SCIENCE CHINA
Technological Sciences

• Article •

January 2021 Vol. 64 No. 1: 122–130
https://doi.org/10.1007/s11431-020-1560-9

Sequence optimization for multiple asteroids rendezvous via cluster analysis and probability-based beam search

LI HaiYang* & BAOYIN HeXi*

School of Aerospace Engineering, Tsinghua University, Beijing 100084, China

Received February 10, 2020; accepted March 10, 2020; published online October 19, 2020

It is of great significance to visit multiple asteroids in a space mission. In this paper, the multiple asteroids mission optimization is implemented using cluster analysis and probability-based beam search. Clustering is performed to select the first asteroid to visit. Four cluster algorithms are investigated and affinity propagation is selected. Then four beam search algorithms that are deterministic beam search and three probability-based beam search variants, probabilistic beam search, ant-colony beam search, and evolving beam search, are applied to search for the rendezvous sequence. Deterministic beam search as a heuristic tree search algorithm is widely applied in multitarget sequence optimization, but it has an obvious drawback of the conflict between the number of pruned nodes and the possibility of finding optimal solutions, which can be improved by probability-based beam search. Among three probability-based beam search, the ant-colony beam search has a learning mechanism, and evolving beam search is constructed based on ant-colony beam search and has an evolutionary mechanism. Results show that the introduction of randomness can improve beam search, and beam search variants with the learning and evolutionary mechanism have an excellent performance.

interplanetary trajectory optimization, multi-target mission, cluster analysis, probability-based beam search

Citation: Li H Y, Baoyin H X. Sequence optimization for multiple asteroids rendezvous via cluster analysis and probability-based beam search. Sci China Tech Sci, 2021, 64: 122–130, https://doi.org/10.1007/s11431-020-1560-9

1 Introduction

Asteroids have attracted vast research interest because of their great significance in various aspects [1–3]. In space missions to multiple targets, substantial benefits are gained since the average expense of exploring each target is lowered [4–6]. NASA's Near Earth Asteroid Rendezvous (NEAR) Shoemaker performed a flyby of the asteroid 253 Mathilde on the way to the asteroid 433 Eros [7]. In the previous plan for the NEAR mission, an ambitious plan called Small-Body Grand Tour, which aimed to achieve flybys of two comets and two asteroids over a 10-year period, was proposed [8]. Deep Space 1 [9] and Dawn [10] also achieved multitarget visits of small celestial bodies. The Global Trajectory Optimization Competitions (GTOCs)[1)], which are some of the most challenging events in space engineering, had several editions greatly focused on multitarget small celestial body missions.

The crucial part of optimizing a multiple asteroids rendezvous mission is finding an ideal sequence of asteroids to gain maximum benefit. However, there are normally a large number of asteroids that can be used as candidates. Sequence selecting is a complex global optimization problem in astrodynamics [11, 12], and the exhaustive search is practical only when the solution space is quite small [13]. Nevertheless, such combinatorial optimization problems are usually nondeterministic polynomial-time difficult problems, and their solution space cannot be thoroughly searched by enumera-

*Corresponding authors (email: lihy15@mails.tsinghua.edu.cn; baoyin@tsinghua.edu.cn)
1) https://sophia.estec.esa.int/gtoc_portal/

tion. Tree search algorithms have been employed in previous studies, where the tree node represents the trajectory between two asteroids. Some strategies have been used to prune nodes that have less potential to grow to a satisfying result, such as the Series Method [14] and search space pruning [15]. Beam search, which is a heuristic deterministic tree search algorithm, is widely applied to search for an optimal sequence and found the best solutions in the 8th China Trajectory Optimization Competition [5] and some GTOCs [16]. Beam search is efficient because they only explore promising nodes according to the heuristic cost so that the computation budget can be reduced, and a good pruning strategy can help find a satisfying solution. However, there is a conflict between the number of pruned nodes and the possibility of finding optimal solutions. Over pruned nodes may lose that possibility, but over reserved nodes may increase the computation time and make the algorithm become an exhaustive search. The balance of this conflict is hard to find when the problems are complex and large-scale, which may lead to an unsatisfying solution. Furthermore, the main drawback of these tree search algorithms is that they focus only on the next level when extending their nodes, which may result in poor nodes in future levels.

Probabilistic beam search [17, 18] is a method that can alleviate this conflict. In probabilistic beam search nodes are selected based on probability. The better the heuristic cost is, the greater the probability of being selected is. The nodes that get pruned in deterministic beam search now have the opportunity to be reserved in probabilistic beam search. The introduction of randomness makes beam search a more powerful tool [19]. However, in probabilistic beam search every complete search process is independent and has no effect on each other. The algorithm will be more powerful if the results of former probabilistic beam search can guide later probabilistic beam search, which means the algorithm can learn from former search. A learning mechanism can be introduced to the probabilistic beam search.

Hybridizing ant colony optimization (ACO) with beam search is a solution to introduce the learning mechanism [20]. ACO is widely used in astrodynamics such as multiple gravity assist trajectory planning [21] and multi-asteroid rendezvous tours [22]. ACO is an algorithm based on the foraging behavior of ants, and the behavior of ants searching for the optimal path is quite similar to the design of space missions searching for the optimal sequence. In ACO, the ant will be guided by both local cost information and global pheromone information when searching for the next target. The global pheromone information is updated based on results of the previous search, so the ant can "learn" from the previous search. The difference between ACO and ant-colony beam search is that in ant-colony beam search the standard ACO solution construction mechanism is replaced by a construction mechanism in which each ant performs a probabilistic beam search. Ant-colony beam search has been applied to multiple asteroids mission design by ref. [23] and proven to be effective.

Besides the learning mechanism, the evolutionary mechanism can also be introduced to further improve the algorithm. Evolutionary algorithms (EA) are another major class of stochastic intelligent optimization algorithms, in addition to the swarm intelligence to which ACO belongs. After a solution phenotype is well-coded into a genome, the operators in EA such as crossover and mutation show excellent performance and convincingly improve current solutions. The combination of ACO and EA through an evolving population is proposed by ref. [24]. In this paper, a novel probability-based beam search named evolving beam search is proposed. The evolving beam search is constructed based on ant-colony beam search. The solution construction mechanism of evolving beam search is the same as that of ant-colony beam search. The difference between evolving beam search and ant-colony beam search is that in evolving beam search there is an evolving population that brings in an evolutionary mechanism. This evolving population consists of ants with the best solution at each iteration and is then involved in evolution at the end of each iteration through EA operators. Then the best ant in the evolving population is used to update the pheromone.

In this paper, the first asteroid to visit is selected using cluster analysis. Facing a large set of asteroids, it is a natural idea to try to divide asteroids that are close to each other into the same group. Cluster analysis can automatically group asteroids based on their orbits. Clustering is a common technique for data mining and has its applications in orbit clustering [16, 25]. Starting from an asteroid in a cluster with more asteroids is more likely to find a better sequence [16], therefore the asteroid in the biggest cluster, named "central asteroid", is selected as the first asteroid to visit. Four cluster algorithms are compared and affinity propagation [26] is selected. Then starting from this asteroid, four beam search algorithms, deterministic beam search, probabilistic beam search, ant-colony beam search, and evolving beam search are applied to search for the rendezvous sequences.

This paper is organized as follows. In Sect. 2, the method of cluster analysis is detailed. In Sect. 3, the sequence optimization problem is built up, and four beam search, deterministic beam search, probabilistic beam search, ant-colony beam search, and evolving beam search are introduced. In Sect. 4, numerical examples are presented, and results of four beam search are analyzed and compared.

2 Problem description

The problem in this paper is optimizing the rendezvous sequence for multiple asteroids rendezvous missions. The rendezvous sequence is a list of asteroids and corresponding rendezvous date. Asteroid data comes from JPL Small-Body Database Search Engine [2)]. Altogether 2599 main-belt asteroids with absolute magnitude less than or equal to 14, eccentricity less than 0.2, and inclination less than 3° are used as candidates. This optimization process can be divided into a two-step process. The first step is determining the departure asteroid and departure date. The second step is determining the subsequent asteroids and dates. In this paper, cluster analysis is used in the first step, and beam search variants are applied in the second step, as shown in Figure 1. Their details are given in the following sections.

3 Asteroids clustering

Asteroids can be divided into groups based on their orbits by clustering asteroids. Cluster analysis is a common technology in data mining [27]. Normally, the number of clusters is unknown before clustering asteroids. Therefore cluster algorithms that can determine the number of clusters by the algorithm itself are applied in our problem. In this paper, four cluster algorithms, mean shift [28], affinity propagation [26], DBSCAN [29], and OPTICS [30] are investigated. Asteroids clustering is defined as, at a given epoch, given a transfer time, group asteroids that are reachable to each other.

The distance between two asteroids must be defined before applying cluster algorithms. Euclidean distance is commonly used in clustering analysis, thus improved orbital indicator d_o proposed by ref. [31] is applied so the transfer ΔV between asteroid A_1 and asteroid A_2 is transformed to an Euclidean distance between their improved orbital indicator $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. The improved orbital indicator is calculated as

$$\boldsymbol{x} = \left[\frac{1}{\Delta T}\boldsymbol{r}(t_0) + \boldsymbol{v}(t_0), \frac{1}{\Delta T}\boldsymbol{r}(t_0), \frac{1}{\Delta T}\boldsymbol{r}(t_0+\Delta T) - \boldsymbol{v}(t_0+\Delta T), \frac{1}{\Delta T}\boldsymbol{r}(t_0+\Delta T)\right], \tag{1}$$

where $\boldsymbol{r}$ and $\boldsymbol{v}$ are the states of the asteroid at the given epoch t_0 and ΔT is the given transfer time. The epoch t_0 is a grid of 30 equal parts between January 1 (2025) (modified Julian date MJD 60676) and January 1 (2030) (MJD 62502), and transfer time ΔT is 425 days.

The detailed description of these four cluster algorithms can be found in refs. [26–30]. In this problem, the concerned parameters are, band width for mean shift, preferences for affinity propagation, maximum neighborhood distance and minimum number of neighbor samples for DBSCAN and OPTICS. These parameters affect how tightness the clusters will be. Take DBSCAN as an example, maximum neighborhood distance is the ΔV_{max} between two asteroids and minimum number of neighbor samples is the number of consequent transfers using ΔV_{max}. To make the four algorithms comparable, these parameters should be set properly to reflect similar tightness for four algorithms. In this paper, band width is set to 5×2000 m/s, preferences is set to –5×2000 m/s, maximum neighborhood distance is set to 2000 m/s and minimum number of neighbor samples is set to 5.

Performance of cluster algorithms is evaluated using the following method. For each asteroid in each cluster, Lambert transfers to the nearest N asteroids in the same cluster are calculated. The nearest N asteroids are found using k-nearest neighbors (knn) technology and improve orbital indicator in eq. (1). In knn, a k-d tree data structure is used to find the nearest k asteroids in an efficient way. N is set to 0.1 times the size of the cluster. Then the average ΔV_{avg} calculated using Lambert transfers is used as the performance index of cluster

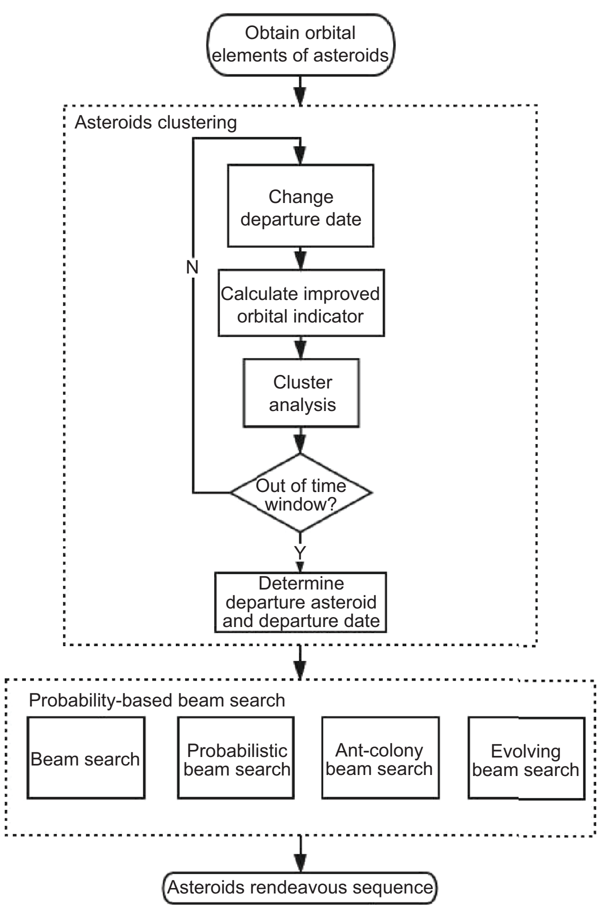

Figure 1 Flowchart.

2) https://ssd.jpl.nasa.gov/sbdb_query.cgi

algorithms. ΔV_{avg} should be small if asteroids are clustered well. The lib used is scikit-learn [32] and other parameters are default. In a cluster, define the central asteroid as the asteroid that has the smallest mean distance from all other asteroids in the same cluster. In this paper, the departure asteroid is the central asteroid of the biggest cluster, and the departure date is the epoch of the biggest cluster.

4 Probability-based beam search

The optimization problem considered in this paper is, how to rendezvous more asteroids and leave more final mass at last, starting from a specific asteroid at a given epoch, and meeting certain constraints. The performance index is defined as

$$h = n + \frac{m - m_{dry}}{m_s - m_{dry}}, \tag{2}$$

where n is the number of visited asteroids, m the current mass of spacecraft, m_s the initial mass, and m_{dry} the dry mass. This performance index means that the more visited asteroids the better, and when the number of visited asteroids is the same the more final mass the better.

To simplify the problem, transfers between asteroids are represented by Lambert arcs. There are two constraints on velocity increment. One is that arcs with velocity increment larger than ΔV_{max} will be abandoned. The other constraint given in eq. (3) is considered so that the Lambert arc can represent a low-thrust arc as low thrust electric propulsion is more suitable for deep space multi-target missions.

$$\Delta V \leqslant \frac{T_{max}}{m}\Delta T, \tag{3}$$

where ΔV is the velocity increment of the Lambert transfer, T_{max} is the maximum thrust of an electric thruster, m is the mass of spacecraft, and ΔT is the transfer time. The transfer time is discrete in a set time window. The mass is updated using

$$m = m^{-\frac{\Delta V}{I_{sp} g_0}}. \tag{4}$$

The search tree is built out of tree nodes. In our problem, a tree node is a single transfer from the start asteroid to the target asteroid, including these information: start asteroid number A_s, target asteroid number A_t, start epoch t_s, final epoch t_f, initial spacecraft mass m_s, final spacecraft mass m_f, and current performance index h.

When searching for the potential childnodes of a tree node, calculating all the Lambert transfers of them is still time-consuming. In order to speed up the search, knn is applied using the improved orbital indicator in eq. (1). Only Lambert transfers to the nearest k asteroids will be calculated. In this paper k is 100.

4.1 Beam search

During the beam search process, breadth-first search is used to establish the search tree, and at each level, all nodes are sorted by their heuristic cost that is the performance index in eq. (2). Only the first beam width bw number of nodes are stored and expanded for next level, and the low ranked nodes are abandoned. The result of beam search is deterministic, that is the result is always the same if the search is run many times. Deterministic beam search is used to refer to the conventional beam search in this paper.

The process for deterministic beam search follows.

Step 1 Set up the root node. In the root node, the target asteroid A_t is the specified first asteroid, the final epoch t_f is the initial epoch, and the final mass m_f is the initial mass.

Step 2 Expand. For each node, the target asteroid is now the start asteroid for the next transfer. Create new transfers to each unvisited asteroid and set the transfer node as its childnode if ΔV constraints are satisfied. All the childnodes make up the nodes for next level.

Step 3 Sort and store. At the current level, calculate the heuristic cost of each node and sort them. Store the first bw nodes.

Step 4 Repeat steps 2 and 3 until the constraints are violated or the scale of new child nodes is zero.

4.2 Probabilistic beam search

Different from deterministic beam search where only the first bw nodes can be extended to next level, in probabilistic beam search all nodes have the opportunity to extend to the next level according to the probability proportional to h. The probability is calculated by

$$p_{ij} = \frac{h_{ij}}{\sum\limits_{l \in T} h_{il}}, \tag{5}$$

where p_{ij} is the probability going to asteroid j from asteroid i, T is the set of feasible components.

To select the childnode when extend to the next level, a probability p_0 $(0 < p_0 < 1)$ is adopted to determine whether the childnode is selected according to probability or is selected in a greedy way. With a probability p_0 the childnode will be selected according to the probability in eq. (5), and with a probability $1{-}p_0$ the best one will be selected as the childnode. If p_0 is 0, then the search becomes deterministic beam search.

4.3 Ant-colony beam search

To introduce the learning mechanism, ant colony optimization (ACO) is hybridized with the probabilistic beam search.

ACO is a stochastic meta-heuristic algorithm inspired by the foraging behavior of ant species and it contains a learning mechanism in which the previous results affect subsequent searches. The generation of multiple asteroid rendezvous sequences is one variation of the classical vehicle routing problem (VRP), for which ACO algorithms are the state-of-the-art [33]. In ACO, the ant selects which edge to go to according to the probability given by

$$p_{ij}^{k} = \frac{\tau_{ij} \cdot \eta_{ij}^{\beta}}{\sum\limits_{l \in T_k} \tau_{il} \cdot \eta_{il}^{\beta}}, \tag{6}$$

where p_{ij}^{k} is the probability of ant k going to asteroid j from asteroid i, τ_{ij} is the pheromone level on the edge, η_{ij}^{β} is the heuristic information of the edge, T_k is the set of feasible components, and β is a weighting parameter. The probability is calculated by two parts: the pheromone and the heuristic information. The edge with a high concentration of pheromone is more attractive and has a high chance to be followed by ants. The heuristic information evaluates the cost spent to go on this edge.

When the ant finishes its construction process, the local pheromone update is performed given by

$$\tau_{ij} = \phi \cdot \tau, \tag{7}$$

where ϕ is the pheromone decay coefficient. Only the pheromone on edges traversed by this ant is updated to encourage subsequent ants to select other edges. If one subsequent ant locates on the site i, the probability to go to the site j will be different, and thus the diversity of the search is ensured.

After all the ants have finished their tour, the offline pheromone update is applied only by the best ant as given by

$$\tau_{ij} \leftarrow \begin{cases} \tau_{ij} + \rho \cdot \Delta\tau_{ij}, & \text{if } (i,j), \quad \text{belongs to best tour,} \\ \tau_{ij}, & \quad\quad\quad\quad \text{otherwise,} \end{cases} \tag{8}$$

where ρ is the evaporation rate, and $\Delta\tau_{ij}$ is the quantity of pheromone laid by the best ant. The updated pheromone consists of the old pheromone left after evaporation and the new pheromone for which the quantity is related to the performance index. The value of pheromone is bounded between $\tau_{\min}$ and $\tau_{\max}$.

In ant-colony beam search, the heuristic information is replaced by the performance index h in eq. (2), $\eta = h$. Each ant perform a probabilistic beam search and the probability is calcualted by eq. (6). The best result in the probabilistic beam search is used as the search result of the ant.

4.4 Evolving beam search

Besides the learning mechanism, the evolutionary mechanism is introduced in evolving beam search to further improve the algorithm. In evolving beam search ACO is replaced by the evolving elitist club algorithm (EECA) proposed in ref. [24]. In EECA, the crucial design is the introduction of the evolving population (EP). The search process of EECA is constructed based on ant-colony beam search. In one iteration after all ants finish their search, the ant with the best search result will be put into the EP. Then operators in EA are applied upon the EP. This process is population evolution. After the evolution, the best ant in the EP is used to update the pheromone using eq. (8). Evolution of the evolving ant population can be achieved by many operators such as mutation operators and cross-over operators, but crossover operators are not applied in our problem because they may generate infeasible solutions. The choice of operators is problem-dependent, as long as they can achieve the evolution of the evolving ant population. In this paper, a mutation operator is performed through the following steps based on ref. [34].

(1) Random selection. Select four members randomly from members of EP. The size of four was proven to be effective [35].

(2) Competition. The one with the best fitness wins, and its genome is copied three times to replace the other three losers. Now the genome of the winner has four copies.

(3) Mutation. Three copies are mutated, and one copy is kept unchanged. Three mutation operators are time mutation, replace, and add. In the time mutation operator, select a visited asteroid randomly and re-set its arrival epoch randomly. Constraints need to be satisfied. In the replace operator, select a visited asteroid randomly, find its k nearest asteroids using knn, and replace it with a random unvisited asteroid in its k nearest asteroids. In the add operator, select a transfer randomly, find the k nearest asteroids of departure and arrival asteroids, and select an unvisited asteroid randomly in their intersection of k nearest asteroids. Insert the selected asteroid in the transfer and its arrival epoch is given randomly satisfying time constraints while other epochs in the sequence remain unchanged. Mutation may create unfeasible solutions. Each operator is carried out N_m times until a feasible solution appears. The genome will keep unchanged if no feasible solution is generated. N_m is 100 in this paper.

5 Numerical simulation

The initial mass of spacecraft is 2000 kg, of which 1200 kg is the dry mass. $T_{\max} = 0.3$ N, $I_{\rm sp} = 3000$ s and $\Delta V_{\max} = 1500$ km/s are used to prune Lambert arcs and calculate mass.

The spacecraft must remain at each reached asteroid for 30 d. The transfer time is between [150 600] days, and a 30 d resolution grid is considered.

5.1 Asteroids clustering

ΔV_{avg} and computing time of four algorithms, cluster algorithms, mean shift, affinity propagation , DBSCAN, and OPTICS, are listed in Table 1. Affinity propagation has the smallest ΔV_{avg} and DBSCAN has the fastest computing time also a second smallest ΔV_{avg}. The difference in the performance of four algorithms may due to their ability to deal with high-dimension data, as the improved orbital indicator is a 12 dimension vector. However, this is just a primary investigation, further work should be done to find or design a better cluster algorithm. In this paper, affinity propagation is selected. Asteroids clusters are shown in Figure 2. There are noise points which do not belong to any clusters in DBSCAN and OPTICS so the number of asteroids in clusters of DBSCAN and OPTICS is less, as can be seen in Figure 2.

The goal of clustering asteroids is to determine the departure asteroids and departure date. Affinity propagation is applied at each epoch in the grid. The number of clusters at different epoch and the number of asteroids in the biggest cluster are shown in Figure 3. The number of clusters at different epoch is between 51 and 79, and the number of asteroids in the biggest cluster is between 82 and 194. The number of clusters reaches a peak near MJD 61700. The cluster of the largest number of core asteroids locates at MJD 62349.83 (August 1 (2029)), and the central asteroid is 12610 Hafez. Therefore, the sequence search will start from asteroid 12610 Hafez at MJD 62349.83. The orbit elements of 12610 Hafez are listed in Table 2. Top 10 biggest clusters at epoch MJD 62349.83 is illustrated in Figure 4. Different colors represent different clusters. The Earth departure is not included in this paper, but it can be taken into consideration by evaluating the cost from the Earth to each cluster.

5.2 Analysis of deterministic beam search

Deterministic beam search with different beam width are used to search for sequences. The beam width is selected in a large range from 10, 20, 100, 500 (1000), and 10000. Each setting is run only once in deterministic beam search. Results are shown in Table 3. In contrast to the intuitive, the optimal index does not increase as the beam width increases. Even a very poor result is obtained when the beam width is 100. The reason can be found by taking the beam width of 10 and the

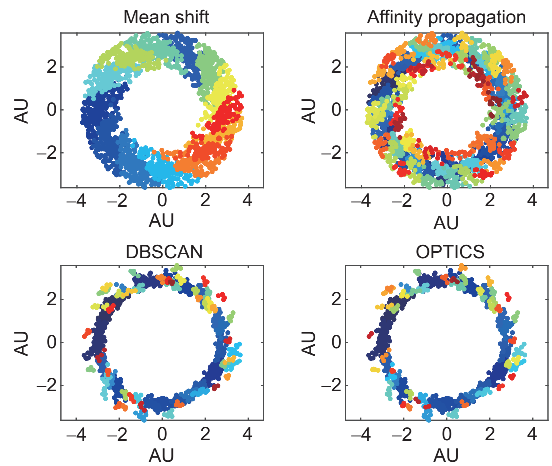

Figure 2 Asteroids clusters for four algorithms.

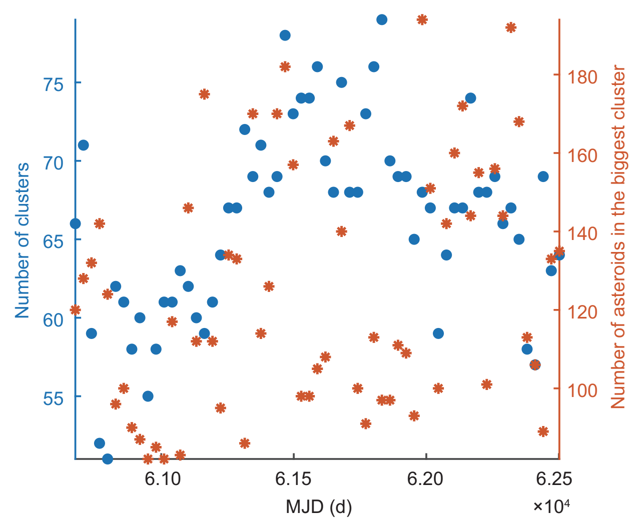

Figure 3 Asteroids clustering results.

Table 1 Performances of cluster algorithms

	Mean shift	Affinity propagation	DBSCAN	OPTICS
ΔV_{avg} (km/s)	58.76	8.24	10.29	10.62
Computing time (s)	16.99	9.72	0.06	3.69

Table 2 Orbit elements of 12610 Hafez at MJD 58600

Semimajor-axis	Eccentricity	Inclination	RAAN	Perigee anomaly	Mean anomaly
2.8472 AU	0.0991	1.7324°	25.4241°	296.1170°	110.0861°

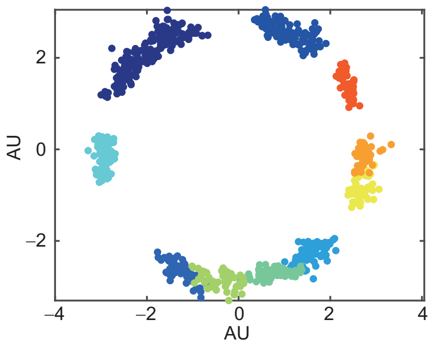

Figure 4 Clusters at epoch MJD 62349.83.

Table 3 Results of beam search with different beam width

Beam width	Index
10	12.0475
20	12.0209
40	12.0076
100	10.0727
500	11.0444
1000	12.0882
5000	12.0834
10000	12.0016

beam width of 20 as an example. The optimal index decreases from 12.0475 to 12.0209 as the beam width increases from 10 to 20. The search process is illustrated in Figure 5.

In Figure 5, the rectangular box represents the nodes that are reserved at each level. The nodes at each level are sorted according to their performance index, and the "index" arrow represents the direction of the performance index from large to small. It can be found that before the Level 4, the beam search result with a beam width of 10 is consistent with the beam search result with a beam width of 20. At Level 4 in the beam search with a beam width of 20, the best nodes are extended by the 18th nodes in Level 3, making the beam search result with a beam width of 20 is significantly better than the beam search result with a beam width of 10. This result is more in line with that reserving more nodes can save more possibility. However, the situation has changed at Level 7. As the tree develops, almost all nodes in the beam search with a beam width of 20 are extended by the 18th node in Level 3, but they are no longer better than nodes in the beam search with a beam width of 10. Selecting better results at Level 4 results in worse results at Level 7. Although more nodes are reserved in the beam search with a beam width of 20, the drawback of beam search as a deterministic and greedy algorithm can still not be avoided.

5.3 Probabilistic beam search

Probabilistic beam searches with the beam width of 20 and 40 run 1000 times respectively. The exploration factor is set to 0.5. Results are listed in Table 4. The mean index, best index, and variance of indexes of each time are given in Table 4. It can be found that even though the beam width is only 20 or 40, probabilistic beam search finds better results than most deterministic beam search. The mean index is also better than that of the deterministic beam search with a beam width of 20 and 40. Seen from the variance, results of the probabilistic beam search with a beam width of 40 are more aggregated. These results indicate that randomness can greatly improve the possibility of finding better results.

5.4 Ant-colony beam search and evolving beam search

Ant-colony beam search runs 100 times. The number of ants is 25. The iteration number of ACO is 50. The initial pheromone τ_0 is 0.05, the $\tau_{\min}$ is $0.1\tau_0$, and $\tau_{\max}$ is 1.0. The pheromone decay coefficient ϕ is 0.9, and the evaporation rate ρ is 0.95. β is selected between 1.0 and 3.0. The beam width is selected between 1 (pure ACO), 20, and 40. Results of ant-colony beam search are shown in Table 5. Evolving beam search runs 100 times using the same settings as ant-colony beam search. The evolving iteration is 100. Results of evolving beam search are shown in Table 6. Evolving beam search is about 1.21 times slower than ant-colony beam search because of the mutation operator in evolving beam search. However, evolving beam search can find the best solution using less iteration, as shown in Figure 6, because EECA use less iteration than ACO to find the best solution [24].

In general, the ant-colony beam search and evolving beam search obtain better results than probabilistic beam search. The best index is 15.0115 found by the evolving beam search

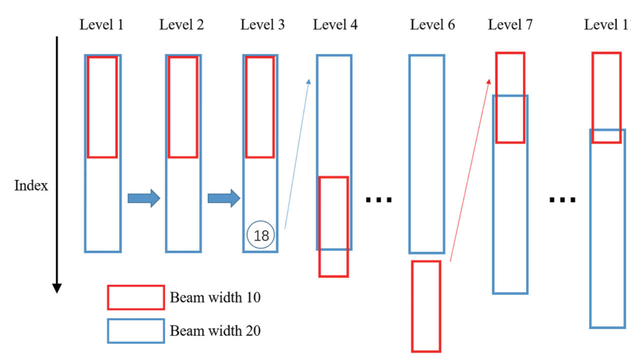

Figure 5 Illustration of beam search.

Table 4 Results of probability beam search

Beam width	Mean index	Best index	Variance
20	12.0265	12.0839	0.0206
40	12.0531	12.0883	0.0003

Table 5 Results of ant-colony beam search

Setting	bw=1, β=1.0	bw=1, β=3.0	bw=20, β=1.0	bw=20, β=3.0	bw=40, β=1.0	bw=40, β=3.0
Mean index	10.8809	11.8089	13.7219	13.9929	13.7864	14.0762
Best index	12.0436	14.0147	15.0050	15.0050	15.0054	15.0054
Variance	0.2982	0.4308	0.2529	0.0862	0.3814	0.0822

Table 6 Results of evolving beam search

Setting	bw=1, β=1.0	bw=1, β=3.0	bw=20, β=1.0	bw=20, β=3.0	bw=40, β=1.0	bw=40, β=3.0
Mean index	11.2090	11.8581	13.7358	13.9959	13.8325	14.1188
Best index	13.0775	14.0527	15.0230	15.0233	15.0115	15.0254
Variance	0.3643	0.4709	0.2856	0.0748	0.3591	0.0188

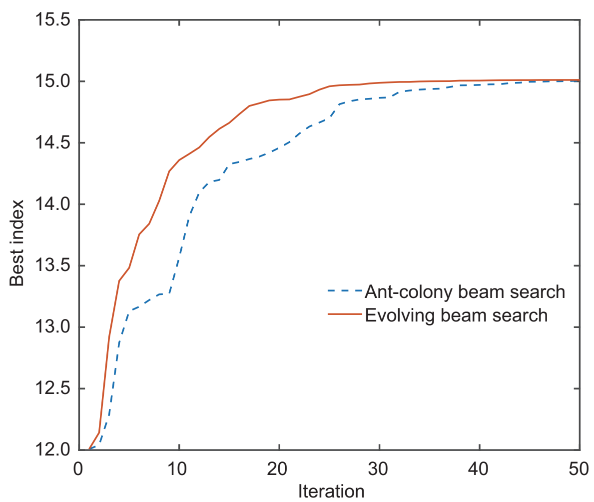

Figure 6 Comparison of ant-colony beam search and evolving beam search.

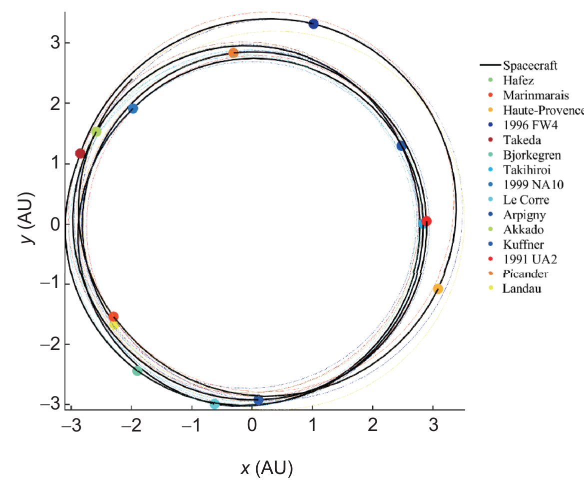

Figure 7 Trajectories of spacecraft and visited asteroids.

with the setting bw=40 and β=3.0, while in deterministic beam search and probabilistic beam search only results of 12 asteroids visited are found. Except the pure ACO, the ant-colony beam search and evolving beam search can find results with mean index better than 13. It can be found that β=3.0 can find better results than β=1.0, which indicates that the heuristic information is still important in the ant-colony beam search and evolving beam search. As the beam width increases, the mean index and bets index also increase. Comparing Table 5 with 6 it can be found that results of evolving beam search have been improved based on the results of ant-colony beam search. In general, the advantage of beam search algorithms with the learning mechanism is obvious, and the evolutionary mechanism has the ability to further improve results. The trajectories of spacecraft and visited asteroids are shown in Figure 7.

5.5 Departure from different clusters

In above simulations, the biggest cluster is used to determine the departure asteroid and departure date. However, it is of interest to investigate cases that depart from different clusters. Results of departure from different clusters using evolving beam search are given in Table 7. Clusters are ranked by their size. It can be seen that departure from the biggest cluster is not necessary to obtain a 15 asteroids solution, but bigger cluster has higher possibility to obtain better results. Further research needs to be done to apply cluster algorithms to sequence optimization problems.

6 Conclusion

Multiple asteroids rendezvous sequence is optimized in this paper. Asteroid clustering is performed to select the first asteroid to visit. Four cluster algorithms are investigated and compared, and affinity propagation is selected. Four beam search methods, deterministic beam search, probabilistic beam search, ant-colony beam search, and evolving beam search are analyzed and compared. Deterministic beam search that is widely used in sequence optimization can find the sequence with 12 asteroids in the beam width range of 10 to 10000. Probabilistic beam search with beam width of 20 and 40 find results with less mass consumption than most deterministic beam search, which indicates that randomness can

Table 7 Departure from different clusters

Cluster	1st	2nd	3rd	10th	50th	100th
Mean index	14.1188	14.1022	14.1123	13.2001	11.1365	12.1492
Best index	15.0115	15.0075	15.0103	14.0323	12.1005	13.0954
Variance	0.0188	0.0238	0.0099	0.0146	0.0591	0.0219

improve the possibility of finding better results. Ant-colony beam search can find 15 asteroids sequences, and results are further improved by evolving beam search. The advantage of beam search with the learning mechanism and evolutionary mechanism is obvious. Further studies can focus on the design of a better learning and evolutionary mechanisms to improve results.

This work was supported by the National Natural Science Foundation for Distinguished Young Scientists of China (Grant No. 11525208), and the National Natural Science Foundation of China (Grant No. 11672146).

1 Yang H, Li S, Bai X. Fast homotopy method for asteroid landing trajectory optimization using approximate initial costates. J Guid Control Dyn, 2018, 42: 585–597
2 Cheng B, Yu Y, Baoyin H. Asteroid surface impact sampling: Dependence of the cavity morphology and collected mass on projectile shape. Sci Rep, 2017, 7: 10004
3 Wie B, Zimmerman B, Lyzhoft J, et al. Planetary defense mission concepts for disrupting/pulverizing hazardous asteroids with short warning time. Astrodynamics, 2017, 1: 3–21
4 Jiang F, Tang G. Systematic low-thrust trajectory optimization for a multi-rendezvous mission using adjoint scaling. Astrophys Space Sci, 2016, 361: 117
5 Li H, Chen S, Baoyin H. J2-perturbed multitarget rendezvous optimization with low thrust. J Guid Control Dyn, 2017, 41: 802–808
6 Chen S, Li H, Baoyin H. Multi-rendezvous low-thrust trajectory optimization using costate transforming and homotopic approach. Astrophys Space Sci, 2018, 363: 128
7 Prockter L, Murchie S, Cheng A, et al. The NEAR shoemaker mission to asteroid 433 eros. Acta Astronaut, 2002, 51: 491–500
8 Farquhar R, Jen S-C, McAdams J V. Extended-mission opportunities for a discovery-class asteroid rendezvous mission. NASA STI/Recon Technical Report A, 1993, 95: 435–447
9 Rayman M D, Chadbourne P A, Culwell J S, et al. Mission design for deep space 1: A low-thrust technology validation mission. Acta Astronaut, 1999, 45: 381–388
10 Russell C, Raymond C. The Dawn Mission to Minor Planets 4 Vesta and 1 Ceres. New York: Springer Science & Business Media, 2012
11 Zhao S, Zhang J, Xiang K, et al. Target sequence optimization for multiple debris rendezvous using low thrust based on characteristics of SSO. Astrodyn, 2017, 1: 85–99
12 Yang H, Tang G, Jiang F. Optimization of observing sequence based on nominal trajectories of symmetric observing configuration. Astrodyn, 2018, 2: 25–37
13 Stuart J R, Howell K C, Wilson R S. Automated design of propellant-optimal, low-thrust trajectories for trojan asteroid tours. J Spacecraft Rockets, 2014, 51: 1631–1647
14 Barbee B W, Davist G W, Diaz S-H. Spacecraft trajectory design for tours of multiple small bodies. Adv Astron Sci, 2009, 135: 2169–2188
15 Izzo D, Becerra V M, Myatt D R, et al. Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories. J Glob Optim, 2007, 38: 283–296
16 Izzo D, Hennes D, Simões L F, et al. Designing complex interplanetary trajectories for the global trajectory optimization competitions. In: Space Engineering. New York: Springer, 2016. 151–176
17 Blum C, Cotta C, Fernández A J, et al. A probabilistic beam search approach to the shortest common supersequence problem. In: Proceedings of the European Conference on Evolutionary Computation in Combinatorial Optimization. Springer, 2007. 36–47
18 Wang F, Lim A. A stochastic beam search for the berth allocation problem. Decision Support Syst, 2007, 42: 2186–2196
19 Li H, Chen S, Yang H, et al. Gtoc9: Results and methods of team 2-Tsinghua University. In: Proceedings of the 31st ISTS (International Symposium on Space Technology and Science), 26th ISSFD (International Symposium on Space Flight Dynamics), 8th NSAT (Nano-satellite Symposium). Matsuyama, 2017
20 Blum C. Beam-ACO—hybridizing ant colony optimization with beam search: an application to open shop scheduling. Comput Operations Res, 2005, 32: 1565–1591
21 Ceriotti M, Vasile M. MGA trajectory planning with an ACO-inspired algorithm. Acta Astronaut, 2010, 67: 1202–1217
22 Stuart J R, Howell K C, Wilson R S. Design of end-to-end trojan asteroid rendezvous tours incorporating scientific value. J Spacecraft Rockets, 2016, 53: 278–288
23 Simões L F, Izzo D, Haasdijk E, et al. Multi-rendezvous spacecraft trajectory optimization with beam p-ACO. In: Proceedings of the European Conference on Evolutionary Computation in Combinatorial Optimization. New York: Springer, 2017. 141–156
24 Li H, Baoyin H. Optimization of multiple debris removal missions using an evolving elitist club algorithm. IEEE Trans Aerosp Electron Syst, 2020, 56: 773–784
25 Gustafson E D, Arrieta-Camacho J J, Petropoulos A E. Orbit clustering based on transfer cost. In: Proceedings of the 23rd AAS/AIAA Space Flight Mechanics Meeting. Kauai, 2013
26 Frey B J, Dueck D. Clustering by passing messages between data points. Science, 2007, 315: 972–976
27 Wu X, Kumar V, Ross Quinlan J, et al. Top 10 algorithms in data mining. Knowl Inf Syst, 2008, 14: 1–37
28 Cheng Y Z. Mean shift, mode seeking, and clustering. IEEE Trans Pattern Anal Machine Intell, 1995, 17: 790–799
29 Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. Port Land: AIAA, 1996. 96: 226–231
30 Ankerst M, Breunig M M, Kriegel H-P, et al. Optics: Ordering points to identify the clustering structure. In: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data and Symposium on Principles of Database Systems. New York: Association for Computing Machinery, 1999. 28: 49–60
31 Hennes D, Izzo D, Landau D. Fast approximators for optimal low-thrust hops between main belt asteroids. In: Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI). Athens, 2016
32 Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in python. J Mach Learn Res, 2011, 12: 2825–2830
33 Dorigo M, Birattari M. Ant Colony Optimization. New York: Springer, 2010
34 Kirk J. Traveling salesman problem-genetic algorithm. 2007. Retrieved from the MATLAB File Exchange website: http://www.mathworks.com/matlabcentral/fileexchange/13680-travelingsalesman-problem-genetic-algorithm
35 Petropoulos A, Grebow D, Jones D, et al. Gtoc9: Methods and results from the jet propulsion laboratory team. In: Proceedings of the 31st ISTS (International Symposium on Space Technology and Science), 26th ISSFD (International Symposium on Space Flight Dynamics), 8th NSAT (Nano-satellite Symposium). Matsuyama, 2017