424573
23. Izzo D (2010) Global optimization and space pruning for spacecraft trajectory design. In: Conway BA (ed) Spacecraft trajectory optimization. Cambridge University Press, Cambridge, pp 178–201. https://doi.org/10.1017/cbo9780511778025.008
24. Armellin R, Di L, Topputo F et al (2009) Gravity assist space pruning based on differential algebra. Celest Mech Dyn Astron 106:1–24. https://doi.org/10.1007/s10569-009-9235-0
25. Schütze O, Vasile M, Junge O et al (2009) Designing optimal low-thrust gravity-assist trajectories using space pruning and a multi-objective approach. Eng Optim 41:155–181. https://doi.org/10.1080/03052150802391734
26. Yang D, Xu B, Gao Y (2015) Search space pruning based on image tools for preliminary interplanetary trajectory design. Trans Nanjing Univ Aeronaut Astron 32:530–540. https://doi.org/10.16356/j.1005-1120.2015.05.530
27. Vasile M, Ceriotti M, Radice G (2007) Global trajectory optimisation: can we prune the solution space when considering deep space manoeuvres? In: Technical Report 06-4101c. The Advanced Concepts Team, European Space Agency, pp 2–117. Retrieved 14 July 2022. https://www.esa.int/gsp/ACT/doc/ARI/ARI%20Study%20Report/ACT-RPT-MAD-ARI-06-4101-CanWePrune-GlasgowReading.pdf
28. Zazzera F, Lavagna M, Armellin R et al (2007) Global trajectory optimisation: can we prune the solution space when considering deep space manoeuvres? Technical Report 06-4101b. The Advanced Concepts Team, European Space Agency, pp 1–124. Retrieved July 14, 2022. https://www.esa.int/gsp/ACT/doc/ARI/ARI%20Study%20Report/ACT-RPT-MAD-ARI-06-4101-CanWePrune-Politecnico-di-Milano.pdf
29. Olympio J, Marmorat J (2007) Global trajectory optimisation: can we prune the solution space when considering deep space manoeuvres? Technical Report 06-4101a. The Advanced Concepts Team, European Space Agency, pp 1–79. Retrieved 14 July 2022. https://www.esa.int/gsp/ACT/doc/ARI/ARI%20Study%20Report/ACT-RPT-MAD-ARI-06-4101-CanWePrune-Ecole-des-Mines.pdf
30. Englander J, Conway B, Williams T (2012) Automated mission planning via evolutionary algorithms. J Guid Control Dyn 35:1878–1887. https://doi.org/10.2514/1.54101
31. Vinko´ T, Izzo D (2008) Global optimization heuristics and test problems for preliminary spacecraft trajectory design. Technical Report ACT-TNT-MAD-GOHTPPSTD. European Space Agency, pp 1–10. Retrieved 14 July 2022. https://www.esa.int/gsp/ACT/doc/INF/pub/ACT-TNT-INF-2008-GOHTPPSTD.pdf
32. Wang H, Liang M, Sun C et al (2021) Multiple-strategy learning particle swarm optimization for large-scale optimization problems. Complex Intell Syst 7:1–16. https://doi.org/10.1007/s40747-020-00148-1
33. Kapnopoulos A, Alexandridis A (2022) A cooperative particle swarm optimization approach for tuning an MPC-based quadrotor trajectory tracking scheme. Aerosp Sci Technol 127:107725. https://doi.org/10.1016/j.ast.2022.107725
34. Chai R, Savvaris A, Tsourdos A et al (2019) A review of optimization techniques in spacecraft flight trajectory design. Progress Aerosp Sci 109:100543. https://doi.org/10.1016/j.paerosci.2019.05.003
35. Mohamed A (2017) Solving large-scale global optimization problems using enhanced adaptive differential evolution algorithm. Complex Intell Syst 3:205–231. https://doi.org/10.1007/s40747-017-0041-0
36. Li Z, Li X (2019) A multi-objective binary-encoding differential evolution algorithm for proactive scheduling of agile earth observation satellites. Adv Space Res 63:3258–3269. https://doi.org/10.1016/j.asr.2019.01.043
37. Choi J, Lee J, Park C (2022) Deep-space trajectory optimizations using differential evolution with self-learning. Acta Astronaut 191:258–269. https://doi.org/10.1016/j.actaastro.2021.11.014

38. Yang X, Gandomi A (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput 29:464–483. https://doi.org/10.1108/02644401211235834
39. Yang X, He X (2013) Firefly algorithm: recent advances and applications. Int J Swarm Intell 1:36–50. https://doi.org/10.1504/IJSI.2013.055801
40. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1:3–18. https://doi.org/10.1016/j.swevo.2011.02.002