Acta Astronautica 202 (2023) 876–888

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

GTOC 11: Results found at Beijing Institute of Technology and China Academy of Space Technology

Shaozhao Lu [a], Yao Zhang [a,*], Jingrui Zhang [a], Han Cai [a], Rui Qi [a], Xingang Li [b], Yu Qi [b], Qian Xiao [a], Ao Shen [a], Tiantian Zhang [a], Kunpeng Zhang [a], Ji Ye [a], Zechuan Tian [a]

[a] *School of Aerospace Engineering, Beijing Institute of Technology, 100081 Beijing, People's Republic of China*

[b] *Institute of Telecommunication and Navigation Satellite, China Academy of Space Technology, 100094 Beijing, People's Republic of China*

ARTICLE INFO

Keywords:
GTOC
Beam search
Parallel computing
Coordinate descent method
Trajectory optimization

ABSTRACT

This paper presents the crucial method for the results found at Beijing Institute of Technology and China Academy of Space Technology in the 11th Global Trajectory Optimization Competition (GTOC-11). To build 12 solar-power stations placed in the "Dyson Ring" orbit around the Sun, GTOC-11 focuses on the flyby trajectory design, low-thrust trajectory optimization, and target assignment problem. The mission with time and state constraints causes many difficulties in the design and optimization, which is performed in the following four steps. The first step is to design the orbital elements of the "Dyson Ring" by the grid search method. An approximate model is utilized to evaluate the minimum transfer time for each candidate at first, and the accurate solutions are obtained by the indirect method. The second step involves the Mother Ship trajectories design. A large quantity of earth-to-asteroid and asteroid-to-asteroid subsequences are generated by parallel beam search. The complete trajectories of Mother Ship are designed by connecting the subsequences using beam search. The third step involves the station assignment problem, which is required to design the asteroid transfer device activation epoch and the target station assignment. Based on the results above, the last step involves the rendezvous trajectories optimization to obtain the accurate thrust direction profile. Finally, the submitted and post-competition results are reported.

1. Introduction

In the 11th Global trajectory optimization competition (GTOC-11) [1], the challenging problem is to build 12 stations in a "Dyson Ring" orbit, using asteroids selected from over 83 000 candidates and transferred to those stations via asteroid transfer devices (ATDs). The participants were required to design the "Dyson Ring" orbital elements, the Mother Ship trajectories, the assignment of each flyby asteroids, and low-thrust rendezvous trajectories. The competition aims to minimize the sum of ΔV used by each Mother Ship, maximize the minimal mass among all building stations, and design the semi-major axis of the "Dyson Ring" orbit as small as possible.

GTOC-11 inherits the features of previous competitions [2–5], which are required to be optimized in huge search space and allowed to use different strategies and methods, but obtaining an excellent performance index is quite difficult [6]. Finding a feasible solution of GTOC-11 is quite complicated because it involves flyby, rendezvous, and target assignment problem. Directly obtaining an excellent solution with a single step, however, is not practical. The problem is intentionally divided into four sub-problems by our team: the "Dyson Ring" orbital elements design, the Mother Ship trajectories design, the assignment of the flyby asteroids, and the low-thrust rendezvous trajectories optimization.

In order to get higher scores, several methods were applied, especially in the flyby sequences design, which determines the upper bound of the performance index in this competition. Previous GTOC and China trajectory optimization competitions (CTOC) also involve flyby or rendezvous sequences optimization problem, such as GTOC-5: maximizing the number of visited asteroids with revisited [3], GTOC-7: maximizing the number of visited asteroid using probes with electric propulsion system [4], GTOC-9: the removal of long chains of multiple debris [5], and CTOC-8A: optimization of the selection of debris and their rendezvous sequences [7]. The branch-and-bound algorithm is widely used in the sequence searching in the previous competitions. Jet Propulsion Laboratory in GTOC-5 adopted the branch-and-bound algorithm with simplified model to search the sequences [8]. Beijing Institute of Technology in CTOC-8A used the branch-and-bound algorithm with greedy strategy to search the debris sequences [7]. However, these problems where several variables must be optimized internally,

* Corresponding author.
E-mail address: zhangyao@bit.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.actaastro.2022.07.039
Received 2 March 2022; Received in revised form 11 May 2022; Accepted 26 July 2022
Available online 30 July 2022

such as GTOC-7, CTOC-9A [9] and CTOC-10 [10], an evolutionary algorithm is more suitable than branch-and-bound algorithm. Nanjing University of Aeronautics and Astronautics used genetic algorithms to globally optimize flight time and orbital elements [11]. An evolving elitist club algorithm was proposed and applied to multi-target space debris removal missions in the GTOC-9A post-competition results [12]. In GTOC-11, traditional beam search is apt to fall into a local optimum owing to the large number of candidates. The parallel beam search was proposed by our team to efficiently generate a subsequence database, which was utilized to design the Mother Ship trajectories.

After completing the step of designing the Mother Ship trajectories, it is necessary to optimize the phase of the first station, station building windows, and flyby asteroids assignment. General evolutionary algorithms, such as genetic algorithm (GA) [13], particle swarm optimization (PSO) [14], are suitable for the first two items. In CTOC-7B, Xi'an Satellite Control Center used the hybrid PSO algorithm to optimize all trajectories by connecting every configuration [15]. In GTOC-9, National University of Defense Technology used a hybrid-encoding genetic algorithm to simultaneously determine the sequence and the referenced transfer time in a single rendezvous mission [6,16]. Furthermore, ant colony optimization (ACO) [17] does well in the traveling salesman problem [18]. In this competition, a bilevel optimization method was proposed by our team. The PSO algorithm was applied to determine the phase of the first station and station building windows, while the coordinate descent method is utilized to solve the station assignment problem for each particle.

The low-thrust rendezvous trajectory optimization is the key to the problem. An unreasonable orbital transfer design method will affect the remaining mass of the flyby asteroids, or even cannot meet the accuracy of the rendezvous condition with the station. Based on the high accuracy integrator Runge–Kutta–Fehlberg 7(8) [19] and the nonlinear equations solver *cminpack* [20], the low-thrust trajectory optimization problem is transformed into a two-point boundary value problem (TPBVP). Given the target stations for the asteroids, however, it is difficult to solve the rendezvous trajectory directly, which involves the determination of the departure epoch and the optimization of the transfer time. Consequently, the free and time-varying terminal phase problem are both considered. Firstly, the free terminal phase problem is solved with different departure epoch, and the optimal transfer time and corresponding arrival phase are obtained. Based on the results above, the departure epoch is determined by judging the coincidence condition of the arrival phase of the asteroid and the target station phase. Then, the time-varying terminal phase problem for rendezvousing with the target station is solved successfully by the reliable departure epoch.

Aiming to present methods and results from Beijing Institute of Technology and China Academy of Space Technology for GTOC-11, the rest of this paper is organized as follows. The preliminary analysis is performed in Section 2. The methods are introduced in Section 3. The competition results, including submitted and post-competition results, are described in Section 4. Finally the concluding remarks are provided in Section 5.

2. Preliminary analysis

The task of this competition is to build 12 stations in the "Dyson Ring" orbit, using flyby asteroids with ATDs. It is necessary to give consideration to the influences of solar radiation efficiency and construction cost when builds these stations. Obviously, the stations' quality as well as the distance between the stations and the sun mainly affect the solar radiation efficiency while the fuel consumption of the Mother Ships represents the construction cost. Accordingly, the performance index to be maximized is as follows:

$$J = B \cdot \frac{10^{-10} \cdot M_{\min}}{a_{Dyson}^2 \sum_{k=1}^{10} \left(1 + \Delta V_k^{Total}/50\right)^2} \tag{1}$$

where B represents the submission time reward coefficient; a_{Dyson} denotes the radius of the designed "Dyson Ring" orbit; ΔV_k^{Total} represents the total fuel cost of the kth Mother Ship; $M_{\min}$ represents the minimum mass of 12 stations. The minimal mass among all building stations is defined as:

$$M_{\min} = \min\left\{M_j | j = 1, 2, \ldots, 12\right\} \tag{2}$$

In the following sections, based on the four aspects as below, the attention is on decoupling the "Dyson Sphere" building problem into several subproblems and obtaining the optimal solution of these subproblems successively, so a global feasible solution would be generated by connecting the subproblems.

(1) Designing the radius of the "Dyson Ring" orbit: The performance index is inversely proportional to the square of radius. Intuitively a larger radius needs to be compensated by a greater station mass, and vice versa. The relationship between the "Dyson Ring" radius and the remaining mass of each asteroid need to be explored. Section 3.1 determines the orbital elements of the "Dyson Ring" by calculating the characteristic mass of each asteroid with the grid search method.

(2) Optimizing the Mother Ship trajectories: Each Mother Ship is required to flyby a sequence of asteroids with small velocity increment. However, the flyby asteroids are selected from over 83 000 candidates, which easily lead to fall into a local optimum. Since the stations' mass is a critical parameter in this problem, the remaining mass of asteroids and the length of flyby sequence will be also viewed as important factors in sequence determination algorithm development. Section 3.2 unveils the parallel beam search sequence determination algorithm, which includes database generation and subsequences connection.

(3) Assigning flyby asteroids to the stations: Since the performance index is related to the minimum mass of 12 stations, a rational stations building strategy needs to be designed. Four aspects are required to be addressed: the station building sequence, the station building windows, the assignment of flyby asteroids and the corresponding ATD activation epoch. Section 3.3 shows that a bilevel optimization problem, by means of particle swarm optimization and coordinate descent method, would generate the assignment scheme.

(4) Low-thrust pursue trajectories optimization: High accuracy is required in the asteroids transfer and rendezvous processes. However, the pursue problem causes difficulties for the low-thrust trajectories design, especially in guessing the transfer time and initial shooting variables at the ATD activation epoch. Section 3.4 transforms the pursue problem into TPBVP, where the initial shooting variables and transfer time are obtained by the solution to the free terminal phase problem described in Section 3.3.

3. Description of the methods

3.1. Determining the "Dyson Ring" elements

Before determining the "Dyson Ring" elements, a statistical analysis on the distribution of the candidates' orbital elements is conducted. As shown in Fig. 1, the semi-axis of the candidates is concentrated in 2.3–3.4 AU with the median of 2.64 AU. Similarly, the eccentricity and inclination are mainly distributed in 0~0.4 and 0~20°, respectively, while the right ascension of the ascending node (RAAN), argument of perigee, and mean anomaly are close to mean distribution. It is difficult to determine the orbital elements of the "Dyson Ring" from the distribution of the candidates' orbital elements.

There are four orbital elements required to be designed: a_{Dyson}, i_{Dyson}, Ω_{Dyson}, and φ_j ($j = 1, 2, \ldots, 12$). In this section, the grid search method is applied in determining the first three elements, while the argument of latitude φ_j (i.e., phase) will be solved iteratively in Section 3.3. The RAAN of the "Dyson Ring" is set to be 0° in order to reduce the search range. For convenience, the characteristic mass is

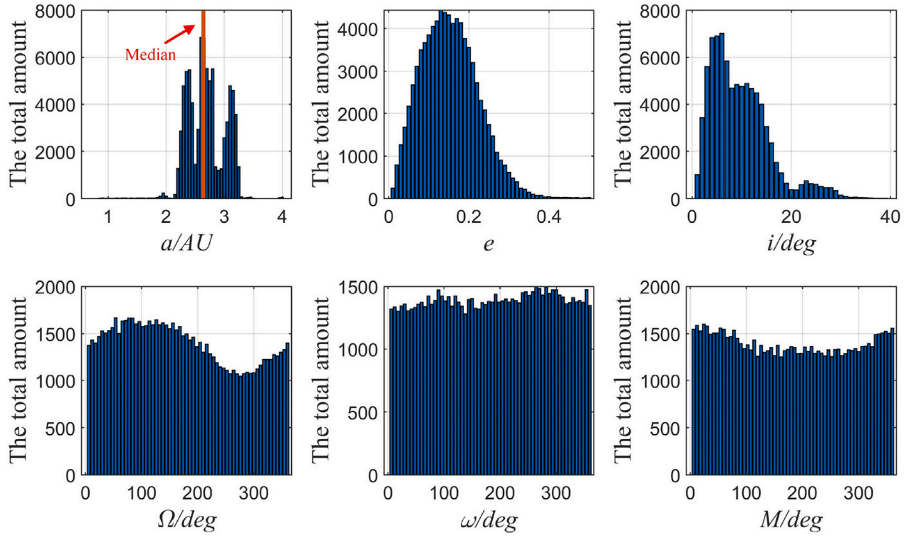


Fig. 1. Candidates analysis.

first defined to explore the relationship between the remaining mass of each asteroid and the radius of the "Dyson Ring", which is defined as:

$$\bar{M}_f = \frac{10^{-10} M_f}{a_{Dyson}^2} = \frac{10^{-10}}{a_{Dyson}^2}\left(1 - \alpha \Delta t\right) m_0^{ast} \tag{3}$$

where $\alpha = 6e{-}9\ \mathrm{s}^{-1}$ is the proportion coefficient; Δt (unit s) refers to the transfer duration from the ATDs activation epoch to the arrival time at the station; m_0^{ast} is the initial mass of the asteroid.

In order to obtain the elements of the "Dyson Ring" accurately, the low-thrust time-optimal control problem is established at first. Considering the computational burden, an approximate method is proposed to evaluate the transfer time before solving the optimal control problem. Based on the accurate and approximate method, the grid search is applied to find the optimal combination of a_{Dyson} and i_{Dyson}.

3.1.1. Formulation of free terminal phase time-optimal control problem

In this section, the free terminal phase low-thrust trajectory is formulated to analyze the relationship between the characteristic mass and the "Dyson Ring" elements. The maneuvers of ATDs on asteroids are modeled as continuous-thrust maneuvers with a fixed magnitude of acceleration Γ, and the dynamics of asteroid subjects to Sun's gravity and low thruster only. The modified equinoctial elements (MEEs) are employed to describe the dynamical model, which can be written as:

$$\dot{\boldsymbol{x}} = \boldsymbol{M}\Gamma\boldsymbol{\gamma} + \boldsymbol{D} \tag{4}$$

where $\boldsymbol{x} = [p, f, g, h, k, L]^T$, $p = a\left(1 - e^2\right)$, $f = e\cos(\omega + \Omega)$, $g = e\sin(\omega + \Omega)$, $h = \tan(i/2)\cos\Omega$, $k = \tan(i/2)\sin\Omega$, $L = \Omega + \omega + \upsilon$; $\boldsymbol{\gamma}$ is the unit vector in thrust direction; $\boldsymbol{M}$ is a 6 × 3 matrix; $\boldsymbol{D}$ is a 6 × 1 vector. The specific form of these matrices are shown in Ref. [21].

The remaining mass of each asteroid depends on its initial mass and the transfer time from activated epoch to rendezvous epoch. To maximize the remaining mass, subject to the dynamical model, the time-optimal control problem is expressed by:

$$J = \int_{t_0}^{t_f} 1 dt \tag{5}$$

The boundary constraints of the considered problem are:

$$\boldsymbol{x}\left(t_0\right) = \boldsymbol{x}_0 \tag{6}$$

$$p\left(t_f\right) = a_{Dyson}, f\left(t_f\right) = 0, g\left(t_f\right) = 0, h\left(t_f\right) = \tan\left(\frac{i_{Dyson}}{2}\right), k\left(t_f\right) = 0 \tag{7}$$

where $e_{Dyson} = 0$ and the RAAN of the "Dyson Ring" is set to be 0° for simplicity; the final true longitudes $L\left(t_f\right)$ is free. It is worthy to note that the free terminal true longitudes condition simplifies the hard-constraint where the flyby asteroids are supposed to rendezvous with the target station. The error between the actual transfer time and the simplification mainly comes from the orbital phase where the ATD is activated, which is acceptable for the reason that the phases of the candidates are close to mean distribution at a given epoch shown in Fig. 1.

The Hamiltonian is formulated as:

$$H = \boldsymbol{\lambda}_{\boldsymbol{x}}^T\left(\boldsymbol{M}\Gamma\boldsymbol{\gamma} + \boldsymbol{D}\right) + 1 \tag{8}$$

where $\boldsymbol{\lambda}_{\boldsymbol{x}} = \left[\lambda_p, \lambda_f, \lambda_g, \lambda_h, \lambda_k, \lambda_L\right]^T$ denotes the costate vector. Based on the Pontryagin's Minimum Principle [22], the optimal thrust direction $\boldsymbol{\gamma}^*$ is expressed by:

$$\boldsymbol{\gamma}^* = -\frac{\left(\boldsymbol{\lambda}_{\boldsymbol{x}}^T\boldsymbol{M}\right)^T}{\left\|\boldsymbol{\lambda}_{\boldsymbol{x}}^T\boldsymbol{M}\right\|} \tag{9}$$

Associated with the dynamical equation (4), the costate equations are derived by $\dot{\boldsymbol{\lambda}}_{\boldsymbol{x}} = -\frac{\partial H}{\partial \boldsymbol{x}}$ and substituted with the above optimal thrust direction:

$$\dot{\boldsymbol{\lambda}}_{\boldsymbol{x}} = -\left(\boldsymbol{\lambda}_{\boldsymbol{x}}^T\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{x}}\boldsymbol{\gamma} + \boldsymbol{\lambda}_{\boldsymbol{x}}^T\frac{\partial \boldsymbol{D}}{\partial \boldsymbol{x}}\right) = \boldsymbol{\lambda}_{\boldsymbol{x}}^T\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{x}}\frac{\left(\boldsymbol{\lambda}_{\boldsymbol{x}}^T\boldsymbol{M}\right)^T}{\left\|\boldsymbol{\lambda}_{\boldsymbol{x}}^T\boldsymbol{M}\right\|} - \boldsymbol{\lambda}_{\boldsymbol{x}}^T\frac{\partial \boldsymbol{D}}{\partial \boldsymbol{x}} \tag{10}$$

According to the transversality condition, the final true longitude costate should be zero due to the free condition:

$$\lambda_L\left(t_f\right) = 0 \tag{11}$$

For a free final-time problem where the problem data are not explicitly dependent on time, the Hamiltonian is identically zero along the optimal solution [22], and thus:

$$H\left(\boldsymbol{x}^*\left(t_f\right), \boldsymbol{\lambda}^*\left(t_f\right), \boldsymbol{\gamma}^*, t_f\right) = 0 \tag{12}$$

Given initial states $\boldsymbol{x}\left(t_0\right)$, initial costate values $\lambda\left(t_0\right)$, and transfer time t_f, integrating the Eqs. (4) and (10) from a given epoch t_0, if the final states and costates satisfy Eqs. (7), (11), and (12), the transfer time t_f satisfies the necessary optimality conditions. Consequently, the time-optimal control problem is transformed into a TPBVP, which is solved by shooting method. The shooting variables are wrapped in:

$$\boldsymbol{z} \overset{\Delta}{=} \left[\lambda_p\left(t_0\right), \lambda_f\left(t_0\right), \lambda_g\left(t_0\right), \lambda_h\left(t_0\right), \lambda_k\left(t_0\right), \lambda_L\left(t_0\right), t_f\right]^T \tag{13}$$

Consequently, the shooting function is built as:

$$\boldsymbol{f}(\boldsymbol{z}) = \left[p\left(t_f\right) - a_{Dyson}, f\left(t_f\right), g\left(t_f\right), h\left(t_f\right) - \tan\frac{i_{Dyson}}{2}, k\left(t_f\right), \lambda_L\left(t_f\right), H\left(t_f\right)\right]^T = \mathbf{0} \tag{14}$$

Thus, a TPBVP associated with time-optimal control problem with free terminal phase condition is obtained, and the *cminpack* solver [20] is employed to solve this problem. The techniques of costate normalization [23], thrust homotopy, and Runge–Kutta–Fehlberg 7 (8) integrator [19] are used to augment the convergence of the algorithm.

3.1.2. Determining the “Dyson Ring” elements using grid search method

After formulating the TPBVP, the next step is to solve the problem in different radius and inclination combinations to obtain the optimal transfer time. The search range of radius and inclination are $\left[a_{Dyson}^{min} : a_{Dyson}^{h} : a_{Dyson}^{max}\right]$ and $\left[i_{Dyson}^{min} : i_{Dyson}^{h} : i_{Dyson}^{max}\right]$, where a_{Dyson}^{h} and i_{Dyson}^{h} denote the search step respectively. For each combination, there are more than 83 000 TPBVPs need to be solved. Therefore, it will take several days to finish the calculation even though one TPBVP solved by *cminpack* is less than 200 ms.

One way to reduce the computational burden of the grid search is to evaluate the minimum transfer time before solving the TPBVP by an analytical method. Edelbaum [24] considered three special cases about minimum-time all-propulsive transfer between given orbits. Lorenzo [25] extended the case of the simultaneous changes of a, e, and i. With the assumption that the changes of a and e are independent, the total ΔV can be estimated as [25]:

$$\Delta V = \sqrt{\left(k_a \Delta a\right)^2 + \left(k_e \Delta e\right)^2 + \left(k_i \Delta i\right)^2} \tag{15}$$

where $\Delta e = \sqrt{\Delta e_x^2 + \Delta e_y^2}$, $\Delta i = \sqrt{\Delta i_x^2 + \Delta i_y^2}$, $e_x = e\cos(\Omega + \omega)$, $e_y = e\sin(\Omega + \omega)$, $i_x = i\cos\Omega$, and $i_y = i\sin\Omega$. It should be noted that the ΔV takes changes of a, e, i, Ω, and ω into account. The cost coefficients

$$k_a = \frac{V_{avg}}{2a_{avg}}, k_e = 0.649V_{avg}, k_i = \frac{\pi}{2}V_{avg} \tag{16}$$

are evaluated by calculating the average values between the initial and final orbits:

$$a_{avg} = \frac{a_1 + a_2}{2}, V_{avg} = \sqrt{\frac{\mu}{a_{avg}}} \tag{17}$$

Based on the accurate and approximate method, the “Dyson Ring” elements determination algorithm is shown in algorithm 1. In line 1, e_{Dyson}, Ω_{Dyson}, and ω_{Dyson} are initialized by 0. Furthermore, in line 6, the minimum mass constant $\mathcal{M}$ is adopted to select big initial mass asteroids and reduce the computational burden. If $m_0^{ast} \geq \mathcal{M}$, the total ΔV of each big mass asteroid is evaluated by Eq. (15), and the estimated transfer time t_e is obtained in line 8. In line 9, the constant coefficient $\eta \in [0, 1]$ is adopted to select the big remaining mass asteroids, while the asteroid whose minimum transfer time exceeds $\eta\frac{1.0}{\alpha}$ would not be carried out the TPBVP. As a result, for each a and i combination, the characteristic mass $\bar{M}_f$ of the qualified candidates would be saved in database.

In this simulation, $a_{Dyson}^{min} = 1.0AU$, $a_{Dyson}^{h} = 0.5AU$, $a_{Dyson}^{max} = 1.4AU$, $i_{Dyson}^{min} = 0°$, $i_{Dyson}^{h} = 2°$, $i_{Dyson}^{max} = 14°$, and the constant $\mathcal{M} = 2.0e13$. The departure epoch $t_0 = 95900MJD$. As shown in Fig. 2, the “Dyson Ring” elements determination algorithm is employed to make statistical analysis on the total amount of asteroid whose final characteristic mass exceed the given lower bound.

According to the statistical results, under the same radius, the number of asteroids decreases as the inclination increases. From this view, it is intuitional to choose the inclination ranging from 0° to 4°. As the lower bound increases, we pay more attention to the asteroids with larger characteristic mass. When $\bar{M}_f \geq 5000$ and $\bar{M}_f \geq 6000$, the radius of $1.0 \sim 1.2$ AU with inclination $0 \sim 2°$ are more competitive than other combinations. Therefore, combined with the above analysis, which considers the total amount of qualified asteroids in different elements combinations and different lower bound, the “Dyson Ring” elements are designed as follows:

Algorithm 1 “Dyson Ring” Elements Determination Algorithm

```
Input: Grid search interval $\left[a_{Dyson}^{min} : a_{Dyson}^{h} : a_{Dyson}^{max}\right]$ and $\left[i_{Dyson}^{min} : i_{Dyson}^{h} : i_{Dyson}^{max}\right]$.
Output: The minimum transfer time of each asteroid in different elements combinations.
1: Initialization: $e_{Dyson} \leftarrow 0$, $\Omega_{Dyson} \leftarrow 0$, $\omega_{Dyson} \leftarrow 0$;
2: for $a = a_{Dyson}^{min}$; $a \leq a_{Dyson}^{max}$; $a = a + a_{Dyson}^{h}$ do
3:     for $i = i_{Dyson}^{min}$; $i \leq i_{Dyson}^{max}$; $i = i + i_{Dyson}^{h}$ do
4:         $a_{Dyson} \leftarrow a$, $i_{Dyson} \leftarrow i$;
5:         for each asteroid $\in$ Candidates do
6:             if $m_0^{ast} \geq \mathcal{M}$ then
7:                 Evaluate the total $\Delta V$ by Eq. (15);
8:                 Calculate the estimated transfer time $t_e$: $t_e = \Delta V / \Gamma$;
9:                 if $t_e \leq \eta\frac{1.0}{\alpha}$ then
10:                    Solve the TPBVP by cminpack;
11:                    Save the characteristic mass $\bar{M}_f$;
12:                end if
13:            end if
14:        end for
15:    end for
16: end for
```

$$a_{Dyson} = 1.1AU, e_{Dyson} = 0, i_{Dyson} = 0°, \Omega_{Dyson} = 0° \tag{18}$$

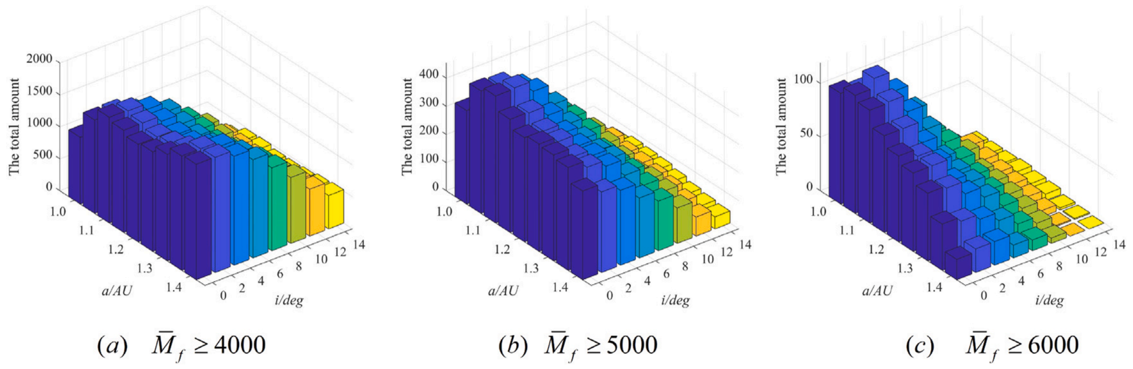
3.2. Mother Ship trajectories design

There are three main difficulties in designing the Mother Ship trajectories. (1) How to select the asteroids in the flyby sequences to meet the needs of small velocity increment and large characteristic mass? (2) How to deal with the huge computational burden caused by the large number of candidates? (3) How to avoid falling into a local optimum prematurely when extends the existing flyby sequence.

To overcome the difficulties mentioned above, 10 Mother Ship trajectories are generated by the following four steps. Firstly, according to the “Dyson Ring” orbital elements designed in the previous section and the characteristic mass of the asteroids under the corresponding elements, the top 10,000 asteroids based on characteristic mass are selected as subsequences database generation candidates. Secondly, based on the two-impulse Lambert problem, our team modified the beam search algorithm to adapt to parallel computing framework, which is employed to build the earth-to-asteroid and asteroid-to-asteroid (ast-to-ast) subsequences database. The generation of database is accomplished within the limited time of the competition owing to the utilization of a powerful parallel computation technique. Thirdly, the complete Mother Ships trajectories are generated by connecting the multiple subsequences. Lastly, for each Mother Ship trajectory, to further reduce the velocity increment, some additional optimizations are required, which include adjusting the transfer time between asteroids and optimizing the flyby velocity.

3.2.1. Database generation using parallel beam search

As described in Eq. (1), the velocity increment and characteristic mass are both design variables. Besides, the trajectory design is supposed to take the departure epoch from the earth and time of flight between asteroids into account. Faced with a large number of candidate asteroids and multi-dimensional design variables, our team employed beam search to design the complete flyby sequence at the



(a) $\bar{M}_f \geq 4000$ (b) $\bar{M}_f \geq 5000$ (c) $\bar{M}_f \geq 6000$

Fig. 2. The grid search results.

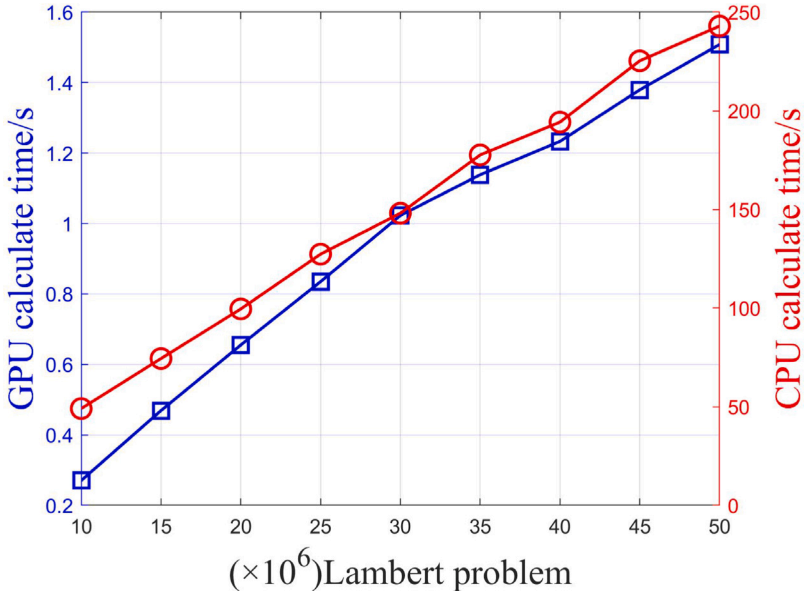


Fig. 3. GPU and CPU time for calculating the lambert problems.

beginning of competition. Beam search is an improved algorithm of greedy search, which provides a hyperparameter, beam width, to adjust the size of the search space. However, the large number of candidates made it easily fall into a local optimum even if a large beam width was applied. Consequently, instead of using beam search directly, the subsequences databases were built using parallel beam search. Owing to the powerful parallel processor, GPUs are now applied to accelerate aerospace scientific computation in a wide range of fields, such as rapid indirect trajectory optimization [26] and real-time propagating and visualizing the uncertainty of multiple orbit satellites [27].

The two-impulse transfer model is adopted to simplify the ast-to-ast trajectory. Therefore, the ast-to-ast transfer is solved directly based on the corresponding Lambert problem [28], which is the most time-consuming part of building the database. As shown in Fig. 3, the calculation results of multiple Lambert problems are compared with GPU and CPU respectively. It is worth noting that all numerical computations in this competition were run on a personal computer with an Intel Core i9-10900X 3.70 GHz CPU, 32 Gb of RAM, and NVIDIA GeForce RTX 2080Ti GPU. According to the simulation results, it takes the parallel computing platform about 1.5 s to calculate 50 million Lambert problems, while the CPU takes 250 s, and the former is 160 times faster than the latter at average. Based on the preliminary analysis of parallel computing advantages, our team proposed a parallel beam search algorithm, as shown in Algorithm 2. The purpose of this algorithm is to generate a series of subsequences starting from a given asteroid Id and corresponding departure epoch T_{dep}. The most bright idea of this algorithm is that millions of Lambert problems are conducted in parallel computing platform.

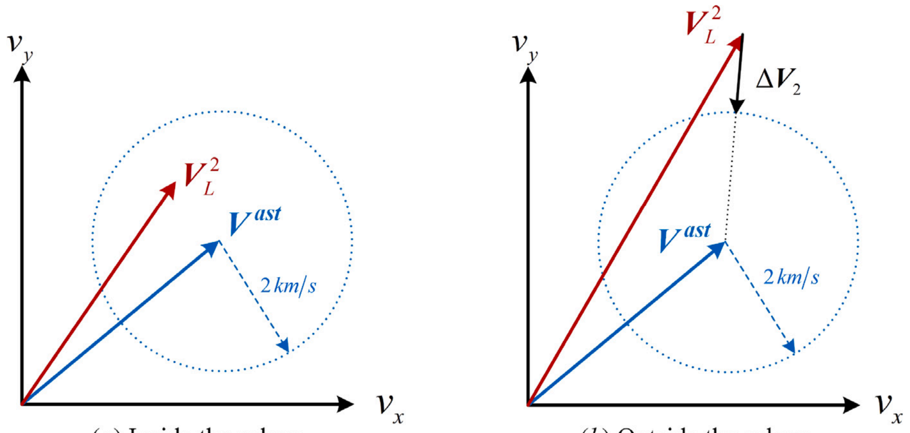
As seen in line 1 of Algorithm 2, the big characteristic mass set $\mathcal{B}$ is initialized by the results of grid search. Our team selected the top 10 000 characteristic mass asteroids as database candidates from the combination of $a_{Dyson} = 1.1AU$ and $i_{Dyson} = 0°$ instead of the over 83,000 asteroids, and thus, $|\mathcal{B}| = 10\,000$. The asteroids' velocity map $\mathcal{D} = \{Id : V_0\}$ and asteroids' departure epoch map $\mathcal{E} = \{Id : T_{dep}\}$ are initialized by the departure asteroid'Id and corresponding velocity V_0 and departure epoch T_{dep} respectively, and thus, $|\mathcal{D}| = |\mathcal{E}| = 1$. At each step, the total number of Lambert problems would be up to several million, while only one Lambert problem needs to be solved for each thread. The total number of threads which need to be assigned at each loop is:

$$T_{idx} = |\mathcal{B}| \times |\mathcal{D}| \times \left(Ft^{max} - Ft^{min}\right) / Ft^{s} \tag{19}$$

where the gridsize G_s and blocksize B_s are assigned according to the number of threads and the hardware configuration of the parallel computing platform.

$$G_s = \lceil \frac{T_{idx}}{B_s} \rceil \tag{20}$$

For each thread, a Lambert problem starting from asteroid a_d and ending at asteroid a_t with flight time Ft would be solved. It is noting that the initial and terminal position of Lambert problem are obtained



(a) Inside the sphere

(b) Outside the sphere

Fig. 4. Two situations of arrival velocity.

from asteroids' position at T_{dep} and $T_{dep}+Ft$ by Kepler's equation [28]. As a result, the initial velocity $\boldsymbol{V}_L^1$ and terminal velocity $\boldsymbol{V}_L^2$ are obtained by solving the Lambert problem in line 7. In line 8, the total velocity increment Δv consists of two parts: one is the departure velocity increment $\|\boldsymbol{V}_L^1 - \boldsymbol{V}_0\|_2$, the other one is the arrival velocity increment that is conducted to satisfy the relative speed condition: $\|\boldsymbol{V}^{mother} - \boldsymbol{V}^{ast}\|_2 \leq 2$ km/s. As shown in Fig. 4, the terminal velocity $\boldsymbol{V}_L^2$ would have two situations. If $\boldsymbol{V}_L^2$ is inside of the sphere whose center is $\boldsymbol{V}^{ast}$ with radius 2 km/s, the arrival velocity increment $\Delta\boldsymbol{V}_2 = \boldsymbol{0}$; otherwise, it should execute $\Delta\boldsymbol{V}_2 \neq \boldsymbol{0}$ to satisfy the flyby condition. The arrival velocity increment $\Delta\boldsymbol{V}_2$ is essentially a piecewise function, as follows:

$$\Delta\boldsymbol{V}_2 = \begin{cases} \boldsymbol{0} & \|\boldsymbol{V}^{ast} - \boldsymbol{V}_L^2\|_2 \leq 2 \\ \boldsymbol{V}^{ast} - \boldsymbol{V}_L^2 - 2\frac{\boldsymbol{V}^{ast}-\boldsymbol{V}_L^2}{\|\boldsymbol{V}^{ast}-\boldsymbol{V}_L^2\|_2} & \|\boldsymbol{V}^{ast} - \boldsymbol{V}_L^2\|_2 > 2 \end{cases} \tag{21}$$

When finishing calculation, the ast-to-ast pairs which satisfy the given velocity increment threshold Δv_t would be stored. At the end of each step, the map $\mathcal{D}$ and $\mathcal{E}$ are supposed to be updated for the next loop. It is noting that for the same arrival asteroid, there might have different arrival velocity and arrival epoch owing to the difference departure velocity and transfer time. However, only one certain value is allowed for each key in the map according to its definition. Therefore, in this case, the new serial number are marked behind the same Id, such as: $\mathcal{D} = \{Id_1^1 : \boldsymbol{V}_1^1, Id_2^1 : \boldsymbol{V}_2^1, \dots, Id_n^1 : \boldsymbol{V}_n^1, Id^2 : \boldsymbol{V}^2\}$, where the superscript denotes the different Id and the subscript denotes the same Id with different arrival velocity (departure velocity for the next loop).

The subsequences database generation algorithm shown in Algorithm 3 is proposed to generate the complete earth-to-ast and ast-to-ast subsequences based on Algorithm 2. It is noting that the input values of Algorithm 3 involve the input values of Algorithm 2 where the subscript e and a denote the input value of earth-to-ast and ast-to-ast subsequences generation respectively.

As seen in line 1∼4 of Algorithm 3, a series of subsequences from the earth to the first asteroid are built with the length of the subsequence $L_e = 2$. In line 2, for convenience, $Id = 0$ denotes the earth and the $\boldsymbol{V}_0$ denotes the earth velocity at T_{dep}. After generating all the possible earth-to-ast subsequences, the ast-to-ast subsequences are built similarly with the length of L_e. Firstly, at each T_{dep} epoch, the departure asteroids'Id and corresponding velocity are wrapped into a map $\mathcal{I}$ from the existing database. (i.e. extract the last asteroids'Id from each existing subsequence whose flyby epoch equals to T_{dep}). For each $(Id : \boldsymbol{V}_0) \in \mathcal{I}$, a series of subsequences would be generated by running the Algorithm 2. In line 10, the subsequence whose total velocity increment that is smaller than $\eta\,(L_a - 1)\,\Delta v_t^a$ would be saved in database, where $0 < \eta \leq 1$. This mainly involves two aspects: on one hand, the subsequences with smaller velocity increment would be saved; on the other hand, the number of $|\mathcal{D}|$ will increase as loop goes, and thus it can relieve the computational burden by downsizing the latest subsequences.

Algorithm 2 Parallel Beam Search Algorithm

```
Input: Departure epoch $T_{dep}$; Velocity increment threshold $\Delta v_t$; Departure asteroid's $Id$ and corresponding velocity $V_0$; Beam width $B_w$; The length of subsequence $L$; Maximum flight time $Ft^{max}$; Minimum flight time $Ft^{min}$; Flight time discretization step $Ft^s$;
Output: A series of subsequences starting from $Id$ at departure epoch $T_{dep}$;
1: Initialization: Big characteristic mass set $\mathcal{B}$; Asteroids' velocity map $\mathcal{D} = \{Id : V_0\}$; Asteroids' departure epoch map $\mathcal{E} = \{Id : T_{dep}\}$;
2: for $l = 1; l < L; l++$ do
3:     Calculate the total required threads: $T_{idx} = |\mathcal{B}| \times |\mathcal{D}| \times (Ft^{max} - Ft^{min})/Ft^s$;
4:     Assigning the gridsize $G_s$ and blocksize $B_s$ according $T_{idx}$;
5:     Assigning the departure asteroids'$Id$ $a_d \in \mathcal{D}$, the target flyby asteroids'$Id$ $a_t \in \mathcal{B}$, the flight time $Ft \in (Ft^{min} : Ft^s : Ft^{max}]$ to each thread;
6:     For each thread:
7:         $[V_L^1, V_L^2] = Lambert(a_d, a_t, \mathcal{E}, Ft)$;
8:         $\Delta v = \|V_L^1 - V_0\|_2 + \|\Delta V_2\|_2$;
9:         if $\Delta v \leq \Delta v_t$ then
10:            Store the tree nodes $(a_d, a_t, \mathcal{E}, Ft)$;
11:        end if
12:    end
13:    Save the top $B_w$ nodes;
14:    Updating map $\mathcal{D}$, $\mathcal{E}$ according to the latest subsequences;
15: end for
```

3.2.2. Subsequences connection

The complete Mother Ship trajectories are obtained by connecting the subsequences from the subsequences database. However, hundreds of millions of subsequences cause the curse of dimensionality during the connection. As shown in Algorithm 4, beam search is applied to connect the subsequences to design the Mother Ship trajectories.

In line 2, the earth-to-ast subsequences are selected to build the complete trajectories sequence set C at first. In line 4∼9, for each sequence $s \in C$, the last asteroid's Id and corresponding flyby epoch T_{now} are obtained, then the ast-to-ast subsequences starting from Id at T_{now} are obtained from the database. For each subsequence, a screening process is required to be conducted for reserving the subsequences

Algorithm 3 Subsequences Database Generation Algorithm

```
Input: Velocity increment threshold $\Delta v_t^e, \Delta v_t^a$; Beam width $B_w$; The length of subsequence $L_e, L_a$; Maximum flight time $Ft_e^{max}, Ft_a^{max}$; Minimum flight time $Ft_e^{min}, Ft_a^{min}$; Flight time discretization step $Ft^s$; Mother Ship's departure windows from the Earth $\left[T_e^0, T_e^f\right]$; Mother Ship's flyby windows $\left[T_a^0, T_a^f\right]$;
Output: The earth-to-ast and ast-to-ast subsequences database;
for $T_{dep} = T_e^0; T_{dep} \le T_e^f; T_{dep} = T_{dep} + Ft^s$ do
    Initializing the input values: $T_{dep}, \Delta v_t^e, Id = 0\,(Earth), \boldsymbol{V}_0, B_w, L_e = 2, Ft_e^{max}, Ft_e^{min}, Ft^s$;
    Running the Algorithm 2;
end for
for $T_{dep} = T_a^0; T_{dep} \le T_a^f; T_{dep} = T_{dep} + Ft^s$ do
    Forming map $\mathcal{I} = \{Id^1 : \boldsymbol{V}^1, Id^2 : \boldsymbol{V}^2, \ldots, Id^n : \boldsymbol{V}^n\}$ from existing database;
    for each $(Id : \boldsymbol{V}_0) \in \mathcal{I}$ do
        Initializing the input values: $T_{dep}, \Delta v_t^a, Id, \boldsymbol{V}_0, B_w, L_a, Ft_a^{max}, Ft_a^{min}, Ft^s$;
        Running the Algorithm 2;
        Saving the subsequence whose total velocity increment small than $\eta\left(L_a - 1\right)\Delta v_t^a$;
    end for
end for
```

whose asteroids are not in the set $\mathcal{P}$. Finally, a large number of up-to-date sequences are generated by connecting the subsequences with the sequence s. With the beam search algorithm, the top B_w sequences would update the set C for the next repeat process. In line 11, t_a is the flight time from the asteroid orbit to the "Dyson Ring" orbit and it is obtained from the result of grid search in Section 3.1. At the end of this algorithm, the top score trajectory in the set C would be selected as a Mother Ship trajectory.

Algorithm 4 Subsequences Connection Algorithm

```
Input: Departure epoch $T_e$ from the Earth; Subsequences database; Beam search width $B_w$; The set $\mathcal{P}$ that includes the asteroids' $Id$ which exist in the previous Mother Ship trajectories;
Output: A Mother Ship trajectory starting from $T_e$;
1:  $T_{now} \leftarrow T_e$;
2:  Saving the top $B_w$ Earth-to-ast subsequences starting from $T_{now}$ into complete trajectories sequence set $C$;
3:  repeat
4:      for each sequence $s \in C$ do
5:          Obtaining the last $Id$ and corresponding flyby epoch $T_{now}$ from the sequence $s$;
6:          Obtaining the ast-to-ast subsequences starting from $Id$ at $T_{now}$ from the database;
7:          Reserving the subsequences whose asteroids are not in the set $\mathcal{P}$;
8:          Connecting the subsequences with the sequence $s$;
9:      end for
10:     Updating the top $B_w$ sequences into $C$;
11: until for all asteroid $\in$ the last asteroid of the set $C$ satisfy $T_{now}+t_a > T_e + 365.25 \times 20$
12: Selecting the top score trajectory from the set $C$;
```

3.2.3. Optimizing the mother ship trajectory

In the subsequences generation algorithm, a two-impulse strategy is used to evaluate the velocity increment between asteroids. The velocity increment Δv for each flyby asteroid consists of two parts: departure and arrival velocity increment, and the arrival velocity increment depends on the relative velocity between the arrival velocity and asteroid velocity. This method can effectively reduce the computational burden for generating subsequences database, but the fuel consumption might increase. As shown in Table 1, after implementing the algorithm 4, a Mother Ship trajectory $\mathcal{T}$ with n asteroids are obtained. For each trajectory, the flyby asteroids' Id_i and corresponding flyby epoch T_i, velocity increment Δv_i, and flyby velocity $\boldsymbol{V}^i$ are recorded. In this subsection, the Mother Ship trajectories are optimized to reduce the total velocity increment by adjusting the transfer time and optimizing the flyby velocity.

Table 1
A Mother Ship trajectory with n asteroids.

	1	2	...	$n-1$	n
Asteroids' Id	Id_1	Id_2	...	Id_{n-1}	Id_n
Flyby epoch	T_1	T_2	...	T_{n-1}	T_n
Velocity increment	Δv_1	Δv_2	...	Δv_{n-1}	Δv_n
Flyby velocity	$\boldsymbol{V}^1$	$\boldsymbol{V}^2$	...	$\boldsymbol{V}^{n-1}$	$\boldsymbol{V}^n$

Firstly, the transfer time between flyby asteroids is adjusted to minimize the total velocity increment, as shown in algorithm 5. In this algorithm, a new Mother Ship trajectory $\mathcal{T}_{new}$ with updating flyby epoch, velocity increment, and flyby velocity are obtained by iterating the transfer time until $\Delta V_{old} - \Delta V < \varepsilon$. It is worth noting that T_{i-1} is equal to the earliest departure epoch from the earth when $i = 1$, and T_{i+1} is equal to the nth asteroid flyby epoch when $i = n$.

Algorithm 5 Adjusting Transfer Time Algorithm

```
Input: A Mother Ship trajectory $\mathcal{T}$;
Output: A new Mother Ship trajectory $\mathcal{T}_{new}$ with updating flyby epoch, velocity increment, and flyby velocity;
Obtaining the total velocity increment $\Delta V = \sum_{i=1}^{n} \Delta v_i$;
repeat
    $\Delta V_{old} \leftarrow \Delta V$;
    for $i = 1; i \le n; i++$ do
        Adjusting flyby epoch $T_i \in \left(T_{i-1}, T_{i+1}\right)$;
        Calculating the total velocity increment $\Delta V$;
        Recording the updating $T_i$ which minimizes $\Delta V$;
    end for
until $\Delta V_{old} - \Delta V < \varepsilon$
```

Secondly, the flyby velocity is optimized to further reduce the total velocity increment. As shown in Fig. 5, the flyby velocity is $\boldsymbol{V}^i$, and the asteroid velocity is $\boldsymbol{V}_a^i (i = 1, \ldots, n)$, where n is the total number of asteroids in the sequence. The optimization model is established with the flyby velocity $\boldsymbol{V}^i$ as design variables, the total velocity increment as objective function and the flyby velocity $\|\boldsymbol{V}^{mother} - \boldsymbol{V}^{ast}\|_2 \le 2$ km/s as constraint conditions. As a result, the optimization problem is formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{V}^i} \quad & \left\|\boldsymbol{V}^1 - \boldsymbol{V}_-^1\right\|_2 + \sum_{i=2}^{n}\left(\left\|\boldsymbol{V}^n - \boldsymbol{V}_-^n\right\|_2 + \left\|\boldsymbol{V}_+^{n-1} - \boldsymbol{V}^{n-1}\right\|_2\right) \\ s.t. \quad & \left\|\boldsymbol{V}^i - \boldsymbol{V}_a^i\right\|_2 \le 2, (i = 1, \ldots, n) \end{aligned} \tag{22}$$

Based on the algorithm 2∼4, a Mother Ship trajectory is generated with 33 flyby asteroids, which is shown in Table 2 and Fig. 6. Before implementing the velocity increment optimization, the total velocity increment of this Mother Ship trajectory is equal to 18.81 km/s. As shown in Table 2, the number of asteroids has no change between before and after optimization, while the transfer time between asteroids is optimized. After implementing the algorithm 5, the total velocity increment decreases to 16.38 km/s. Then, the flyby velocity of each asteroid is further optimized, and the final total velocity increment decreases to 16.05 km/s. The specific flyby epoch and velocity increment of before and after optimization are shown in Table 2.

3.3. Stations building strategy

After obtaining 10 Mother Ship trajectories and corresponding flyby asteroids, the next step is to build 12 stations. Five aspects are need to

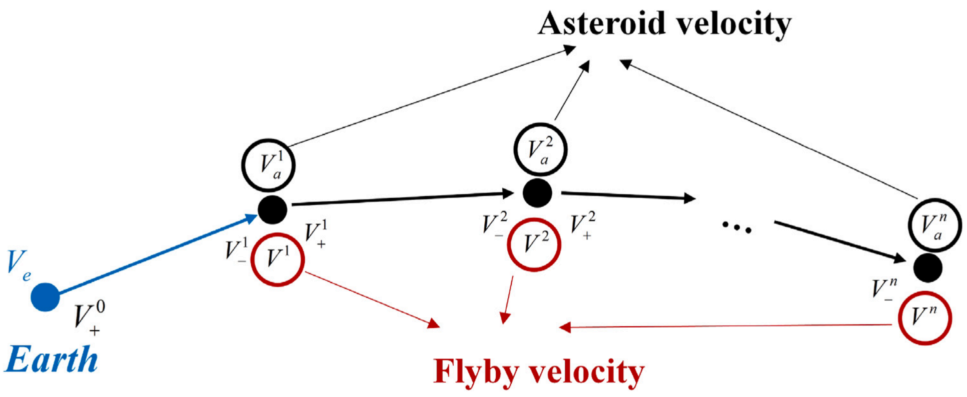


Fig. 5. Optimizing flyby velocity.

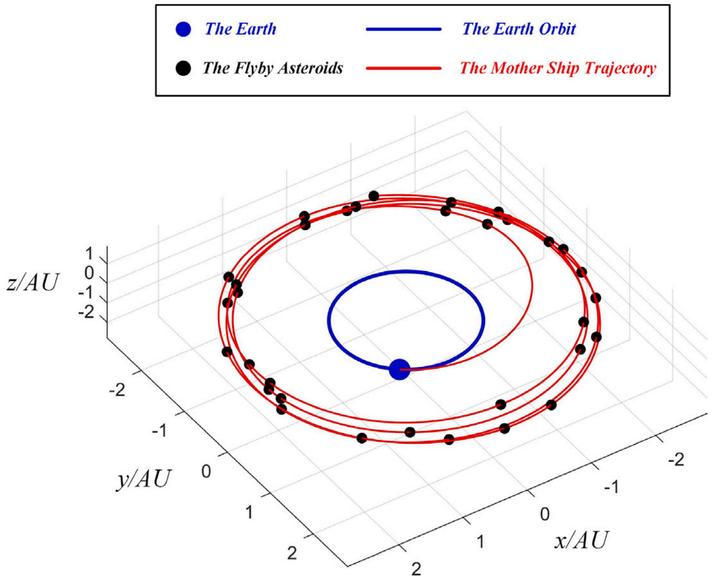


Fig. 6. A Mother Ship trajectory and corresponding flyby asteroids.

Table 2
The data of a Mother Ship trajectory before and after optimization.

Seq.	*Id*	Before		After		Seq.	*Id*	Before		After	
		Epoch	Δv	Epoch	Δv			Epoch	Δv	Epoch	Δv
0	0	96 030	–	96 045	–	17	75 573	98 850	0.46	98 865	0.34
1	54 925	96 320	2.95	96 345	2.51	18	77 264	99 110	0.42	99 126	0.14
2	79 402	96 390	1.22	96 403	1.14	19	71 489	99 270	0.49	99 276	0.52
3	69 339	96 500	0.64	96 521	0.77	20	83 102	99 400	0.63	99 416	0.18
4	81 589	96 750	0.38	96 766	0.32	21	82 274	99 640	0.35	99 644	0.58
5	64 889	96 880	0.54	96 883	0.31	22	82 933	99 820	0.32	99 820	0.35
6	79 502	96 920	0.31	96 923	0.39	23	78 002	99 880	0.67	99 892	0.18
7	81 938	97 180	0.60	97 184	0.35	24	81 275	100 090	0.29	100 109	0.27
8	46 343	97 340	0.39	97 354	0.69	25	80 747	100 260	0.62	100 160	0.20
9	55 928	97 500	0.59	97 514	0.45	26	80 554	100 520	0.26	100 534	0.49
10	83 132	97 660	0.41	97 659	0.32	27	68 234	100 650	0.68	100 665	0.54
11	30 517	97 770	0.57	97 775	0.49	28	81 013	100 910	0.64	100 904	0.50
12	57 818	97 980	0.55	97 986	0.31	29	83 288	101 130	0.33	101 119	0.38
13	78 188	98 160	0.40	98 162	0.58	30	83 340	101 390	0.55	101 391	0.25
14	82 926	98 180	0.37	98 182	0.43	31	69 120	101 560	0.07	101 562	0.35
15	65 315	98 340	0.62	98 342	0.35	32	83 379	101 670	0.35	101 664	0.19
16	77 108	98 600	0.53	98 599	0.71	33	72 942	101 830	0.60	101 815	0.48

be considered: φ_1, the station building sequence, the station building windows, as well as the ATD activation epoch and the target station for each asteroid. To simplify the optimization problem, the attainable stations for each asteroid with given building windows, sequence, and φ_1 are analyzed at first. Based on the attainable station results for each asteroid, the coordinate descent method is utilized to optimize the target allocation problem and obtain the ATD activating epoch with the corresponding target station for each asteroid.

Furthermore, the complete station building optimization model is established in such a form that the building windows and the phase

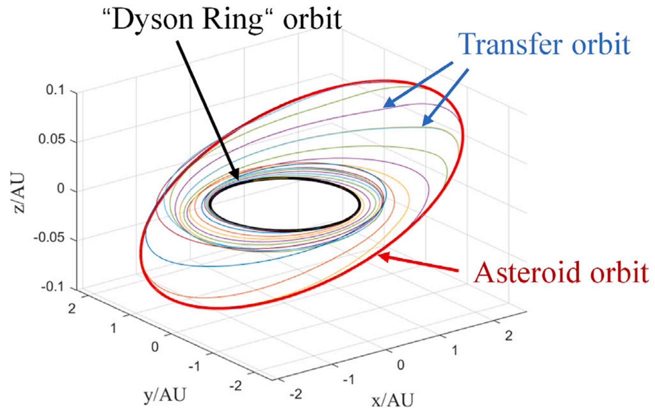


Fig. 7. Time-optimal database generation.

of the first building station φ_1 are the optimization design variables, the coordinate descent for the target assignment is the inner loop optimization problem, and the maximizing the minimum mass among all building stations is the objective function. Finally, the departure epoch of each flyby asteroid and the corresponding target station are obtained.

3.3.1. Attainable station calculation

The problem of determining the attainable station for the flyby asteroids is essentially a time-optimal rendezvous problem that includes the building window constraints. Based on the TPBVP formulated in Eq. (14), the unconstrained free terminal phase time-optimal problem is utilized to generate a database for each flyby asteroid that includes the ATD activation epoch, initial costate variables, transfer time, and corresponding arrival phase.

As shown in Algorithm 6, the time-optimal database generation algorithm is proposed to map the departure epoch with the terminal phase in the “Dyson Ring” orbit. For each asteroid, it is inevitable that the inaccurate guess of the initial costate values and transfer time affect the convergence and the speed of the computation. In line 4, at the earliest possible activation epoch, the TPBVP is solved with random initial variables $\boldsymbol{z}_0$. It is worth noting that, in line 6∼11, in the first orbital period T_p^a, the initial shooting variables can be obtained from the result of the previous discretization step, and the simulation results are depicted in Fig. 7. Additionally, for each asteroid, 10 orbital period datas are built in line 12∼14 directly by propagating the shooting solution and arrival phase from the first period.

The time-optimal database mentioned above provides an offline way to generate the attainable stations set for each flyby asteroid. Algorithm 7 offers an interface, convenient for the outer loop optimization. In line 2, the time-optimal database of asteroid a are extracted from the offline database calculated by Algorithm 6. For each station, given the phase of the first station: φ_1 and their building windows $\left[Tb_i^0, Tb_i^f\right]$, the phases of station i are easily obtained by Kepler’s equation. The asteroid arrival phases and the station i phases are both depicted in Fig. 8, and both of them are changing periodically. As shown in Fig. 8, if the arrival phase of asteroid a equals to the phase of station i, the station i, the minimum transfer time and corresponding departure epoch would be recorded.

As a result, the assignment table is listed in Table 3 based on the attainable station results. As shown in Table 3, M_i^j denotes the characteristic mass of the ith asteroid assigned with the jth station, while 0 represents the ith asteroid could not reach the jth station. It is obvious that the number of attainable stations may be more than 1 (e.g. A_1), some may be equal to 1 (e.g. A_2), and some may be equal to 0 (e.g. A_3). In the next section, coordinate descent method would be applied to optimize the assignment problem, and the target station for each flyby asteroid would be obtained.

Algorithm 6 Time-optimal Database Generation Algorithm

```
Input: The set of the flyby asteroids’ $Id$: $\mathcal{F}$ and the corresponding flyby epoch; Flight time discretization step $Ft^s$;
Output: The time-optimal database for each flyby asteroid;
1:  for each asteroid $a \in \mathcal{F}$ do
2:      Obtaining the flyby epoch $T_{fb}^a$ of asteroid $a$;
3:      Initializing the shooting variables $\boldsymbol{z}_0$ randomly;
4:      Solving the TPBVP of Eq. (14) with state constraints at $T_{fb}^a + 30$ MJD;
5:      $T_i = T_{fb}^a + 30 + Ft^s$;
6:      while $T_i \le T_{fb}^a + 30 + T_p^a$ do
7:          The initial shooting variables $\boldsymbol{z}_0$ are inherited from the previous step solution;
8:          Solving the TPBVP of Eq. (14) with state constrains at $T_i$ MJD;
9:          Saving the solution into database;
10:         $T_i = T_i + Ft^s$;
11:     end while
12:     for $p = 2; p \le 10; p++$ do
13:         Propagating the shooting solution and arrival phase from the first period;
14:     end for
15: end for
```

Algorithm 7 Attainable Station Calculation Algorithm

```
Input: The phase of the first station: $\varphi_1$; Station building windows: $\left[Tb_i^0, Tb_i^f\right]$, $i = 1, 2, \ldots, 12$; The set of the flyby asteroids’ $Id$: $\mathcal{F}$; The time-optimal database;
Output: The asteroids attainable station and corresponding ATD activation epoch;
1: for each asteroid $a \in \mathcal{F}$ do
2:     Obtaining the time-optimal database of asteroid $a$;
3:     for $i = 1; i \le 12; i++$ do
4:         Calculating the phases of the station $i$ in $\left[Tb_i^0, Tb_i^f\right]$;
5:         if The arrival phase of asteroid $a$ equals to the phase of station $i$ then
6:             Saving station $i$, minimum transfer time and corresponding departure epoch;
7:         end if
8:     end for
9: end for
```

Table 3
Initial assignment table $Q_{n\times12}$.

Asteroids’Id	Stations’Id					
	S_1	S_2	S_3	…	S_{11}	S_{12}
A_1	M_1^1	0	M_1^3	…	0	M_1^{12}
A_2	0	M_2^2	0	…	0	0
A_3	0	0	0	…	0	0
A_4	M_4^1	0	M_4^3	…	M_4^{11}	0
…	…	…	…	…	…	…
A_{n-1}	0	M_{n-1}^2	0	…	0	M_{n-1}^{12}
A_n	M_n^1	M_n^2	0	…	M_n^{11}	0

3.3.2. Coordinate descent method

As mentioned above, the building windows and the phase of the first building station are both design variables, which were solved by PSO algorithm, as shown in Fig. 9. Because of the maturity of the PSO algorithm, the specific algorithm implement of PSO will not repeat here. It is noting that the station building sequence is assumed to be fixed for simplification. For each particle, a station assignment problem would be solved to obtain fitness. With the rapid convergency and finite

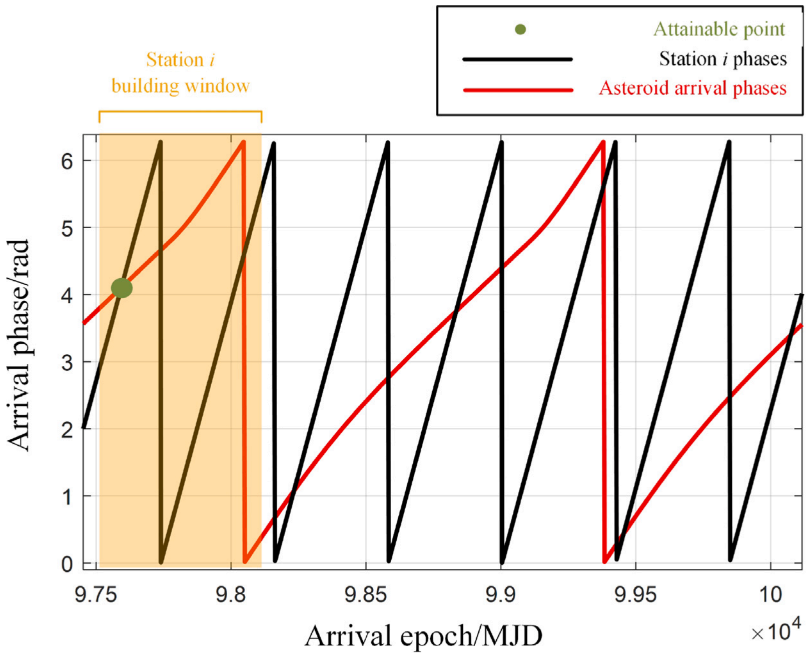


Fig. 8. Attainable station calculation.

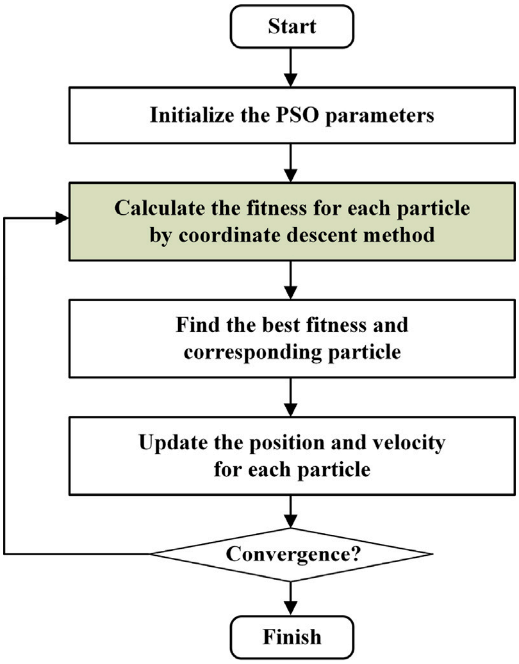

Start
Initialize the PSO parameters
Calculate the fitness for each particle by coordinate descent method
Find the best fitness and corresponding particle
Update the position and velocity for each particle
Convergence?
Finish

Fig. 9. Station assignment flow-process diagram.

iteration property, coordinate descent method is utilized to solve the station assignment problem.

The coordinate descent method for assigning station is shown in algorithm 8, which would yield an assignment vector $V_{n\times 1}$ based on the assignment table $Q_{n\times 12}$. The algorithm is initialized in line 1∼4, where the maximum characteristic mass corresponding station would be saved in the assignment vector for each flyby asteroid. Line 7∼16 represents the iteration process, and the assignment vector would be updated in every iteration until the difference of minimum station mass between two iteration small than a small positive constant.

Algorithm 8 Coordinate Descent Method for Assigning Station

```
Input: Assignment table $Q_{n\times 12}$;
Output: Assignment vector $V_{n\times 1}$;
 1: for $i = 1; i \le n; i++$ do
 2:     Obtaining the maximum characteristic mass of the $i^{th}$ row;
 3:     Saving the corresponding station: $V[i] = j^*$;
 4: end for
 5: $M_{\min}[1] = \min(S_1, S_2, ..., S_{12})$;
 6: $k \leftarrow 2$;
 7: repeat
 8:     for $i = 1; i \le n; i++$ do
 9:         for $j = 1; j \le 12; j++$ do
10:             $j^* = Max\left[\min(S_1, S_2, ..., S_{12})\right]$;
11:         end for
12:         $V[i] = j^*$;
13:     end for
14:     Updating $M_{\min}[k]$;
15:     $k = k + 1$;
16: until $\left(M_{\min}[k] - M_{\min}[k-1]\right) < \varepsilon$
```

3.4. Asteroid rendezvous trajectory design

After completing the station assignment mission, the departure epoch and the target station of each asteroid would be obtained. However, the terminal condition propagating from the solution to the free phase time-optimal problem obtained from algorithm 7 could not meet the accuracy of the problem. The asteroid rendezvous trajectory problem is essentially a time-varying terminal phase low thrust trajectory optimization problem. Similar to Section 3.1.1, the asteroid rendezvous trajectory problem is formulated as follows:

The dynamical model, performance index, initial boundary constraints, Hamiltonian, optimal thrust direction, and costate equations are the same as Eqs. (4), (5), (6), (8), (9), and (10), respectively. Instead of the free terminal phase condition, the time-varying phase of the jth

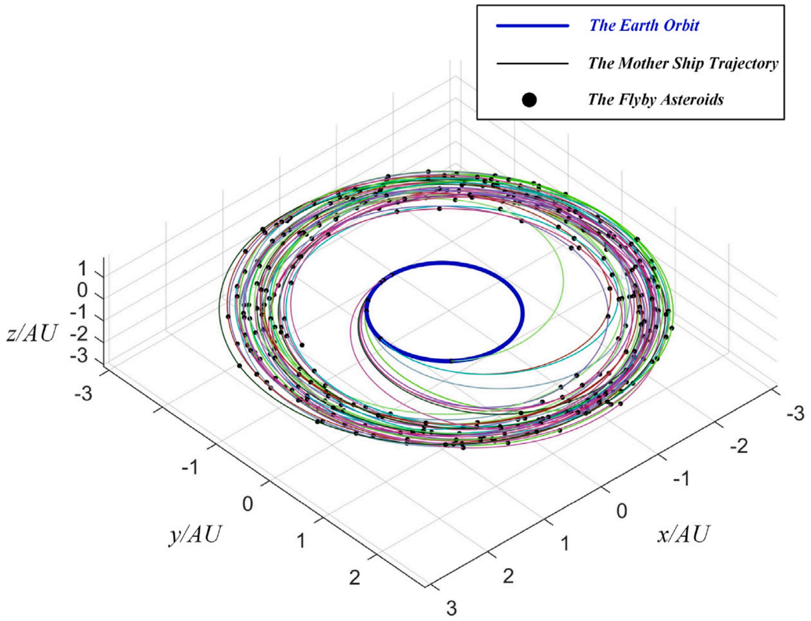


Fig. 10. The Mother Ship trajectories and flyby asteroids.

station is obtained by:

$$L_j(t) = L_j(0) + n_{Dyson} t \tag{23}$$

where $n_{Dyson} = \sqrt{\mu / a_{Dyson}^3}$.

Consequently, the shooting function is built as:

$$f'(z) = \left[p\left(t_f\right) - a_{Dyson}, f\left(t_f\right), g\left(t_f\right), h\left(t_f\right) - \tan\frac{i_{Dyson}}{2}, k\left(t_f\right), L\left(t_f\right) - \left(L(0) + n_{Dyson} t_f\right), H\left(t_f\right) \right]^T = \mathbf{0} \tag{24}$$

The problem is much more difficult than the free phase time-optimal problem, but fortunately, Algorithm 7 provides an accurate initial costate values and transfer time for each flyby asteroid.

4. Competition results

4.1. Submitted results

At the end of the competition, our team is ranked at the tenth place among 94 teams. The final obtained performance index $J = 3532.7$ with the number of flyby asteroids $N = 199$ and the minimum station $M_{min} = 8.0 \times 10^{14}$ kg. During the competition, the velocity increment was a major obstacle that hinders the improvement of the grade. Based on the updating method after competition, we focus on the post-competition results described in the next section.

4.2. Post-competition results

During the competition, the “Dyson Ring” orbital elements were not optimized in the submitted version due to the lack of analyzing the relationship between the radius and the characteristic mass for each candidate. Additionally, excessive focus on the asteroids’ mass when choosing the flyby asteroids by our team causes the large velocity increment, more than 26 km/s for each Mother Ship. Besides, the large transfer time between flyby asteroids results in the small number of flyby asteroids. After the competition, our team focused on analyzing how to increase the number of flyby asteroids and reduce the velocity increment. The algorithms for solving the question by our team are all illustrated in Section 3.

The Mother Ship trajectories determine the upper bound of the performance index. Both the velocity increment and the station mass depend on the Mother Ship trajectories. Algorithms 2∼4 provide a parallel beam search algorithm, database generation strategy, and subsequence connection strategy to design the complete trajectory. Firstly, the parameters of the algorithm need to be initialized. In algorithm 3, the departure epoch windows from the earth$\left[T_e^0, T_e^f\right] = [95\,900, 96\,100]$ MJD, where the earth-to-ast database are established. Let the length of earth-to-astL_e and ast-to-ast L_a equal to 2 and 3 respectively. The transfer time between asteroids is $\left[Ft_a^{min}, Ft_a^{max}\right] = [10, 280]$ days, with discretization step $Ft^s = 10$ days.

As shown in Fig. 10 and Table 4, 10 Mother Ship trajectories are generated by the algorithms 2∼4. The total number of flyby asteroids $N = 332$. Based on the algorithm 5 and the flyby velocity optimization model Eq. (22), the velocity increment of each Mother Ship trajectory is optimized. As shown in Fig. 11, the velocity increment is optimized by implementing the adjusting transfer time algorithm at first, and the flyby velocity is further optimized. As a result, the Δv could be saved 2.5 km/s on average for each trajectory, and the specific values of the after optimization are listed in Table 4. Assuming that all flyby asteroids are evenly assigned to an appropriate target station, the characteristic mass of the minimal mass station equals to 1.078×10^5. Accordingly, the estimated performance index $J = 6071$.

After obtaining the flyby sequences, the PSO algorithm and the coordinate descent method are applied to assign the flyby asteroids to their target station. The number of assigned asteroids and station mass for each station are shown in Table 5. The minimal mass among all building stations is 1.19×10^{15} kg, while the number of assigned asteroids is 294. It is worth noting that the asteroids which are not assigned are owing to the inappropriate building windows. Especially, there are few attainable stations for the asteroids which are close to the end of the sequences, and these asteroids would be removed from the sequence to reduce the velocity increment. As shown in Table 6, the Mother Ship trajectories are updated by removing the asteroids which are not assigned and at the end of the sequences, while the asteroids which are in the middle of the trajectories are reserved. As a result, the total number of flyby asteroids $N = 301$. The final performance index J:

$$J{=}1.0 \times \frac{10^{-10} \times 1.19 \times 10^{15}}{a_{Dyson}^2 \sum_{k=1}^{10} \left(1 + \Delta V_k^{Total}/50\right)^2} = 5750.3 \tag{25}$$

Table 4
The characteristic mass and Δv for each Mother Ship trajectory.

Mother Ship	1	2	3	4	5	6	7	8	9	10
Asteroid quantity n	33	33	33	35	33	33	33	33	33	33
$\sum_{i=1}^{n} \bar{M}_f$ $(\times 10^5)$	1.236	1.318	1.224	1.361	1.284	1.291	1.313	1.279	1.352	1.277
$\sum_{i=1}^{n} \Delta v_i$ (km/s)	17.43	18.33	16.62	17.28	15.90	16.90	15.44	15.52	16.05	16.73

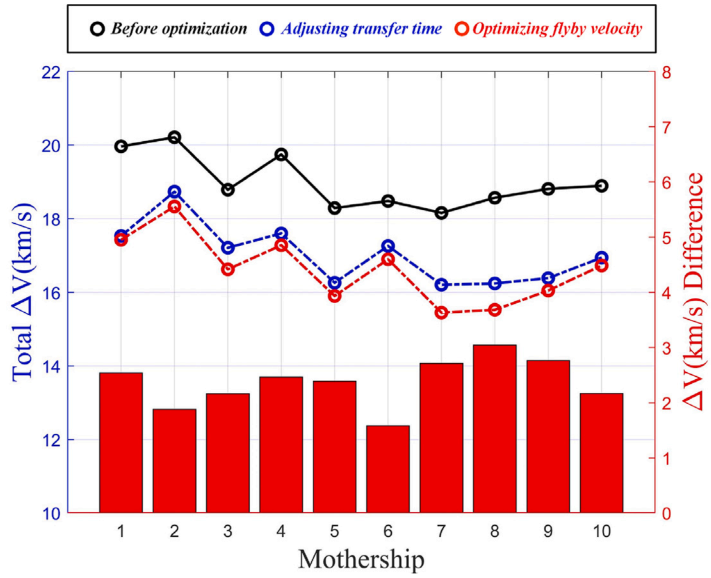


Fig. 11. The Mother Ship trajectories velocity increment optimization results.

Table 5
The number of assigned asteroids and station mass for each station.

Station	1	2	3	4	5	6	7	8	9	10	11	12
Asteroid quantity n	26	24	24	28	23	28	25	23	25	25	22	21
Mass $(\times 10^{15}$ kg)	1.21	1.19	1.21	1.21	1.20	1.20	1.20	1.21	1.19	1.20	1.20	1.20

Table 6
The number of asteroids and corresponding velocity increment for each Mother Ship after station assignment.

Mother Ship	1	2	3	4	5	6	7	8	9	10
Asteroid quantity n	32	32	29	32	32	25	30	32	29	28
$\sum_{i=1}^{n} \Delta v_i$ (km/s)	16.97	17.65	14.98	16.27	15.85	13.59	15.06	15.44	14.53	14.52

5. Conclusions

GTOC-11 provides an intricate and complex problem, aiming to build 12 stations in the “Dyson Ring” orbit. To solve this problem, plenty of expertise and optimization methods were utilized. The four-step method described above mainly decouples the interlinkages of the design variables. Firstly, the “Dyson Ring” elements are designed; Secondly, 10 Mother Ship trajectories are generated and optimized; Thirdly, the target station assignment problem is solved by the coordinate descent method in the PSO framework; Finally, the rendezvous trajectories are determined by solving a series of TPBVPs. The proposed method is valid for this problem, and the latest performance index $J = 5750.3$.

However, the methods proposed by our team put more emphasis on the subproblems optimization. In the future study, by analyzing the relationship between these subproblems and establishing a global optimization method, a higher score would be obtained.

Declaration of competing 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.

Acknowledgments

This study was funded by the National Natural Science Foundation of China under the grant 11972077.

References

[1] H. Shen, Y. Luo, Y. Zhu, A. Huang, Dyson sphere building: On the design of the GTOC11 problem and summary of the results, Acta Astronaut. (2022).

[2] D. Izzo, 1St ACT global trajectory optimisation competition: Problem description and summary of the results, Acta Astronaut. 61 (9) (2007) 731–734, http://dx.doi.org/10.1016/j.actaastro.2007.03.003.

[3] I.S. Grigoriev, M.P. Zapletin, GTOC5: PRoblem statement and notes on solution verification, Acta Futura 8 (2014) 9–19, http://dx.doi.org/10.2420/AF08.2014.9.

[4] L. Casalino, G. Colasurdo, Problem Description for the 7th Global Trajectory Optimisation Competition, GTOC Portal, 2014, http://sophia.estec.esa.int/gtocportal.

[5] D. Izzo, M. Märtens, The Kessler run: on the design of the GTOC9 challenge, Acta Futura 11 (2018) 11–24, http://dx.doi.org/10.5281/zenodo.1139022.

[6] S. Li, X. Huang, B. Yang, Review of optimization methodologies in global and China trajectory optimization competitions, Prog. Aerosp. Sci. 102 (2018) 60–75, http://dx.doi.org/10.1016/j.paerosci.2018.07.004.

[7] S. Zhao, J. Zhang, K. Xiang, R. Qi, Target sequence optimization for multiple debris rendezvous using low thrust based on characteristics of SSO, Astrodynamics 1 (1) (2017) 85–99, http://dx.doi.org/10.1007/s42064-017-0007-4.

[8] A.E. Petropoulos, E.P. Bonfiglio, D.J. Grebow, T. Lam, J.S. Parker, J. Arrieta, D.F. Landau, R.L. Anderson, E.D. Gustafson, G.J. Whiffen, et al., GTOC5: Results from the jet propulsion laboratory, Acta Futura 8 (1) (2014) 21–27, http://dx.doi.org/10.2420/AF08.2014.21.

[9] S. Zhao, R. Qi, J. Zhang, K. Xiang, C. Zhang, J. Jin, Problem a of 9th China trajectory optimization competition: Problem description and summary of the results, Acta Astronaut. 150 (2018) 178–181, http://dx.doi.org/10.1016/j.actaastro.2018.04.025.

[10] X. Huang, B. Yang, P. Sun, S. Li, H. Yang, 10 Th China trajectory optimization competition: Problem description and summary of the results, Astrodynamics 5 (1) (2021) 11, http://dx.doi.org/10.1007/s42064-020-0089-2.

[11] B. Yang, X. Huang, H. Yang, S. Li, X. Liu, P. Sun, W. Li, Problem a of 9th China trajectory optimization competition: Results found at NUAA, Acta Astronaut. 150 (2018) 182–192, http://dx.doi.org/10.1016/j.actaastro.2018.03.044.

[12] H. Li, H. Baoyin, Optimization of multiple debris removal missions using an evolving elitist club algorithm, IEEE Trans. Aerosp. Electron. Syst. 56 (1) (2019) 773–784, http://dx.doi.org/10.1109/TAES.2019.2934373.

[13] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, 1998.

[14] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948, http://dx.doi.org/10.1109/ICNN.1995.488968.

[15] C. Jing, S. Hongxin, L. Hengnian, Global optimization strategy of multi-satellite formation reconfiguration, Mech. Eng. 38 (6) (2016) 697–704, http://dx.doi.org/10.6052/1000-0879-16-176.

[16] Y.-Z. Luo, Y. Zhu, H. Zhu, Z. Yang, Z.-J. Sun, J. Zhang, GTOC9: Results from the national university of defense technology (team NUDT), Acta Futura 11 (2018) 37–47.

[17] A. Colorni, M. Dorigo, V. Maniezzo, et al., Distributed optimization by ant colonies, in: Proceedings of the First European Conference on Artificial Life, Vol. 142, Paris, France, 1991, pp. 134–142.

[18] A. Petropoulos, D. Grebow, D. Jones, G. Lantoine, J. Sims, GTOC9: MEthods and results from the jet propulsion laboratory team (team JPL), Acta Futura 11 (2018) 25–35.

[19] E. Fehlberg, Classical Fifth-, Sixth-, Seventh-, and Eighth-Order Runge-Kutta Formulas with Stepsize Control, National Aeronautics and Space Administration, 1968.

[20] J. Moré, B.S. Garbow, K.E. Hillstrom, L.A. National, User guide for minpack-1, J. Coll.. Gen. Pract. (1980).

[21] M. Walker, A set of modified equinoctial orbit elements, Celestial Mech. 38 (4) (1986) 391–392, http://dx.doi.org/10.1007/bf01238929.

[22] J.M. Longuski, J.J. Guzmán, J.E. Prussing, Optimal Control with Aerospace Applications, Springer, 2014.

[23] F. Jiang, H. Baoyin, J. Li, Practical techniques for low-thrust trajectory optimization with homotopic approach, J. Guid. Control Dyn. 35 (1) (2012) 245–258, http://dx.doi.org/10.2514/1.52476.

[24] T.N. Edelbaum, Propulsion requirements for controllable satellites, Ars J. 31 (8) (1961) 1079–1089, http://dx.doi.org/10.2514/8.5723.

[25] L. Casalino, Approximate optimization of low-thrust transfers between low-eccentricity close orbits, J. Guid. Control Dyn. 37 (3) (2014) 1003–1008, http://dx.doi.org/10.2514/1.62046.

[26] T. Antony, M.J. Grant, Rapid indirect trajectory optimization on highly parallel computing architectures, J. Spacecr. Rockets 54 (5) (2017) 1081–1091, http://dx.doi.org/10.2514/1.A33755.

[27] K. Liu, B. Jia, G. Chen, K. Pham, E. Blasch, A real-time orbit satellites uncertainty propagation and visualization system using graphics computing unit and multi-threading processing, in: 2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), 2015, pp. 8A2–1–8A2–10, http://dx.doi.org/10.1109/DASC.2015.7311467.

[28] H. Curtis, Orbital Mechanics for Engineering Students, Butterworth-Heinemann, 2013.