11th Global Trajectory Optimisation Competition

'Database Method' used in the GTOC11

Team　：GHWZZ

Members：**He Shengmao**（CSU,CAS）

Wen Changxuan, Wang Yangxin, Zhang Guoxu,

Zhang Yining, Sunyang Yuxi（BIT）

Zhu Zhengfan（CACS）

Zhang Yang（AIR,CAS）

Changsha China

December 18, 2021

Contents

Orbital Parameters of 'Dyson Sphere'

- Preset: e_{Dyson}, i_{Dyson}, Ω_{Dyson}, ω_{Dyson} = 0
- How to determine the key orbital paremeter a_{Dyson}?

First, we consider a low-thrust trajectory from an orbit [a_{ast} 0 0 0 0 0] to Dyson-sphere orbit [a_{Dyson} 0 0 0 0 0] with the minimum transter time t_f .

Then, simplify performance index.

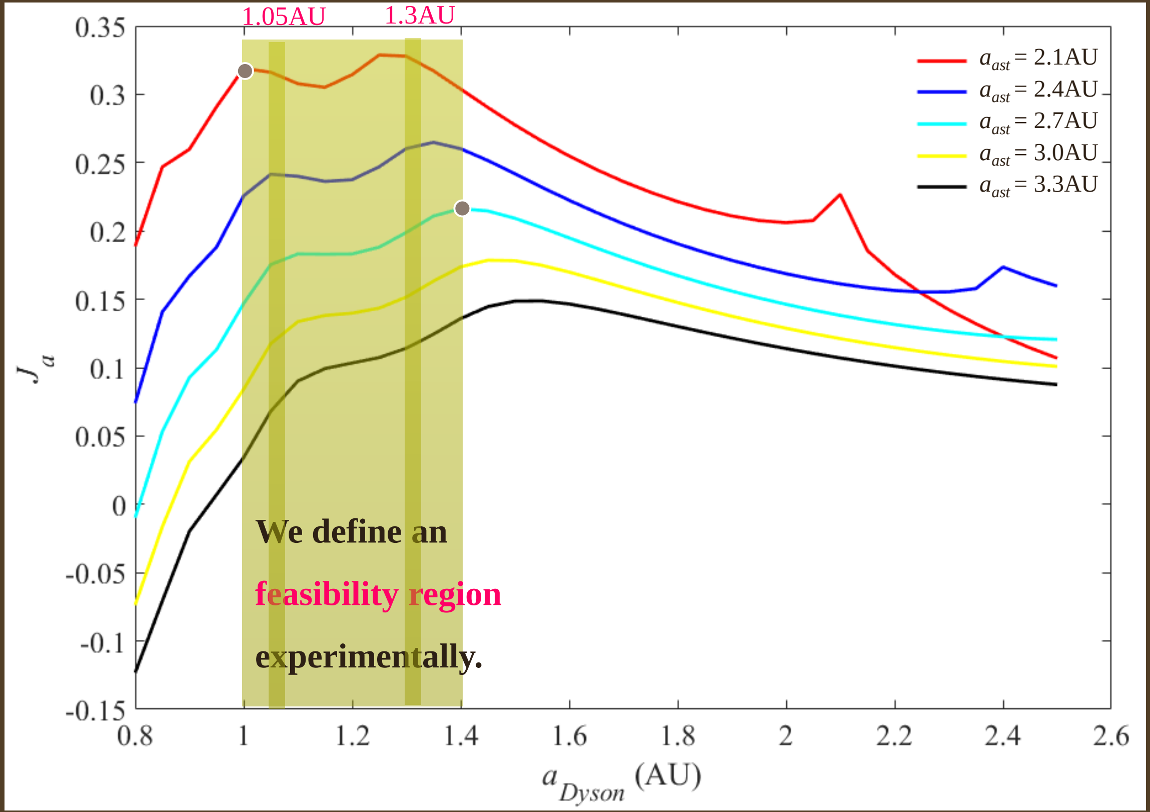
$$J = B \cdot \frac{10^{-10} \cdot M_{\min}}{a_{Dyson}^2 \sum_{k=1}^{10} \left(1 + \Delta V_k^{Total} / 50\right)^2}$$

$$\begin{aligned}
&(1) \quad \Delta V_k^{Total} \equiv Const \\
&(2) \quad M_0 \equiv Const \\
&(3) \quad M_f = c \cdot \left(20 year - 1 year - t_f\right)
\end{aligned}$$

$$J = C \cdot \underbrace{\frac{19 year - t_f}{a_{Dyson}^2}}_{J_a}$$

Finally, we obtain the optimal value of a_{Dyson} through minimizing J_a .

Orbital Parameters of 'Dyson Sphere'



From the figure, we found:

- The value range of a_{ast} is [2.1,2.7]AU.
- The optimal value range of a_{Dyson} is [1,1.4]AU, and the most suitable values are 1.05 and 1.3AU. Finally, we choose 1.3AU.

'Database Method'

Four uniform steps:

- step1： rebuild the asteriod database
- step2： build asteriod-to-asteriod transfer database
- step3： build pointer database
- step4： global search method

step1:rebuild asteriod database

- Asteriod parameters

time from MJD95739

new ephemeris

residual mass for building stations

initial mass

rendezvous-asteriod deadline

newNO	t_0	a	e	i	OM	om	$theta$	m_f	m_0	t_{end}	***old NO***
	0										

- How to determine the values of m_f and t_{end}?

We optimized the low-thrust trajectory from each asteriod to dyson-sphere's orbit to minimize the transter time t_f without considering the rendezvous phase constraint. Then, m_f and t_{end} were finally caculated as below.

$$m_f = m_0\left(1 - t_f * 6e^{-9}\right) \qquad t_{end} = 20year - t_f$$

step1: rebuild asteriod database

- constraint for selected asteriod :

$$m_f > 0.36, a > 2, i < 12°$$

Finally, there exist 50000+ asteriods with m_f > 0 among the total 80000+ asteriods, and we selected 9809 asteriods according to mf .

newNO	t_0	a	e	i	OM	om	$theta$	m_f	m_0	t_{end}	$old\ NO$
1	0	2.2314	0.1950	0.0376	0.6317	4.4024	-1.8687	0.9712	1.9494	6.3071e+03	67396
2	0	2.2310	0.1586	0.0153	3.8791	3.3727	2.7412	0.9579	1.9249	6.3059e+03	83147
3	0	2.1662	0.1927	0.0718	0.7881	4.9136	-3.0255	0.9551	1.9098	6.3107e+03	82747
4	0	2.2003	0.1334	0.0870	1.8047	4.9978	1.9369	0.9432	1.9635	6.2727e+03	82675
5	0	2.2554	0.1817	0.0552	3.2886	2.1039	-2.4674	0.9408	1.9541	6.2747e+03	82552
6	0	2.1956	0.1025	0.0570	0.3760	3.3466	1.8363	0.9403	1.9588	6.2720e+03	73809
7	0	2.2120	0.1412	0.0849	1.8887	5.0698	-0.5732	0.9364	1.9494	6.2726e+03	80910
8	0	2.1502	0.1164	0.0687	1.6740	2.7831	-2.2248	0.9341	1.8742	6.3075e+03	82743
9	0	2.2467	0.1409	0.0665	0.0639	5.4645	2.0683	0.9293	1.9635	6.2590e+03	82943
10	0	2.2153	0.1799	0.0894	2.3332	2.9139	0.4384	0.9246	1.9459	6.2626e+03	82682
11	0	2.2061	0.0998	0.0328	3.3308	5.1963	-2.2280	0.9148	1.9191	6.2656e+03	83385
12	0	2.2291	0.1215	0.0769	0.4787	2.7211	-2.4608	0.9074	1.9260	6.2548e+03	83442
13	0	2.1204	0.1183	0.0439	4.4664	1.5637	1.5049	0.9029	1.7865	6.3210e+03	83191
14	0	2.2478	0.1601	0.0393	6.0092	4.9561	-0.8707	0.9025	1.8503	6.2869e+03	73785

step2： build ast-to-ast trajectory database

- constraint of tansfer trajectory：

$$\text{depart time：}\quad 0 < t_b \le 7300day$$
$$\text{transfer time：}0 < \Delta t \le 300day$$
$$\text{depart impulse：}\ \Delta v_1 \le 2.5km/s$$
$$\text{rendezvous impulse：}0 < \Delta v_2 \le 300day$$

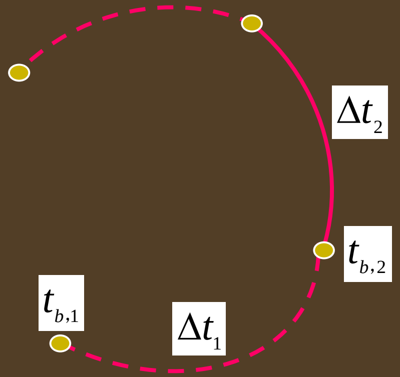
- time discretization：

$$\text{depart time：}\quad t_b = [0,1,...,730]*10days$$
$$\text{transfer time：}\Delta t = [1,...,30]*10days$$

- ast-to-ast trajectory selected constraints：

Could be patched forward and backward！ ！

patch model:



patched backward

patch constraints:

$$t_{b,1} + \Delta t_1 = t_{b,2}$$
$$NO_{arrive,1} = NO_{depart,2}$$
$$NO_{arrive,2} \ne NO_{depart,1}$$

step2: build asteriod-to-asteriod database

Finally, 2.8e8 ast-to-ast trajectories are remained.

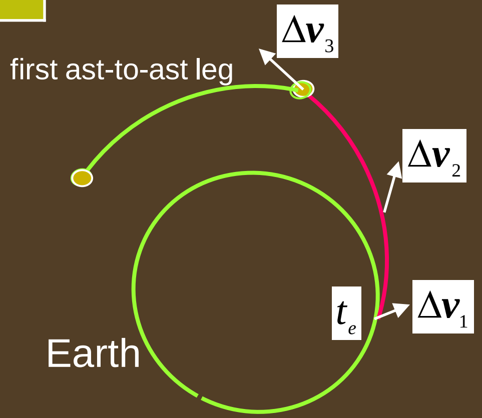
NO.depart	NO.arrive	depart time(*10days)	transfer time(*10days)	detaV(km/s)
1	143	30	18	INF
1	143	30	19	INF
1	143	30	20	3.5
1	143	30	21	INF
...	...	...	...	...

- After that, compute detaV consumed from Earth to each ast-to-ast trajectory.

Adopt three impulses model:

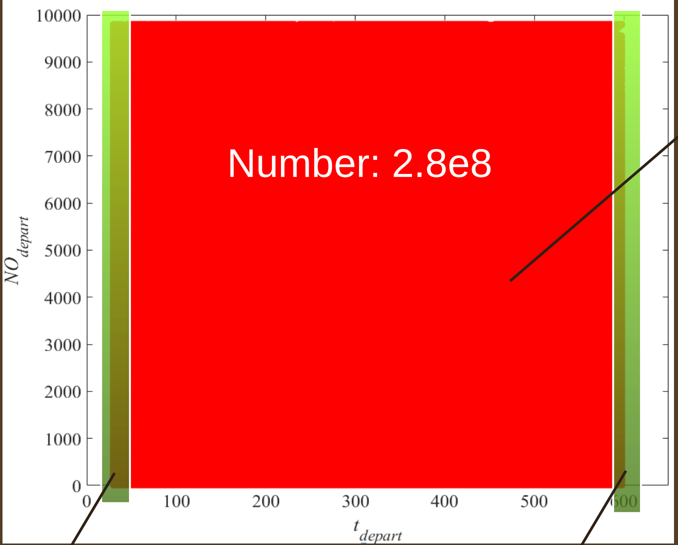
index: $\min\left(\max\left(\left\|\Delta \boldsymbol{v}_1\right\|-6,0\right)+\left\|\Delta \boldsymbol{v}_2\right\|+\left\|\Delta \boldsymbol{v}_3\right\|\right)$

variables: t_e $\Delta \boldsymbol{v}_1$ Δt_1



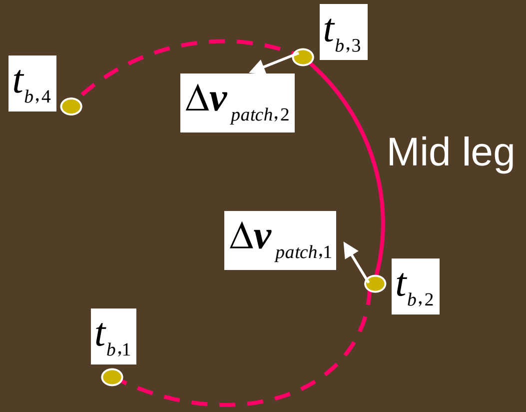
step3：build pointer database

- Downsize the ast-to-ast trajectories database



Middle leg: Could patch forward and backward

$\Delta \boldsymbol{v}_{patch,1}, \Delta \boldsymbol{v}_{patch,1} < 0.9 km/s$



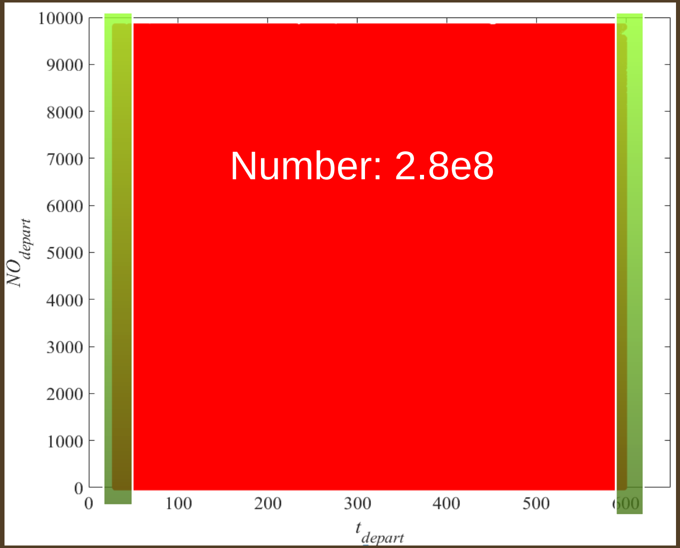
First leg: $t_{depart} < 600 day$

$\Delta V < 4.2 km/s$

Last leg: $t_{depart} > 5800 day$

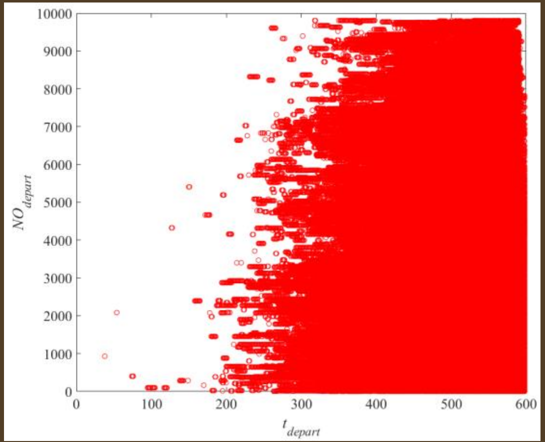
step3： build pointer database

- Downsize the ast-to-ast trajectories database



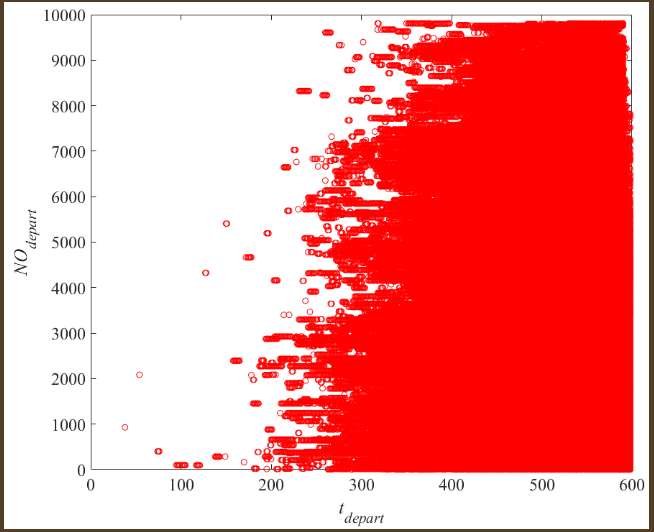
- There are 724 ast-to-ast trajectories satisfied the first leg constraints, and we build 724 ast-to-ast trajectory database according to the first legs.
- There are about 1e5~1e7 ast-to-ast trajectories in each database.

A typical ast-to-ast trajectory database are shown below.



step3：build pointer database

- Build pointer database and index parameter database



There may exist 1~10+ asteriods and 1~100+ ast-to-ast trajectories patched forward for an ast-to-ast trajectory. We set the max number of pointers is 80, and recorded pointer database according to shorter transfer time.

The pointer database is shown as below. to reduce memory consume

database 1

	line_begin	line_end
1	1	10
2	11	11
3	12	19
4	20	25
5	…	…

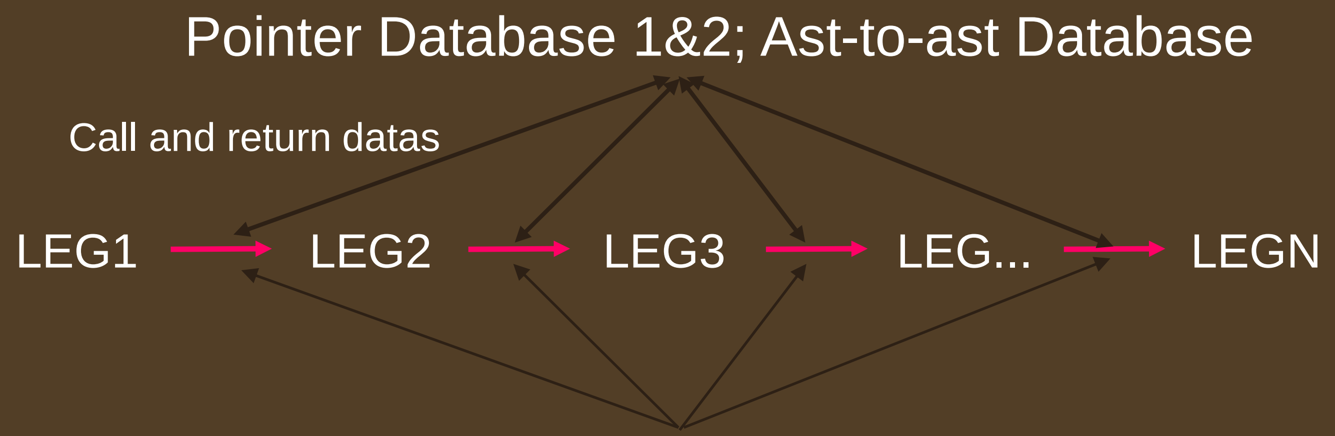
database 2

	line_pointer to ast-to-ast database	detav_add 1e4/VU	mass_add /1e10kg
1	9	9	9 pointer number
2	2	237	8032
3	3	231	8032
4	4	233	8032
…	…	…	…

10

step4:global search method

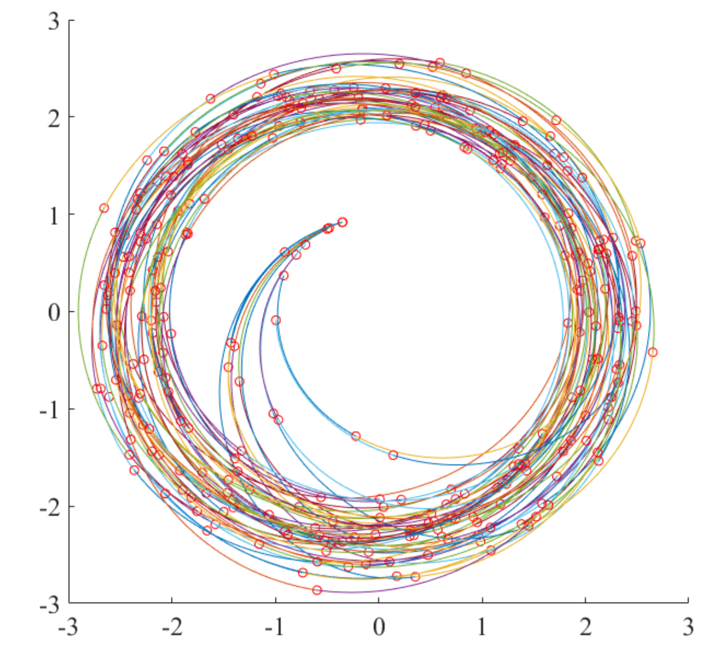
- construct whole trajectories



- Supposed the total pointer number is M, and the pointer will be a random value from 1 to M.
- Cheak the asteriod NOs, if repeated then break.
- 100000 whole trajectories are generated randomly, and record the optimal one.
- 10 trajectories are optimized one by one.
- Finally, we got the optimal solution base on the 9809 asteriods.

Result and Conclution

10 satellite trajectories



total impulses,km/s

15.0254
16.9751
18.6382
17.2610
18.0450
14.7594
16.8250
15.4589
18.6238
16.0347

- The estimated total mass for building each station is 1.57e15 and the estimated Score is 7200+.
- However, we did not complete the low-thrust trajectory optimisation at the specified time.

Result and Conclution

- Conclutions：

(1) Database Method is suitable for global trajectory optimization problem with small- and medium-scale targets .

(2) Database Method is introduced in order to reduce repetitive computation and invalid computation.

(3) Actually, Database Method can be reduce computation effectively, but memory consumption will increase accordingly.

(4) Monte Carlo Method is adopted for global search and it is very suitable.

Thank You !!!