GTOC12：Methods and Results of ADL

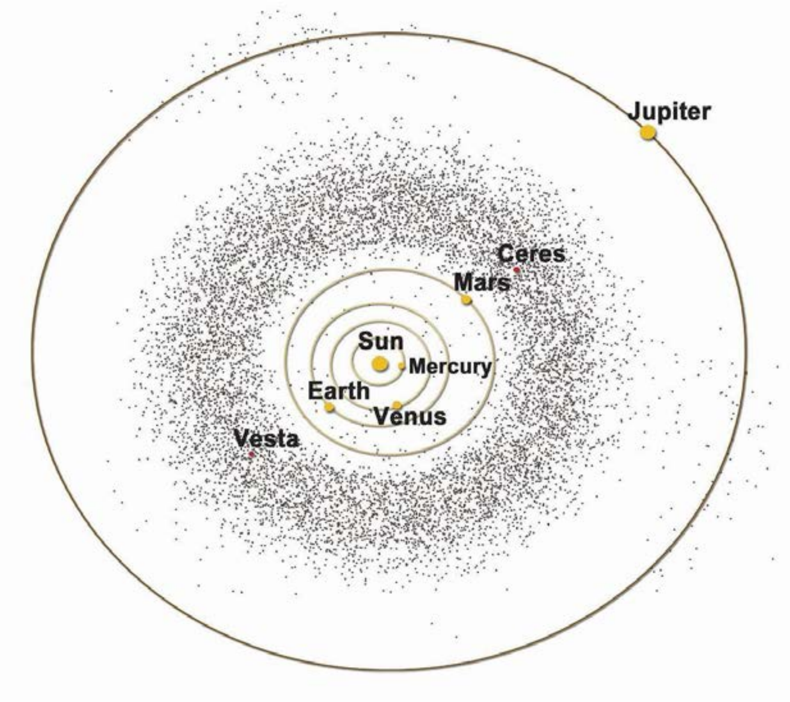
An-yi Huang，Hong-xin Shen， Li Jia， Zhao Li， Zheng-Zhong Kuai
State Key Laboratory of Astronauts Dynamics
15-Jan-24

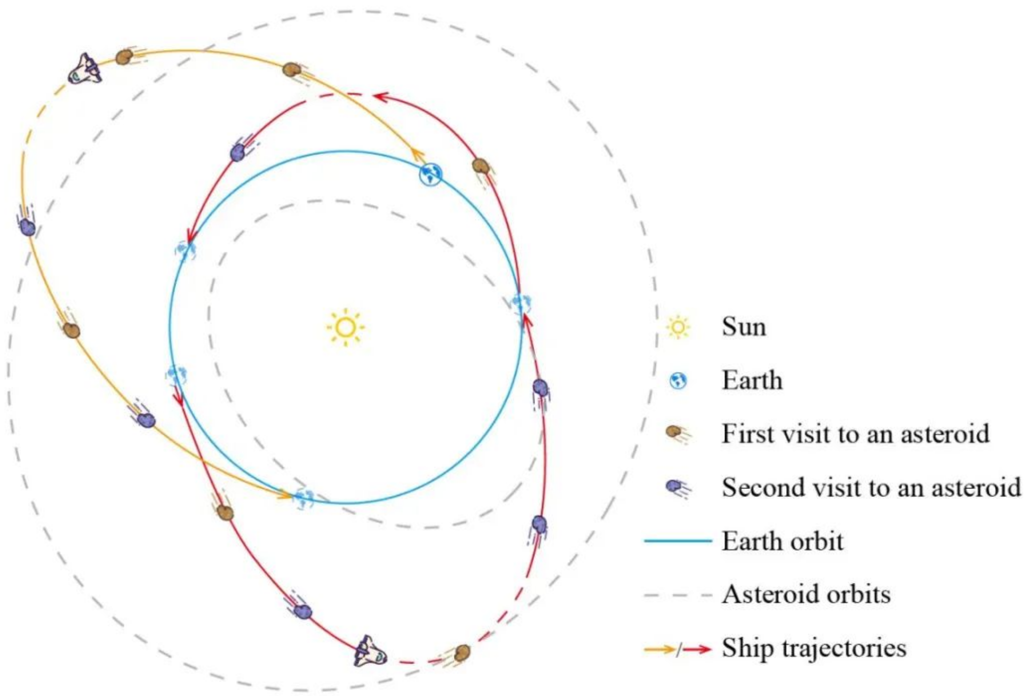
Catalog

Introduction

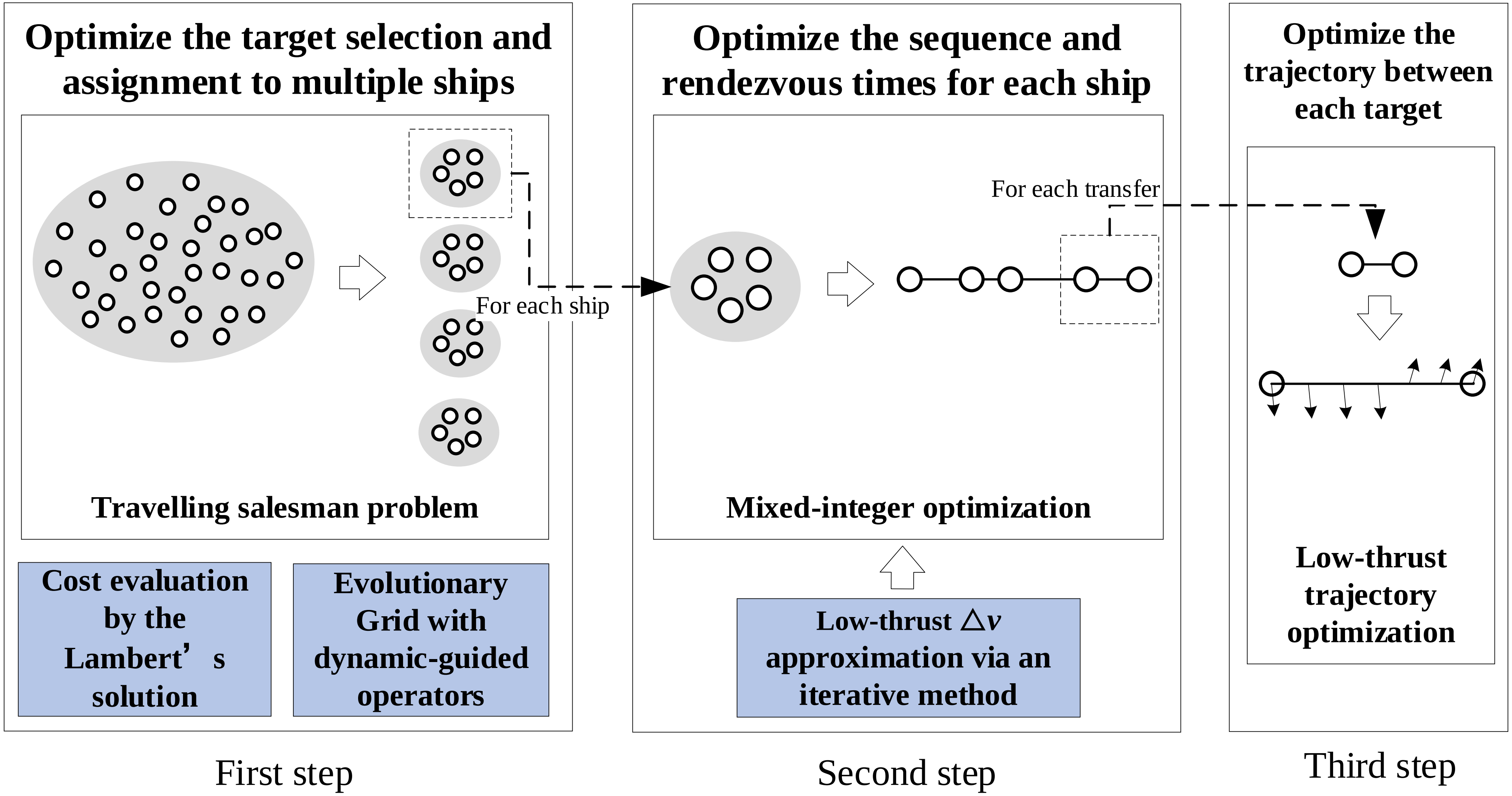
GTOC12 Problem: Sustainable Asteroid Mining

We think the major challenge is the coupling between the mined mass and the number of ships. More collected mass allows more ships and higher score. However, if the constraints are not satisfied, the solution will be completely invalid.





Optimization Framework——Problem Decomposition



[1] Huang, A. Y., Luo, Y. Z., and Li, H. N., “Global Optimization of Multiple-Spacecraft Debris Removal Mission via Problem Decomposition and Dynamics-Guide Evolution Approach,” *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 1, 2022, pp. 171–178

Step I

Two methods were applied:

1. Randomly generate long sequences as many as possible, and then search for the optimal *N* sequences to obtain a combined solution.

2. Evolutionary grid: a fixed-interval grid with customized evolutionary operators for the large-scale orbital travelling salesman problems (detailed in the next chapter).

Step II

The mixed-encoding differential evolution algorithms in [1] is modified:

(1) The initial values are set according to the solution obtained in Step I.

(2) The decision variables include the order of the target asteroid and the rendezvous times.

(3) The objective function is to maximize $\sum_{i=1}^{n_{release}} (t_i^0 - t_i) + \sum_{j=1}^{n_{collect}} (t_j - t_j^0)$, which is equivalent to maximize the mined mass.

(4) Constraints:

The final mass must be greater than $M_{\mathrm{dry}}+M_{\mathrm{collected}}$

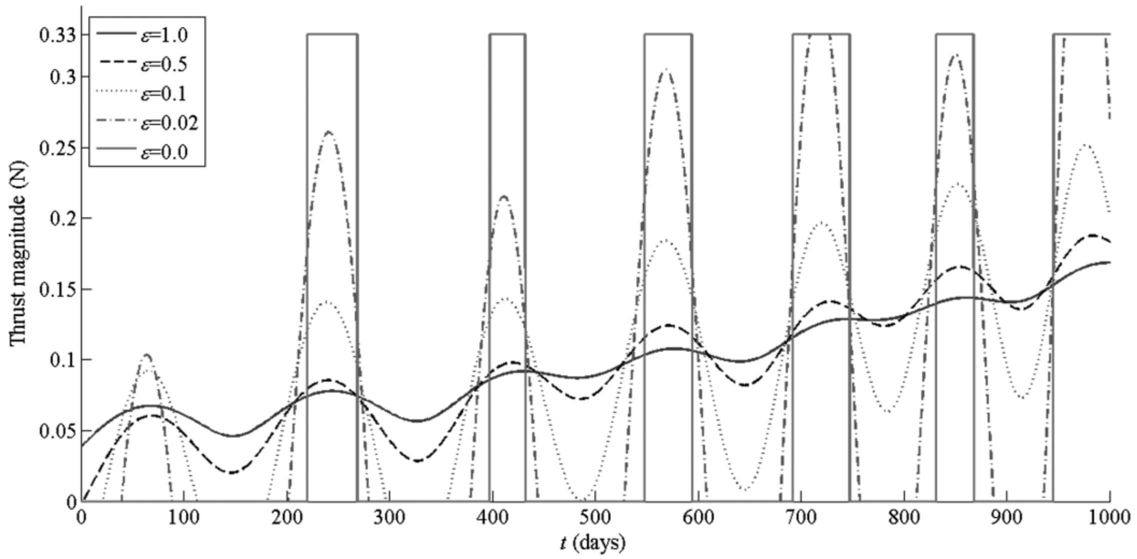
The flight times must be sufficient for low-thrust transfers.

692	1318	25	190	62	156	897	137	1135	314	1152	240	386	175	54	824	532

Adjust the order **Adjust the flight time**

Step III

The indirect method in [2] is applied to solve each low-thrust trajectory with fixed start and arrival times.



[2] Jiang, F., Baoyin, H., and Li, J., "Practical Techniques for Low-Thrust Trajectory Optimization with Homotopic Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 1, 2012, pp. 245–258.

Evolutionary Grid

Encoding the solution via a fixed-interval grid

Ship no	The first rendezvous (release phase)												The second rendezvous (collect phase)																													
1	-1	-1	692	1318	25	62	190	156	897	137	-1	61	1135	-1	-1	-1	-1	314	-1	-1	-1	-1	1152	-1	-1	-1	240	-1	386	-1	-1	-1	175	-1	54	-1	-1	824	-1	-1	532	-1
2	-1	-1	614	659	346	171	37	175	48	94	1207	-1	1207	-1	-1	-1	1175	-1	-1	-1	-1	-1	810	-1	-1	-1	945	-1	-1	77	-1	-1	-1	80	-1	68	-1	-1	487	-1	-1	-1
3	244	-1	108	1073	1170	376	565	-1	266	417	213	-1	-1	-1	453	-1	-1	-1	45	-1	-1	-1	499	-1	-1	-1	-1	960	-1	-1	-1	-1	-1	-1	213	-1	-1	224	-1	266	-1	223
4	-1	-1	246	57	1113	256	670	532	-1	199	195	1223	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	744	-1	-1	1170	-1	-1	1223	-1	-1	284	-1	-1	15	-1	14	-1	62	-1	156	-1
5	-1	-1	15	117	1071	656	1297	-1	1143	857	1312	1178	-1	635	-1	-1	-1	-1	-1	1230	-1	-1	-1	-1	1178	-1	-1	-1	1088	-1	-1	-1	171	-1	137	-1	1113	-1	1102	-1	-1	-1
6	-1	-1	219	338	214	1022	558	-1	437	-1	-1	-1	-1	-1	198	-1	-1	1161	-1	-1	-1	-1	-1	1315	-1	-1	-1	152	-1	-1	-1	-1	-1	617	-1	-1	-1	-1	1318	-1	226	-1
7	-1	-1	202	-1	651	-1	-1	226	1034	287	-1	1230	857	-1	-1	59	-1	-1	-1	670	-1	-1	104	-1	-1	195	-1	-1	-1	78	-1	-1	199	-1	-1	1	-1	-1	11	-1	659	-1
8	-1	9	10	29	1171	1284	-1	1145	230	365	-1	286	250	-1	-1	-1	-1	-1	-1	-1	-1	371	-1	-1	-1	262	-1	-1	-1	-1	-1	-1	1008	-1	-1	414	-1	-1	459	-1	316	-1
9	318	257	75	2	617	-1	11	78	22	1102	810	181	2	-1	-1	-1	-1	-1	-1	774	-1	-1	1073	-1	-1	181	-1	135	-1	-1	565	-1	-1	-1	26	-1	-1	210	-1	-1	-1	-1
10	-1	-1	364	-1	1074	65	59	64	-1	14	-1	184	-1	-1	184	-1	61	-1	-1	202	-1	-1	76	-1	971	-1	-1	1316	-1	-1	105	-1	-1	-1	1071	-1	654	-1	1156	-1	-1	-1

0

Interval: 120 d or 60 d

t

The first target

The last target

-1: Empty node

Evolutionary Grid

Data structure of the nodes

Nodes(1,3)

Ship_index	1
N ode_index	3
A rrive_time	$t_0+2\Delta t$
Astro_index	$S_{1,1}$
A rrive_dv	Δv (Earth,$S_{1,1}$)
R eturn_dv	Null
Is_head	True
I s_tail	False
Front _node_inde	Null
Next_node_inde	5

Nodes(1,5)

Ship_index	1
N ode_index	5
A rrive_time	$t_0+4\Delta t$
Astro_index	$S_{1,2}$
A rrive_dv	Δv ($S_{1,1}$, $S_{1,2}$, 2 Δt)
R eturn_dv	Null
Is_head	False
I s_tail	False
Front _node_inde	3
Next_node_inde	6

Nodes(1,C-3)

Ship_index	1
N ode_index	C-3
A rrive_time	$t_0+(C\text{-}4)\Delta t$
Astro_index	$S_{1,l1}$
A rrive_dv	Δv ($S_{1,l1-1}$, $S_{1,l1}$, $2\Delta t$)
R eturn_dv	Δv ($S_{1,l1-1}$, *Earth*)
Is_head	False
I s_tail	True
Front _node_inde	C-5
Next_node_inde	Null

Ship 1	-1	-1	$S_{1,1}$	-1	$S_{1,2}$	$S_{1,3}$	…	…	…	…	$S_{1,l1-1}$	-1	$S_{1,l1}$	-1	-1	-1

1. Only astro_index is the decision variable. The other variables of a node are re-calculated when astro_index changes during the global search.
2. The time of each node is fixed after the grid has been initialized.
3. The flight time between two targets is determined by the number of empty nodes.
4. The ship no and node index of each target are recorded in a map to quickly locate a visited asteroid in the grid.

Evolutionary Grid

Initialization

1. Set the ship number N and the node number C (C_1: nodes for the first rendezvous, C_2: nodes for the second).
2. For each ship, randomly choose a asteroid that can be launched from the Earth as the start node.
3. Greedily select the fuel-optimal legs one by one from the candidate set (only for the first rendezvous).
4. Obtain the target set (have been visited once). Then, greedily select the fuel-optimal legs for the second rendezvous until the total Δv exceeds the upper limit.

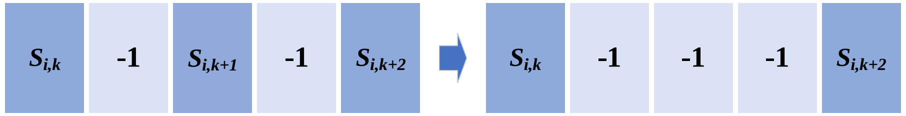
Evolution

1. Obtain n copies of current solution.
2. For each copy, execute four customized mutation operators (detailed in the next page) and update the objective functions.
3. Replace current solution with the best copy after mutation.
4. Repeat steps 2 and 3 until the objective function converges.

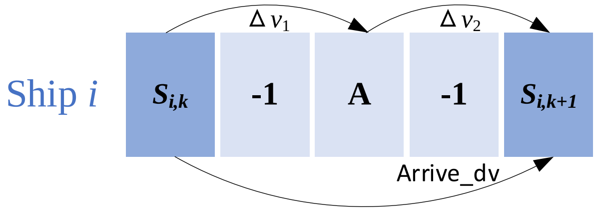
Remark: Multiple solutions can be randomly initialized and evolved independently.

Evolutionary Grid

Cut: Randomly remove *k* couples of targets from the solution.



Insert: 1. Randomly select an empty node in the release phase;
2. Check all the candidate targets:
(1) If the velocity increment of the jth ship won't exceed the upper limit when inserting A, go to (2) ;
(2) If another empty node exists in the collect phase to insert A, add A to the candidate list and record its location.
3. Find and insert the target collecting the most mass in the candidate list.

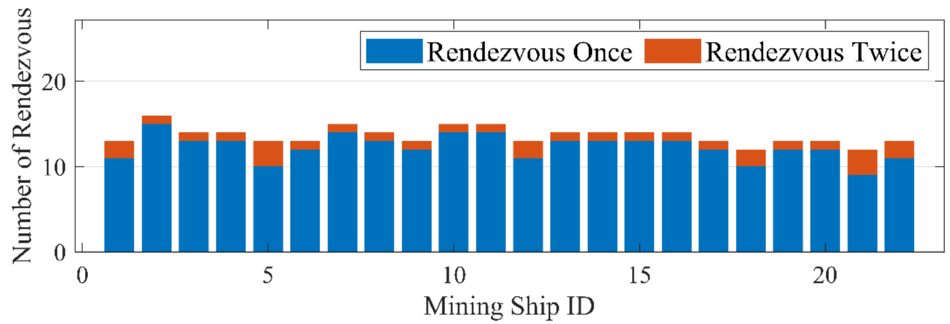


$$\Delta v_{\text{ship } i} = \Delta v_1 + \Delta v_2 - \text{Arrive\_dv}$$

Replace: Do 'cut' and 'insert' for a random selected non-empty node.

Exchange: 1. Exchange the locations of two non-empty nodes (reduce Δv, the obj remains unchanged).
2. Exchange the locations of a non-empty node and its neighboring node (reduce Δv, the obj slightly changes).

Results——Submitted solution

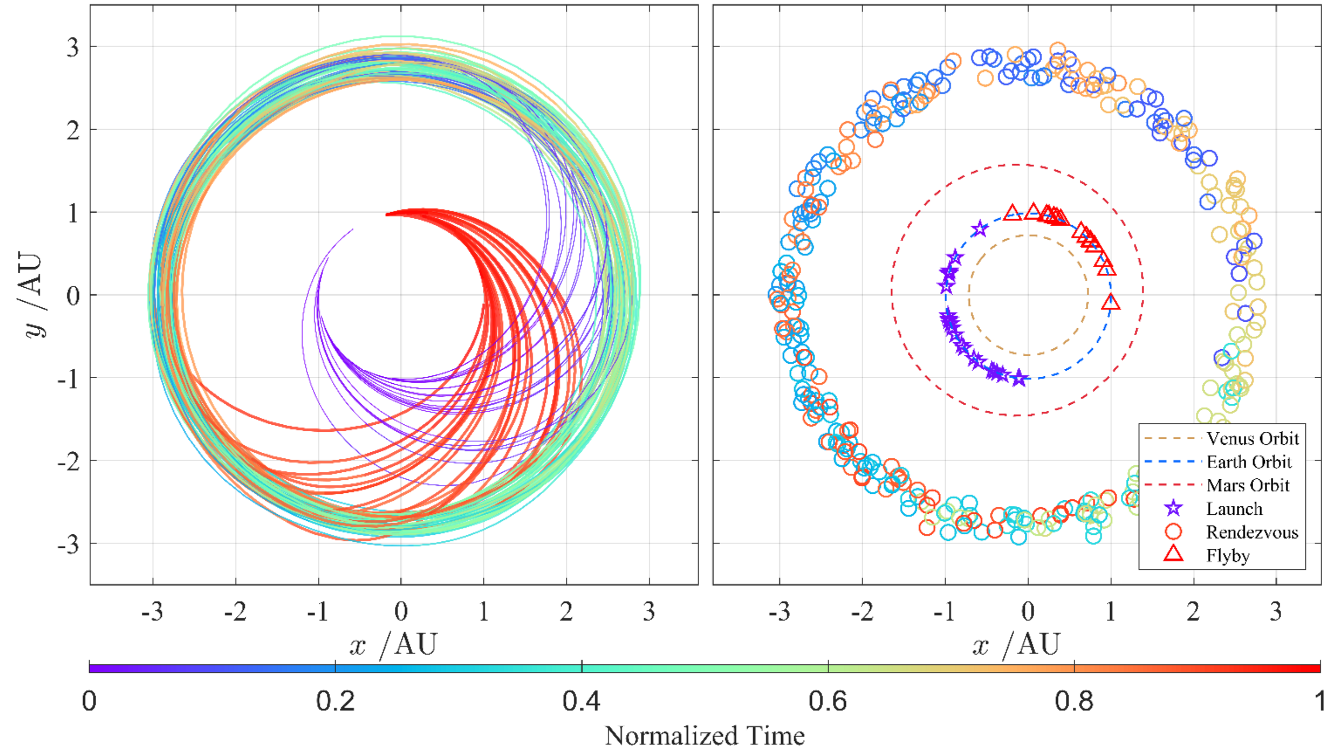


Final Score: 12062

Total collected mass: 13193 kg

Number of mining ships: 22

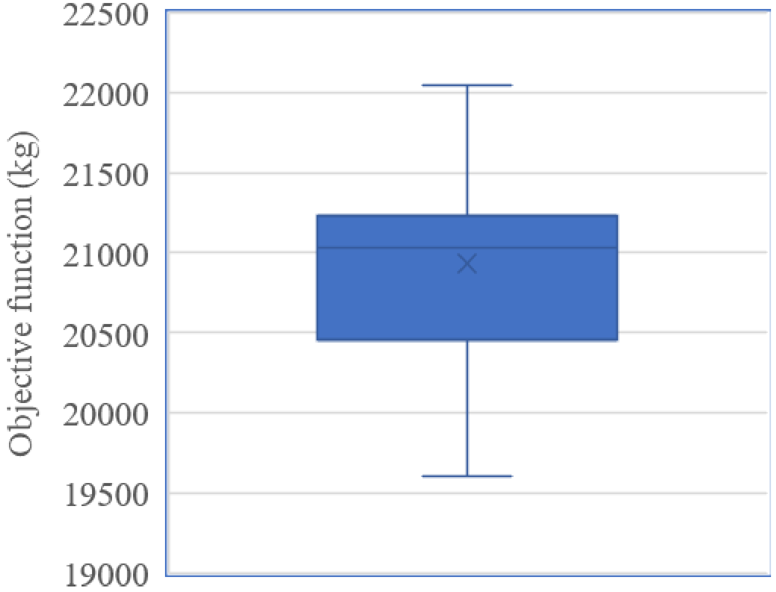
Number of mined asteroids: 165



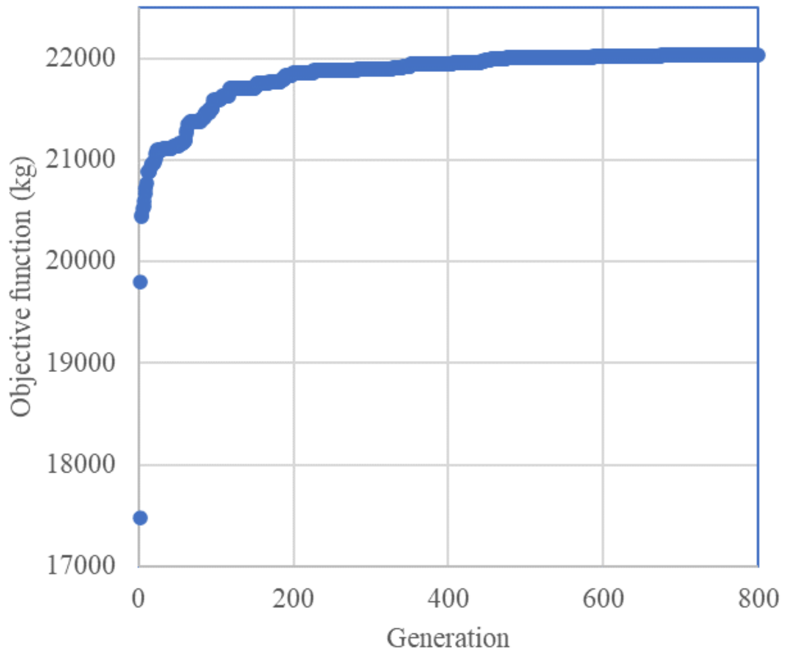
Results——Post competition

N=32

Calculation efficiency: 1 s for one generation (CPU: 4.7GHz)



J of 100 runs



History of *J*

Results

Todo:

1. More Efficiency Δv evaluation for the transfers between asteroid and Earth (including low-thrust propulsion and a launch/return impulse)
2. More accurate Δv evaluation for a single-ship mission.

   When the sequences are re-optimized in Step II, some of them cannot strictly satisfy the constraint on the fuel. Then, small adjustments would be still very time-consuming.

Thanks!

Email: hay04@foxmail.com