Acta Futura 11 (2018) 71-77

DOI: 10.5281/zenodo.1139250

Acta Futura

# GTOC 9: results from the German Aerospace Center (team DLR)

MARCUS HALLMANN*, MARKUS SCHLOTTERER, ANSGAR HEIDECKER, MARCO SAGLIANO
FEDERICO FUMENTI, VOLKER MAIWALD, RENÉ SCHWARZ

INSTITUTE OF SPACE SYSTEMS, DLR, ROBERT-HOOKE-STR. 7, 28359 BREMEN (GERMANY)

**Abstract.** This paper discusses the methods used by the team from the German Aerospace Center (DLR) for solving the 9th Global Trajectory Optimization Competition (GTOC) problem. The GTOC is an event taking place every year lasting roughly one month during which the best aerospace engineers and mathematicians world wide challenge themselves to solve a nearly-impossible problem of interplenatery trajectory design.

## 1 Introduction

The paper is organized as follows: section 2 summarizes briefly the problem statement; section 3 points out how the overall strategy was developed; section 4 focuses on the combinatorial part of the problem and 5 on the transfer between two debris. Finally, in sections 6 and 7 we discuss the results and draw some conclusions.

## 2 Problem Statement

The task was to design a scenario with $n$ missions which collect a given set of 123 space debris on Sun-synchronous Low Earth Orbits. The following cost function has to be minimized:

$$J = \sum_{i=1}^{n} c_i + \alpha(m_{0_i} - m_{dry})^2 \tag{1}$$

where $c_i$ is the base cost (increasing linearly during the competition time frame from 45 MEUR to 55 MEUR). Each spacecraft initial mass $m_0$ is the sum of dry mass, propellant mass and $N$ times the deorbit package mass: $m_0 = m_{dry} + m_p + Nm_{de}$, with $m_{de} = 30\,\text{kg}$. The $\alpha$ parameter is set to be $2.0 \cdot 10^{-2}$ MEUR/kg$^2$.

In order to control the spacecraft, five impulsive manoeuvres are allowed during the debris to debris transfer in addition to an impulsive manoeuvre at departure and at arrival. The overall time between two successive debris rendezvous, within the same mission, must not exceed 30 days. The deorbit package deployment takes 5 days. That results in a maximum transfer time of 25 days. The time between two missions must be at least 30 days. And the mission must take place between 23467 MJD2000 and 26419 MJD2000. The radius of pericenter $r_p$ is constrained to be smaller than $r_m = 6600\,\text{km}$.

The spacecraft dynamics is described by the follow-

*Corresponding author. E-mail: Marcus.Hallmann@dlr.de

ing set of Ordinary Differential Equations (ODE):

$$\begin{aligned}
\ddot{x} &= -\tfrac{\mu x}{r^3}\left(1+\tfrac{3}{2}J_2\big(\tfrac{r_{eq}}{r}\big)^2\left(1-5\big(\tfrac{z}{r}\big)^2\right)\right)\\
\ddot{y} &= -\tfrac{\mu y}{r^3}\left(1+\tfrac{3}{2}J_2\big(\tfrac{r_{eq}}{r}\big)^2\left(1-5\big(\tfrac{z}{r}\big)^2\right)\right)\\
\ddot{z} &= -\tfrac{\mu z}{r^3}\left(1+\tfrac{3}{2}J_2\big(\tfrac{r_{eq}}{r}\big)^2\left(3-5\big(\tfrac{z}{r}\big)^2\right)\right)
\end{aligned} \tag{2}$$

which describes a Keplerian motion perturbed by an oblate Earth. The orbital elements of the space debris are given for a certain epoch and are propagated via a more simplified model than equation (2):

$$\begin{aligned}
\dot{\Omega} &= -\tfrac{3}{2}J_2\left(\tfrac{r_{eq}}{p}\right)^2 n\cos i\\
\dot{\omega} &= \tfrac{3}{4}J_2\left(\tfrac{r_{eq}}{p}\right)^2 n\left(5\cos^2 i-1\right)
\end{aligned} \tag{3}$$

It can be seen that the ascending node $\Omega$ is the orbital element which encounters the most variations caused by $J_2$. That will have an impact on the overall strategy. For more details on the problem statement refer to the GTOC 9 problem statement [1] or visit https://kelvins.esa.int/gtoc9-kessler-run/.

## 3 Overall Strategy

The problem to be solved can be classified as Time Dependent Traveling Salesman Problem, with a nested optimal control problem for each transfer. One way to solve the combinatorial part would be to explicitly evaluate all possible combinations. That approach is only applicable for small dimensions. In this case there are 123 debris to be sorted for the best sequence (which gives 123! permutations). In addition they have to be chopped into $n$ missions, which increases the dimension of the problem even more. Even with the most powerful computers and the smartest approach to calculate the $\Delta V$ for transfer from one debris to the next one it would take years to determine the entire tree. Section 4 deals about the solution of the combinatorial part of the problem.

Before looking into the transfer it makes sense to analyze the design space to get some reasonable boundaries for design variables like transfer time, delta v range, needed number of missions and so on. The first important question to answer is how are the debris pieces spread out regarding inclination $i$, eccentricity $e$ and semi major axis $a$. Figure 1 shows the range of the orbital elements for the debris pieces. It can be seen that the orbits are nearly circular, inclination ranges from 96 deg to 102 deg and the orbital height (or $a - r_{eq}$) goes from 600 km to 900 km. These elements do not change over time for the dynamic model which is used for the debris.

If we only consider the change in inclination and semimajor axis during a transfer, the problem can be treated as a simple Traveling Salesman Problem (TSP). The $\Delta V$ which is needed to travel from debris A to debris B is the sum of the inclination change $\Delta V_{inc}$ plus the $\Delta V_{sma}$ for the Hohmann transfer:

$$\begin{aligned}
\Delta V_{inc} &= 2V\sin((i_A - i_B)/2)\\
\Delta V_{sma} &= \sqrt{\mu/r_1}(\sqrt{2k/(1+k)}-1)+...\\
&\qquad \sqrt{\mu/(r_1 k)}(1-\sqrt{2/(1+k)})\\
\Delta V_{AB} &= \Delta V_{inc} + \Delta V_{sma},
\end{aligned}$$

where k is the ratio between $a_A$ and $a_B$, assuming circular orbits. With that equations it is possible to set up a cost matrix showing the cost for the transfer from one debris to the next debris, ignoring the phasing in true anomaly and right ascension. With these assumptions it is possible to apply a genetic algorithm implemented in *Matlab* to find the optimal route. Figure 2 shows the result of one run with 500 populations and $1 \cdot 10^5$ iterations. The total distance is around 2654 m/s, so an average $\Delta V$ of 21.7 m/s is needed for one transfer. In theory and with no constraints on the mission time, this result would equal to $J < 100$ MEUR. In practice the 8 years mission time constraint and the 25 day transfer time constraint has to be fulfilled.

So some reasonable $\Delta V$ has to be invested for changing the right ascension. That can be done in two ways:

- a direct plane change,
- an indirect plane change via a change in semi major axis.

Assuming that all other elements besides $\Omega$ are the same, one can compare the $\Delta V$ for both cases. With this assumption the equation for the direct plane change is similar to the one used for the inclination change.

$$\Delta V_{\Omega} = 2V\sin((\Omega_A - \Omega_B)/2) \tag{4}$$

Like the inclination change it is quite a cost intensive maneuver. If there is enough time available, an indirect transfer is cheaper. An important figure to look at is the


DOI: 10.5281/zenodo.1139250

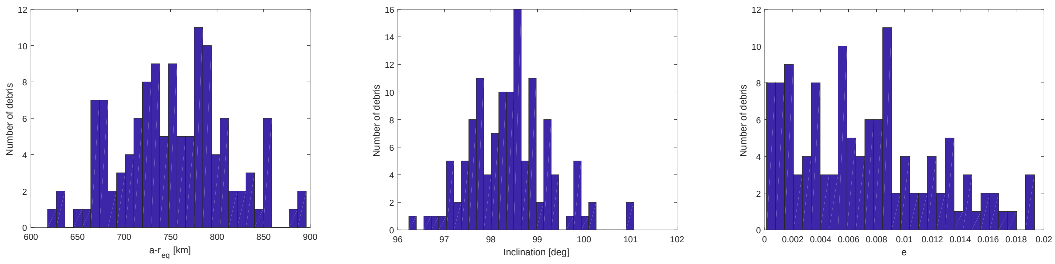


**FIGURE 1.** *Debris orbital elements*

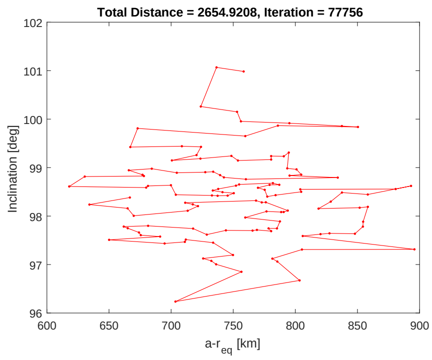


**FIGURE 2.** *Optimal Path in the $i$ and $a$ TSP*

change in right ascension for the debris which is plotted in Figure 3 over height (with an inclination equal to 98 deg). Out of that one can see that the orbits are

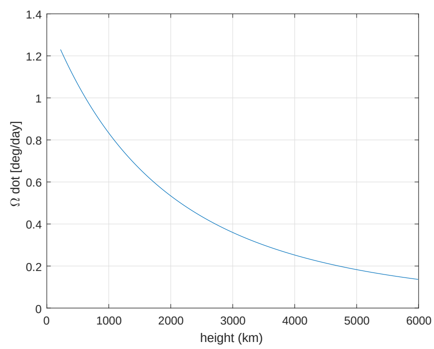


**FIGURE 3.** $\dot{\Omega}$ *over height*

drifting between 1.2 deg/day and 0.2 deg/day. Depending on the initial height the differential drift is around 0.5 deg/day. Assuming a 20 day transfer time it is possible to overcome a delta of 10 deg in $\Omega$. For the indirect change two Hohmann like transfers are needed. The first one to reach the desired drift orbit, the second one to get the semi major axis of the arrival debris.

Assuming the same orbital elements for both debris as: $a = 600\,\text{km} + r_{eq}$, $e = 0$, $i = 98\,\text{deg}$, only a change in $\Omega$ needs to be considered. On Figure 4 one can see

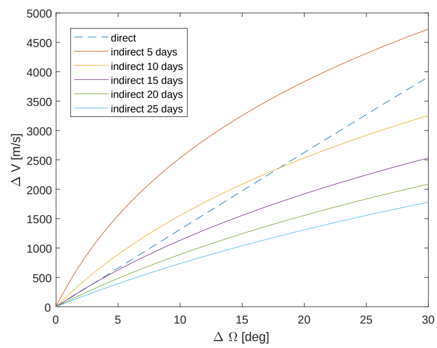


**FIGURE 4.** $\Delta V$ *over* $\Delta\Omega$

that the $\Delta V$ needed for a direct change increases nearly linear with $\Delta\Omega$ (dashed line). The other curves in the figure represent an indirect transfer for different transfer times (5:5:25 days). One can see that for transfer times larger than 15 days it is always better to make an indirect transfer. And that does even not take into account that it is possible to save some $\Delta V$ because of different semi major axis and inclination of the departing and arrival debris. The inclination change can either be performed before or after the drift change maneuvers. This choice also has an impact on the required $\Delta V$, as it is a function of the inclination (see equation 3). With that thoughts one has a good set up for the combinatorial problem, which will be discussed later.

The next interesting question to look at is how many missions one may need (does it make sense to stack one launcher as full as possible) and what does the cost function look like. Assuming a range of different av-

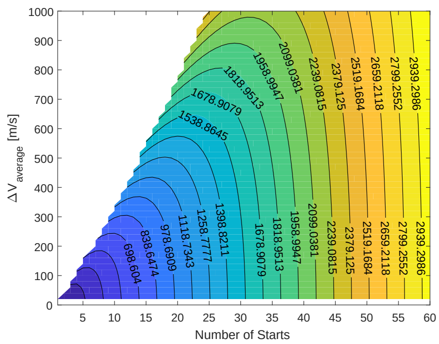


**FIGURE 5.** *J over n and $\Delta V_{average}$*

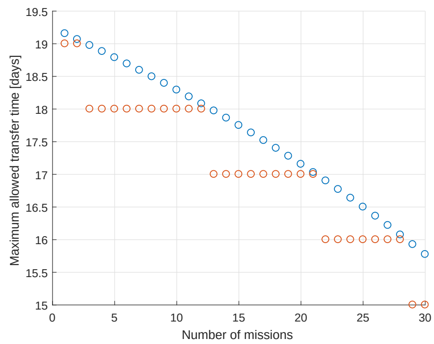


**FIGURE 6.** *Maximum transfer time over $n$*

erage transfer $\Delta Vs$ and number of missions ($n$) one gets Figure 5, which shows the $J$ function in MEUR. It can be seen that it is not advisable to use the maximum propulsion available. For an average $\Delta V$ of 300 m/s, $J$ is 904.1 MEUR for 9 starts, while for 12 starts it is 827.6 MEUR and for a larger number of starts $J$ increases again. Although one would have to add 12 times the base cost for the launcher instead of only 9 times, the used propellant mass goes in quadratic. So using the total allowed 5000 kg is not optimal. It is better to reduce the total allowed fuel or $\Delta V$ per mission by 10% to 20%, depending on how many missions are needed. The issue is that this number is not known before hand.

## 4 Combinatorial Problem

In section 3 basic figures have been derived to narrow down the problem. With these figures it is possible to estimate the cost or $\Delta V$ to perform a debris to debris transfer. The issue is, the problem is time variant, because the cost matrix depends on the epoch of the transfer. Or using the TSP syntax it is not a city to city routing problem, it is a boat to boat one, where the boats are sailing around.

Before going into detail of the graph implementation a deeper look into the handling of the transfer time is needed. As derived in section 3, for the drift change maneuver, it would be better to have a large transfer time available. At a first guess it might make sense to use the maximum allowed 25 days. But using that for all of the transfers, the total mission time would be larger than the allowed 8 years. And that even depends on the number of needed missions. Figure 6 shows the maximum allowed transfer times over the number of needed missions, which is again not known before solving the graph problem. In fact it is only possible to use average transfer times between 15 and 19 days. In that case the next question is, if it would make sense to have the transfer time as a design variable in the combinatorial problem or to keep it fixed to an average value. To not further increase the permutation space it was decided to keep it fixed. That approach also allows to use a look up table for all possible transfers with a 1 day grid size. This table is precalculated and loaded into the graph algorithm. In the cost matrix it is also possible to handle the radius of periapsis constraint, by setting the $\Delta V = \infty$ for all transfers where $r_p < r_{p_m}$.

For solving the routing of the time dependent problem the same genetic algorithm has been used, which already delivered good results for the inclination-sma routing problem. But it didn't brought any good results. Instead a graph algorithm which uses a certain beam width has been developed.

Each mission can be represented as a graph or tree (see Figure 7). For the first mission there are 123 nodes or debris as an option to start from. Keeping in mind that the cost function doesn't give any penalty at which debris the mission starts, it's a free design parameter. So the initial beam width would be 123. Using that one can calculate 122 possible debris transfers for each of that 123 debris. Because each debris should be only visited ones. This results in 123 times 122 possible options. There are many methods to explore such kind of trees or graphs:

- Depth First Search
- Breadth First Search
- Beam Search

 DOI: 10.5281/zenodo.1139250

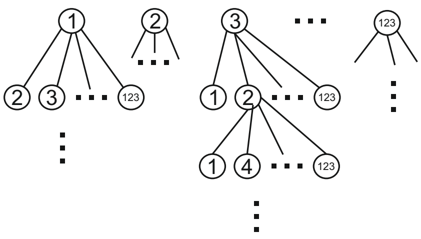


**FIGURE 7.** *Mission Graph*

- Greedy Search

The Depth First Algorithm travels along the left or right side of the tree. In this case it would just visit the debris either in the sequence 1:1:123 or 123:-1:1. It would be possible to add some backtracking if the algorithms gets stuck somewhere or to apply some heuristic, like the average $\Delta V$ is getting too large along the current path.

Breadth First in the opposite explores the tree first horizontal and than continues to the next level with all possible permutations. In this case like mentioned already a full breadth first search would not be possible cause the permutation would be to large.

Greedy search is like depth first, but it makes a decision on some heuristic which path to take. The easiest implementation is to make the decision on the next shortest (lowest $\Delta V$) path. That was the first method used to find a proper sequence and it brought results for $J$ around 2500. The Greedy search has the typical tree drawback that the best cookies are eaten first and the bad ones are left over in the end, and one still has to eat them, cause the entire set has to be collected and not only a subset. And that's something that has been observed when running a greedy search on the tree. The last transfers had a quite high $\Delta V$ and that caused a high number of missions and a resulting high $J$ value.

In order to overcome that issue a beam search has been applied. Instead of only travelling along one path, $k$ best options are selected. Depending on the beam width $k$ the computational time of course grows in that case. For this problem an initial beam width of 123 has been selected for the first mission. On the next level the maximum beam width is already $123 \cdot 122 = 15006$ and so on. Limiting the beam width for the first runs to 2000 already gave good results for $J$ around 1000. The entire idea behind that method is, that it is possible to look into the future by taking also some bad paths hopping they turn into golden paths in the end.

At each level each possible solution has been checked for uniqueness. That can also be explained when looking again at Figure 7. The sequence 1-3-2 is equal to 3-1-2, because for the next level the start node (in that case node 2) as well as the left over debris-set is the same. So the algorithm only takes the best sequence out of that two, because the beam width is limited and many different permutations are needed to find the golden path.

The graph algorithm has been implemented in *Matlab* and took roughly 1 hr computation time on a Intel Xeon CPU E3 3.50GHz, with a beam width set to 20000. Running the tree after the first mission the left nodes are getting less and the computation time goes down.

With the 5000 kg propellant a maximum $\Delta V$ of 5000 m/s can be achieved. But with the results from the qualitative $J$-function analysis the maximum has been set to 4500 m/s and also runs with lower values have been performed. For the first mission it was possible to perform 23 transfers. But the beam width at that point was only around 10 to 20. So in that case there are not enough permutations for the next missions. Instead of taking the maximum transfer solution one with a higher beam width has been taken. The algorithm has been set up in a way that the number of transfers is the same for all beams. That may not be the optimal choice and some further investigation may be performed to see if a free number of transfers brings a significant improvement.

The final sequence will be discussed in the Results section 6

## 5 Transfer Problem

Before solving the debris to debris final transfer the sequence coming out of the beam search algorithm needs to be re-optimized. As already discussed in section 4, the transfer time for all transfers has been kept to a fixed value. That has been re-optimized using the local optimizer *fmincon* in *Matlab*. The cost function in that case is the sum of the $\Delta V$s for all transfers in that particular mission. The design variables are the transfer times. The upper bound has been set to 25 days, constrained by the problem statement. The lower bound was set to 1 day, cause some time for the final phasing of the true anomaly may be needed, which has been ignored completely so far. In the combinatorial part a fixed grid size for the transfer time has been used (e.g. always 17 days

DOI: 10.5281/zenodo.1139250

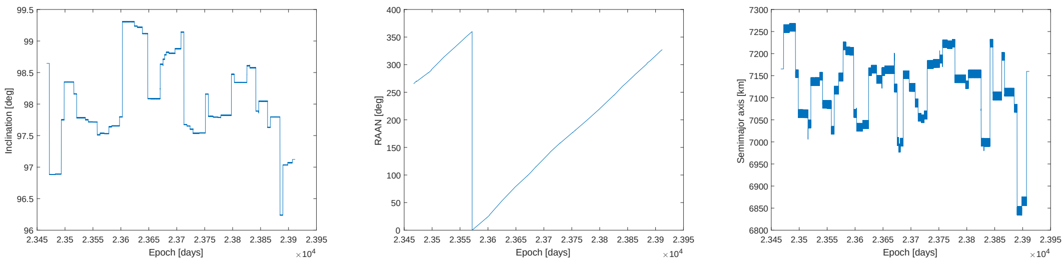


**FIGURE 8.** *Evolution of orbital elements for mission 1*

for the first 6 missions and than 19 or 20 for the remaining, depending how many debris is left to collect). An inequality constraint had to be introduced that the sum of the transfer times is not larger than the old sum of the fixed transfer times (That means one can only shift some transfer time from one transfer to the other). For the re-optimizer the look up table for the $\Delta V$ was not used. Instead it was calculated during the *fmincon* call. That has the advantage of getting a real value for the transfer time. With that approach between 10% and 25% $\Delta V$ per mission has been saved.

For each transfer the following information is known:

- departure epoch
- arrival epoch
- transfer time
- estimated $\Delta V$

To solve that problem again *Matlab fmincon* with an interior point method has been used. The control parameters are the times between maneuvers and the thrust of 5 maneuvers itself in cartesian form. The cost function is quite easy in that case, it's just the sum of all 5 $\Delta V$s we applied. There are 3 deep space maneuvers in addition to one at departure and one at arrival. The more demanding part is the constraint function passed to *fmincon*. Here the equations of motion are integrated between the maneuvers until we reach our final state. Than the final state should equal the arrival debris state at that time. There is a global parameter in order to activate or deactivate the constraints, and it is possible to choose between the cartesian state vector, Keplerian elements, or a mixture, or a subset. Another inequality constraint had to be introduced, taking care that the sum off all transfer times between maneuvers is not larger than the transfer time from the tree, otherwise the following transfers are messed up.

When using an ODE-solver in *fmincon*, the integration errors from the ODE solver may disturb the Jacobian or Hessian. That was the case for the first runs. An investigation has been performed which ODE solver brings reasonable results. The conclusion was that a fixed step size is more stable than a variable step size solver. In the end a RK8 has been used, implemented in C++ with a step size of 50 s (the RK4 needed 1 s step size), cause the *Matlab* implementation was to slow. An interesting observation was also that when using the debris dynamic model first and rerunning the optimizer with the spacecraft ODE, faster and better results have been achieved.

For the initial guess the first and last two maneuvers have been set to the Hohmann like maneuvers coming out of the drift strategy. The maneuver in the middle is set to 0. The inclination and phasing change has been solved by the optimizer. The first and last transfer times where set to a half orbital period. And the remaining two transfer times where chopped up equally (kind of mid course maneuver).

One result is that scaling the state vector $x$, the constraints and the cost function all close to 1 is crucial for success. Although one would assume that this techniques should be handled by optimizers automatically, that seems not to be the case.

With the RK8, the algorithm took roughly 5 minutes on a Intel Xeon CPU E3 3.50GHz. And all transfers converged proper. The achieved $\Delta V$ was even lower than the estimated one out of the tree search, cause in the tree search the $\Delta V$ for the inclination change and the drift maneuver has been added separately. In practice they can be combined and the optimizer seems to have taken care of that.

 DOI: 10.5281/zenodo.1139250

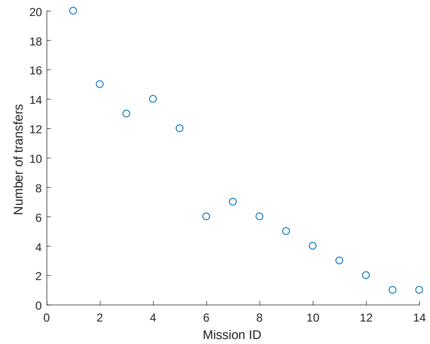


**FIGURE 9.** *Mission Graph*

## 6 Results

The solution submitted by the DLR team had a total of $n = 14$ missions and a performance index $J = 949.85$ MEUR. Figure 9 shows the number of transfers per mission. For the first mission the evolution of $\Omega$, $a$ and $i$ are plotted over time (see Figure 8). It can be seen that mainly the inclination and semi major axis was changed by the $\Delta V$ and the right ascension just drifts along to the next target.

## 7 Conclusion

For the combinatorial part genetic algorithms are suitable when the problem is time invariant. But for time variant problems graph algorithms seem to be the better choice. The Beam search algorithm brought reasonable results, but still suffers a bit from the greedy effect: there are good sequences in the beginning but bad ones in the end. One option to improve that may be to select some feasible continuation beams randomly. In the transfer problem one may use a multiple shooting method to get rid off the ODE-integration issue.

## References

[1] D. Izzo and M. Märtens. The Kessler Run: On the Design of the GTOC9 Challenge. *Acta Futura*, 11:11–24, 2018.

[2] M. Cerf. Multiple Space Debris Collecting Mission - Debris Selection and Trajectory Optimization. *Journal of Optimization Theory and Applications*, 156:761–796, 2013.

[3] Andrew John Drake. *Approaches for solving some scheduling and routing problems*. PhD thesis, University of Southampton, February 2009.

[4] R. C. Engels and J. L. Junkins. The gravity-perturbed Lambert problem: a KS variation of parameters approach. *Celestial Mechanics*, 24:3–21, May 1981.

[5] Hernán Abeledo, Ricardo Fukasawa, Artur Pessoa, and Eduardo Uchoa. The time dependent traveling salesman problem: polyhedra and algorithm. *Mathematical Programming Computation*, 5(1):27–55, March 2013.

[6] D. Izzo, I. Getzner, D. Hennes, and L. F. Simões. Evolving solutions to TSP variants for active space debris removal. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1207–1214, July 2015.

[7] D. Izzo, D. Hennes, L. F. Simões, and M. Märtens. Designing complex interplanetary trajectories for the global trajectory optimization competitions. *Space Engineering*, pages 151–176, 2016.

[8] D. Hennes and D. Izzo. Interplanetary Trajectory Planning with Monte Carlo Tree Search. In *International Joint Conference on Artificial Intelligence*, pages 769–775, 2015.

[9] L. F. Simões, D. Izzo, E. Haasdijk, and A. E. Eiben. Multi-rendezvous Spacecraft Trajectory Optimization with Beam P-ACO. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 141–156, 2017.

DOI: 10.5281/zenodo.1139250