GTOC9: Methods and Results from the California Polytechnic State University Team

Max Rosenberg[*], Paul Gilles, Adam Frye, Liam Smith

California Polytechnic State University, 1 Grand Avenue, San Luis Obispo, California 93407 (United States)

May 12, 2017

Abstract

The solution methodology created by students of California Polytechnic State University for the 9th Global Trajectory Optimization Competition is presented. Pertinent details of the optimization program are explained, and the program's system architecture is presented both in its current form and in an improved state to guide future work.

1 Introduction

The 9th Global Trajectory Optimization Competition (GTOC 9) tasked entrants to design a mission to place de-orbit packages on 123 simulated pieces of space debris.[2] A list of debris ephemerides and a mission cost calculator were provided to entrants, along with conditions for submitting a valid solution. We will discuss these conditions as they apply to our submissions. A complete enumeration of the conditions, as well as an explanation of the nomenclature and abbreviations used throughout, can be found in the GTOC 9 Prompt.

The mission consists of simulated 'launches' that contribute some initial sunk cost to every sub-mission required to generate a solution, as well as sub-missions to place de-orbit packages on each object. Each sub-mission's contribution to the total cost is evaluated as a function of the fuel required to guide a 2000kg dry-weight probe along a chosen trajectory, plus a number of 30kg de-orbit packages to be placed on each object visited during any given sub-mission. Due to the large number of debris objects, missions consist generally of more than one sub-mission. The disparate orbits of the objects are illustrated visually in Figure 1.

As can be inferred from the figure, the orbits of the debris objects differ from one another significantly in two specific orbital elements: Right-Ascension of the Ascending Node, Ω, and inclination, i. Transferring between orbits differing greatly in these two elements requires maneuvers known

[*]Corresponding author. E-mail: mcrosenb@calpoly.edu

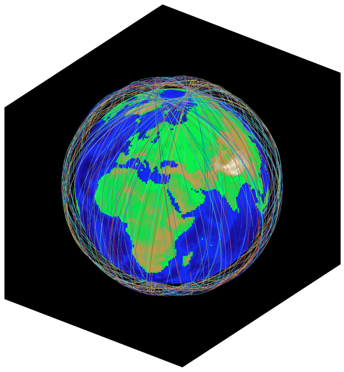

Figure 1: One orbit of each debris object is shown beginning at MJD=20376.6438.

as Node-Line Shifts and Plane Changes. These maneuvers are known to be extremely expensive in terms of fuel, and thus *i* of an Earth-orbiting satellite is normally determined during launch. For the purposes of this competition, a crucial factor would be grouping debris pieces based on the similarity of these parameters associated with each piece.

2 Approach

The first conceptual step taken was to split the problem into two parts. The first part, trajectory optimization, would involve generating optimal transfers between debris pieces during a sub-mission. The degree of optimality, or cost, of the trajectory optimization part would be evaluated by the ΔV required to perform each transfer. The second part, Mission Optimization, would ideally break the list of 123 debris objects into smaller subsets of objects, which could be de-orbited during 1 sub-mission.

The initial work focused on generating sub-missions through the program designed to solve the first part of the problem. This program consisted of three parts:

- A master script, which sets the debris objects of interest to a particular sub-mission, as well as the starting Epoch
- An optimization function and script, which generates the optimal sub-mission trajectory and tabulates the spacecraft state vectors at every impulsive maneuver
- A cost function, which computes the ΔV for each transfer and state vectors at each impulsive maneuver, as output from an algorithm designed to solve Lambert's problem

The actual cost being optimized in this system is the sub-mission cost, not the mission cost. Writing the mission cost optimizer was something that eluded our team due to time constraints. The system architecture of our program is presented in Figure 2.

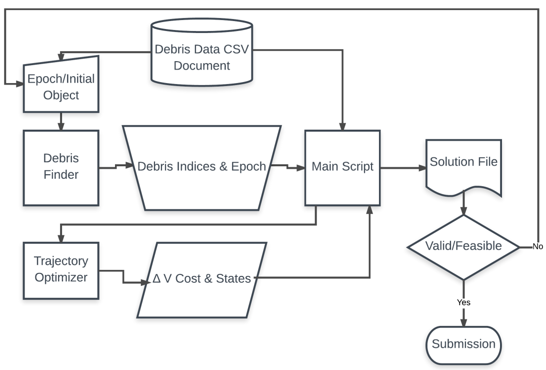

Figure 2: The flow of information through the solution code.

At the core of the optimization was David Eagle's algorithm to solve the J2-Perturbed Lambert Problem.[1]. The use of a solution to Lambert's problem that included the J2 perturbation was imperative to meet the relative distance tolerance required by the contest rules. The use of a fast solver that did not include the perturbation resulted in relative distances in excess of 5 km, while the tolerance was 100 m. This distance frustrated our efforts to validate our own solutions until the root of the discrepancy was discovered.

The J2-Perturbed Lambert Solver was operated on by one of MATLAB's built-in optimization functions, FMINCON. This function searches for a local minimum of the cost function when subjected to constraints. We chose to constrain only the states of the spacecraft at arrival and departure to be the same as the debris piece. Constraining additional parameters to reduce the number of invalid solutions generated would have made implementing a higher level optimizer much easier, but would have slowed the run times of the code significantly, something we could not afford in the final days of the competition. Consequently the final mass was not constrained, and sub-missions required inspection to ensure that they did not violate mass requirements. Likewise the pericenter of the transfer trajectories was not constrained, leading to a large number of solutions marked as invalid because of a close encounter with the Earth. The resulting tool was lean, if not entirely robust. We attempted to generate a tool to implement some sort of genetic algorithm to provide us with an optimal mission profile, as opposed to generating as many optimal trajectories as we could. We were not successful in implementing this step before the competition deadline. The task fell to us to become the mission optimizers ourselves. As reflected in the system architecture shown in Figure 2, the Epoch and Debris IDs for each sub-mission are user inputs to the process. What this meant for us was a frantic scramble to generate valid solutions in the hours leading up to the competition deadline, essentially by hand-picking debris objects for each sub-mission.

3 Next Steps

As mentioned previously, the system we designed to compute solutions to the problem was not fully enclosed. Manual entry of the debris ID and solution start Epoch were still necessary, precluding a truly global solution, or even anything close to that. As we work to improve our methods, a suitable next step would be to refine the system architecture as shown in Figure 3.

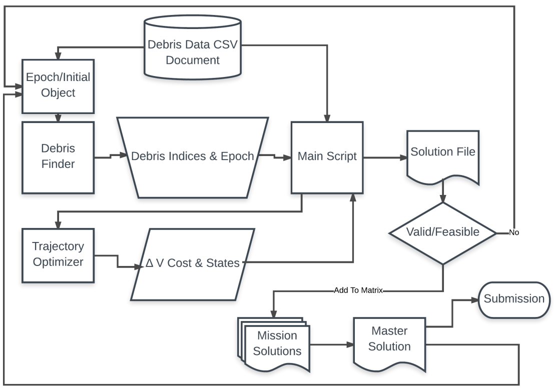

Figure 3: The desired flow of information through the solution code. Note the elimination of the manual step of choosing the Epoch and Initial Object.

As can be seen there are several differences between the current iteration of our software and what we believe to be an effective means of finding a global solution. The most salient of these differences can be seen to the upper left of Figure 3. The block where the Epoch and initial object are selected is changed from a 'manual input' block to a 'process' block. Under this new system, a choice is made based on the validity and quality of a solution generated from the main script and used to refine the initial conditions, rather than relying on the heuristic understanding of the system operators. The sub-missions are eventually curated into a single stack of documents that then form the ultimate submission.
This stage could take the form of a simple loop, or become the cost function inside of a new optimizer. Genetic algorithms are especially suited to problems of this nature. In this case, the list of debris objects to be visited on each sub-mission would constitute a 'chromosome.' We investigated implementing an algorithm of this nature for this purpose, however we were unsuccessful in reaching a valid solution via this method prior to the competition deadline.

4 Conclusion

Cal Poly is honored to have taken part in this competition. The submission of several sub-missions constituting valid solutions to the GTOC 9 problem represents a significant milestone for our team. While the original contribution made to the literature may be small, we feel that we have achieved a strong foothold of understanding how to work problems like the one posed by GTOC 9. This experience will allow us to refine our approach for future competitions, and will serve to grow the body of knowledge at our institution. We see our participation in this competition as a true manifestation of our University's motto, Discere Faciendo.

Acknowledgement

The authors thank Dr. Kira Abercromby for her guidance and inspiration. Her uncompromising demand that each student reach their maximum potential and her tireless efforts to that end motivated the authors to persevere when the problem seemed insurmountable, and she is ultimately responsible for us reaching our goal of finding a solution.

References

[1] David Eagle. Orbital Mechanics with MATLAB: Lambert's Problem, July 2014.

[2] Dario Izzo. *Problem description for the 9th Global Trajectory Optimisation Competition*, May 2017.