Proceedings of GO 2005, pp. 1 – 7.

An Efficient Pruning Technique for the Global Optimisation of Multiple Gravity Assist Trajectories

V.M. Becerra,[1] D.R. Myatt,[1] S.J. Nasuto,[1] J.M. Bishop,[2] and D. Izzo[3]

[1] *Department of Cybernetics, The University of Reading, Reading RG6 6AY, United Kingdom*

[2] *Department of Computing, Goldsmiths College, London SE14 6NW, United Kingdom*

[3] *D. Izzo, Advanced Concepts Team, European Space Agency, Noordwijk, The Netherlands*

Abstract With application to the specific problem of multiple gravity assist trajectory design, a deterministic search space pruning algorithm is developed that displays both polynomial time and space complexity. This is shown empirically to achieve search space reductions of greater than six orders of magnitude, thus reducing significantly the complexity of the subsequent optimisation.

Keywords: mission design, multiple gravity assist, global optimisation, constraint propagation, heuristic search.

1. Introduction

A gravity assist manoeuvre uses a celestial object's gravity in order to change a spacecraft's trajectory. When a spacecraft approaches a celestial object, a small amount of the object's orbital momentum is transferred to the spacecraft. This manoeuvre was used for the first time in the 1970's, when the spacecraft Voyager used multiple gravity assist flybys of Jupiter, Saturn, Uranus and Neptune, to propel itself beyond these planets. Gravity assist manoeuvres (GAs) are frequently used to reduce fuel requirements and mission duration [1]. Most interplanetary trajectory design problems can be stated as optimisation problems, where one of the fundamental goals is the minimisation of fuel requirements, with consideration also given to intermediate planetary flybys, mission duration, type of arrival, launch and arrival windows, and velocity constraints. Traditionally, local optimisation has been used to attempt to solve these design problems [2, 3]. However, because of nonlinearities and the periodic motion of the planets, multiple local minima exist and, as a result, local optimisation only helps to find local minima which are heavily dependent on the initial guesses employed and are not necessarily good solutions. The use of global optimisation techniques has been proposed for tackling these problems, as these methods have better chances of finding good solutions approaching the global optimum [4]. Genetic algorithms and similar techniques have been employed, but these techniques may face difficulties in tackling realistic missions due to the large size of the search space associated with these problems. This paper considers the problem of multiple gravity assist (MGA) trajectories with a known planetary sequence and no deep space manoeuvres. In such cases, it can be shown that the vast majority of the search space consists of infeasible, or very undesirable, solutions. This observation motivated the development of a method for producing reduced search spaces by pruning, thus allowing standard global optimisation techniques to be applied more successfully to the reduced box bounds [5]. The technique presented in this paper has been named Gravity Assist Space Pruning (GASP).

2. Gravity assist: space pruning algorithm

This section describes the motivation behind and functionality of the GASP algorithm. Consider the MGA problem with a defined planetary sequence (e.g. Earth-Venus-Venus-Earth-Jupiter-Saturn) and no deep space maneouvres. The decision vector for this problem is as follows

$$\mathbf{x} = \{t_0, t_1, t_2, t_3 ...\}, \tag{1}$$

where t_0 is the launch date, t_1 is the phase time from the first to second planet, t_2 from the second to third planet etc. An efficient Lambert solver [6] is used to calculate appropriate Keplerian orbits between the planetary positions in the given time, and then a powered swingby model is applied, such as that designed by Gobetz [7].

2.1 Single interplanetary transfer

Consider the simplest case of a single interplanetary transfer with a braking manoeuvre at the target planet. The objective function assumed is a simple minimisation of total thrust (the sum of the initial hyperbolic excess velocity, v_i, and braking manoeuvre, v_f), so

$$f = v_i + v_f. \tag{2}$$

The decision vector in the single transfer case will be $\mathbf{x} = \{t_0, t_1\}$. An important observation is that this search space will contain a line for each time t, that a probe can arrive at the final planet, such that $t_0 + t_1 = t$. Obviously, at a given time t, regardless of the launch time or departure time, the target planet will be *in the same position and have the same velocity*. Therefore, it is beneficial to consider the search space as t_0, $t_0 + t_1$, this is departure time at the first planet compared to arrival time at the second. The optimisation method to be investigated is grid sampling. Grid sampling is usually considered a very inefficient optimiser, particularly in high dimensionalities. For example, using the enumerative search in the Swingby Calculator application [8] yields optimisation times approaching an hour for relatively small search spaces (on a 600Mhz Pentium Processor). However, for only 1 and 2 dimensions grid sampling is computationally tractable, as long as the objective function is reasonably smooth and the exact optimum is not required. Therefore, the objective function for a single interplanetary transfer may be grid sampled at an appropriate resolution in the departure time vs arrival time domain efficiently, although in this case most other optimisation methods would yield better results in terms of objective function evaluations. However, the grid sampled version will require many less Ephemeris calculations, as the same positions/velocities need not be recalculated for a given departure or arrival time. If the 2D search space was discretised into k cells in each dimension, only $2k$ Ephemeris calculations are required for the entire sampling, and k^2 Lambert problem solutions. By comparison, two Ephemeris calculations would be required by each objective function evaluation in a standard optimiser. Even in the single interplanetary transfer case, a large proportion of the search space corresponds to undesirable solutions i.e. those with impractical C3[1]. To illustrate this, the optimisation of an Earth-Mars transfer was considered between the dates (-1200 to 600 MJD2000) and phase times of 25 to 515 days. A sampling resolution of 10 days was used in both axes. Only 12.5% of this search space had a C3 of less than 25km^2/s^2. Figure 1 shows this search space plotted as departure time vs arrival time - the diagonal lines delineate the sampled portion of the search space, and the dark regions within the lines indicate trajectories with a fasible C3 value lower than 25km^2/s^2.

As a consequence, in gravity assist and multiple gravity assist cases starting with an Earth-Mars transfer in these bounds, at least 87.5% (100%-12.5%) of the overall search space *must* correspond to undesirable solutions. Even allowing an enormous C3 of 100km^2/s^2, only 33% of the search space becomes valid. The GASP algorithm was design to efficiently detect and prune infeasible parts of the space, leaving several sets of box bounds with vastly smaller

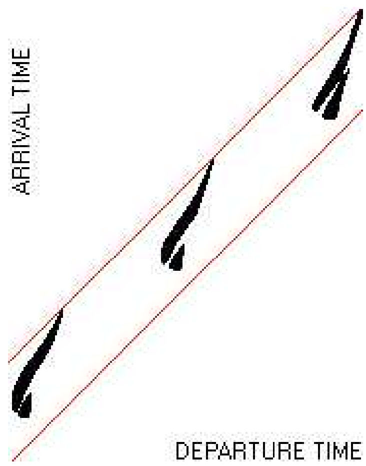

Figure 1. A grid sampled Earth-Mars transfer. The white regions within the delineating diagonal lines indicate solutions with a C3 of greater than $25\text{km}^2/\text{s}^2$

contents. These reduced box bounds may then be optimised efficiently using a standard optimisation method.

Hyperbolic Excess Velocity Constraint: The maximum allowable hyperbolic excess velocity is the first main constraint of the GASP algorithm, as it determines possible launch dates to the first target planet.

Braking Manoeuvre Constraint: As well as the C3 constraint, it is logical to add a constraint on the maximum braking manoeuvre that the spacecraft can perform. Applying a C3 constraint of $25\text{km}^2/\text{s}^2$ and a braking manoeuvre constraint of 5km/s it can be estimated that less than 5% of the search space yields feasible trajectories. By applying two very simple constraints to the interplanetary case it has been shown that a very significant reduction in search space can been achieved, leaving clear launch windows and arrival time windows.

2.2 Forward Constraining

It has been shown that the C3 and braking manoeuvre constraints alone significantly reduce the search space content for an interplanetary transfer. From Figure **??**, it can be seen that for many values of the arrival time there are no feasible departure times. This observation is the key principle on which the GASP algorithm is based: if no feasible trajectories arrive at a planet on a given date then there can be no departures from the planet on that date (assuming the change in velocity from the swingby is instantaneous). Now consider a trajectory with a single gravity assist. Using grid sampling on this function would usually involve sampling in three dimensions, and hence as additional planets were added the number of objective function evaluations would increase exponentially. Instead, with GASP, the search space is sampled as a cascade of two dimensional search spaces, each with possible departure dates (in the horizontal axis) and prospective arrival dates (in the vertical axis). Because of this, the number of Lambert problem evaluations is vastly reduced.

2.3 Gravity assist thrust constraint

Two constraints are added in order to maximise the probability of gravity assists being feasible. The first such constraint is the gravity assist thrust constraint, which limits the maximum absolute difference between incoming and outgoing velocities during a gravity assist to some threshold, T_v. This threshold is set separately for each gravity assist. The following is then performed for *every* arrival time at a planet:

1 Calculate the bounds on incoming velocity, v^i_{min} and v^i_{max}.

2 Invalidate any outgoing trajectories that do not have outgoing velocities in the range $[v^i_{\text{min}} - T_v - L_v, v^i_{\text{max}} + T_v + L_v]$, where L_v is an appropriate tolerance based on the Lipschitzian constant of the current phase plot.

3 Calculate the modified bounds on outgoing velocity, v^f_{min} and v^f_{max}.

4 Invalidate any incoming trajectories with velocities outside the range $[v^f_{\text{min}} - T_v - L_v, v^f_{\text{max}} + T_v + L_v]$.

2.4 Gravity assist angular constraint

The gravity assist angular constraint removes infeasible swingbys from the search space on the basis of them being associated with a hyperbolic periapse under the minimum safe distance for the given gravity assist body. This is determined over every arrival date at a planet as follows, assuming i valid incoming trajectories and j valid outgoing trajectories:

1. For all i incoming trajectories
2. For all j incoming trajectories
3. If the swingby is valid for the current incoming and outgoing trajectory, mark both incoming and outgoing trajectory as valid.
4. End
5. End
6. Invalidate all trajectories not marked as valid

The swingby angle is decreased by an appropriate Lipschitzian tolerance θ_L, in order to compensate for the effects of the grid sampling of the search space.

3. Time and space complexity

This section determines the time and space complexity of the GASP algorithm. It will be shown that GASP scales quadratically in space and quartically in time with respect to the number of gravity assist manoeuvres considered. For simplicity, the following analysis assumed that the initial launch window and *all* phase times are the same.

3.1 Space Complexity

Consider a launch window discretised into k bins and a mission phase time also discretised into k bins. For the first phase k^2 Lambert problems must be sampled. The next phase will need to sample $(k + k)k = 2k^2$, as the number of possible times that the planet may be arrived at is doubled (minimum launch date, minimum phase time to maximum launch date, maximum phase time). The third phase will require $3k^2$ Lambert function evaluations, and the n^{th} phase nk^2. This gives the series

$$O(n) = k^2 + 2k^2 + 3k^2 + \ldots + nk^2 \tag{3}$$

$$O(n) = k^2(1 + 2 + 3 + \ldots + n) \tag{4}$$

$$O(n) = k^2 \frac{n(1 + n)}{2}. \tag{5}$$

Therefore, the amount of space required for n phases is only of the order $O(n^2)$, rather than $O(k^n)$ for full grid sampling.

Similarly, it is clear that the space complexity with respect to the resolution k, is also of the order $O(k^2)$.

3.2 Time Complexity

The memory space required is directly proportional to the maximum number of Lambert problems that must be solved, and hence the time complexity of the sampling portion of the GASP algorithm must also be of the order $O(n^2)$.

Launch energy constraint complexity: The launch energy constraint is only applied in the first phase, and hence is independent of the number of swingbys. The time complexity is $O(k^2)$ with respect to resolution.

Gravity assist thrust constraint complexity: The time complexity of applying the gravity assist thrust constraint is $O(n^2)$ with respect to dimensionality (number of phases), due to the inevitable increase in size of later phase plots to encompass all possible arrival dates. The first phase requires of the order of $2k \times (k + 3k)$ operations in order to perform the constraining of outgoing velocity from incoming velocity (the back constraining may be ignored at this point). The second phase requires of the order of $3k \times (2k + 4k)$ operations. In general, the n^{th} phase requires of the order of $2n^2k^3$ operations. Therefore, the total number of operations over all phases is

$$2k^2[2^2 + 3^2 + 4^2 + \ldots + n^2] = 2k^2\frac{n(n+1)(2n+1)}{3} \tag{6}$$

Therefore, applying this constraint yields cubic time complexity in dimensionality and quadratic complexity in resolution.

Gravity assist angular constraint complexity: The maximum number of swingby models that must be calculated for the first phase is close to $k \times 2k \times 3k = 6k^3$. For the second swingby, this is $2k \times 3k \times 4k = 24k^3$. In general, for n phases, the upper bound on the number of swingby calculations, α, is

$$\begin{aligned}\alpha = \;& 3 \times 2 \times 1 \times k^3 + 4 \times 3 \times 2 \times k^3 \\ & +5 \times 4 \times 3 \times k^3 + \ldots + (n+2)(n+1)nk^3\end{aligned} \tag{7}$$

From [9], it can be shown that the total number of these operations must be

$$\alpha = k^3 \sum_{j=1}^{n} (j+2)(j+1)j = k^3\frac{n(n+1)(n+2)(n+3)}{4}. \tag{8}$$

Therefore, the overall time complexity with respect to resolution is $O(k^3)$, while the time complexity with respect to dimensionality is $O(n^4)$. Therefore, the gravity assist angular constraint is the most computationally expensive and hence is applied after GA thrust constraint in order to minimise the number of swingby models that must be calculated.

Overall time complexity: The overall time complexity, taken from the most complex part of the algorithm (the gravity assist angular constraint), is cubic with respect to resolution and quartic with respect to dimensionality.

4. Differential evolution

Differential Evolution (DE) [10] is a novel incomplete probabilistic global optimiser based on Genetic Algorithms [11], and was the highest ranked GA-type algorithm in the First International Contest on Evolutionary Computation. Following [10], scheme DE1 is used in this work as the crossover operator as it has been shown to perform the best on the most complex test function examined.

5. Results

This section demonstrates the improvements that GASP can make over Differential Evolution alone in one test case. Consider the optimisation of an EVVEJS transfer with an orbital insertion, where the objective function is the minimisation of the sum of the launcher and probe thrust. The bounds on the decision vector were as follows:

- $t_0 \in [-1200, 600]$ MJD2000
- $t_1 \in [14, 284]$ days
- $t_2 \in [22, 442]$ days
- $t_3 \in [14, 284]$ days
- $t_4 \in [99, 1989]$ days
- $t_5 \in [366, 7316]$ days

GASP was applied to this problem with a sampling resolution of 10 days. In order to complete the sampling, 144498 Lambert problem solutions and 3749 Ephemeris calculations were required. The following constraints were defined in the GASP algorithm:

- $T_{HEV} = 8000 m/s$
- $T_{GA_{1,\ldots,4}} = 1000 m/s$
- $T_{\text{Brake}} = 5000 m/s$

This configuration yields two major solution families, one with a launch window of -920 to -660MJD2000, and the other 280 to 490MJD2000. Differential Evolution was applied to the accumulation of each solution family (the tightest decision vector bounds that all solution family nodes exist within). A population of 40 individuals was used and a terminal number of 2000 iterations were allowed. Note that this corresponds to $40 \times 2000 \times 5 = 400000$ Lambert problem solutions and 480000 Ephemeris calculations. The later launch window was eliminated immediately as applying Differential Evolution did not yield any valid solutions. Further optimisation on this launch window has consistently optimised to the same invalid minima. The earlier launch window proved much more promising and, as a consequence, 20 optimisation trials were performed. Of these trials, 19 found the second best known optima to this problem (5225m/s) to within 1m/s, and one came close to the best known optimum, (4870m/s). Experimentation has shown this minimum has a very small basin of attraction with respect to this objective function, and is exceptionally hard to find even in the reduced search space. When applying Differential Evolution alone to the entire search space, only 7 out of the 20 trials found the second best minimum, and none the best known. Using GASP, it is apparent that there is an extremely high probability that at least the second best solution will be found. In subsequent trials, the objective function was altered to penalise any trajectory with hyperbolic excess velocity of greater than 3000m/s. Only two of 20 of the GASP constrained trials failed to find the best known solution to within 10m/s in this case (instead finding the second best one), while again 7 out of 20 optimisation of the entire domain found the second best solution, and none located the best. Again, these results highlight the significant advantages of using the GASP algorithm. Not only does it allow effective visualisation of the search space, but it drastically reduces the requirement for optimisation restarts in order to find good solutions, and at a fraction of the computational expense of an optimisation restart.

6. Conclusions

This paper has described the Gravity Assist Space Pruning algorithm, proved that it has both polynomial time and space complexity, and furthermore demonstrated that it produces significant benefits over optimising the entire domain with relatively little computational expense. Additionally, the GASP algorithm allows intuitive visualisation of a high dimensional search space, and facilitates the identification of launch windows and alternative mission options.

7. Acknowledgments

This work has been funded by the European Space Agency under Contract 18138/04/NL/MV, Project Ariadna 03/4101.

Notes

1. C3 (units km^2/s^2) is the square of the hyperbolic excess velocity.

References

[1] Debban, T.J., McConaghy, T.T., Longuski, J.M., "Design and Optimization of Low-Thrust Gravity-Assist Trajectories to Selected Planets." AIAA 2002-4729, Astrodynamic Specialist Conference and Exhibit, 2002.

[2] Hargraves, C., and Paris, S., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation", Journal of Guidance Control and Dynamic Vol 10, pp. 338-342, 1987.

[3] Betts, J.T., "Practical Methods for Optimal Control Using Nonlinear Programming", Philadelphia: SIAM, 2001.

[4] Vasile, M., "Systematic-Heuristic Approach for Space Trajectory Design" Annals of the New York Academy of Science, Vol 1017, pp. 234-254, 2004.

[5] Myatt, D.R., Becerra, V.M., Nasuto, S.J., and Bishop, J.M., "Advanced Global Optimisation for Mission Analysis and Design." Final Report. Ariadna id: 03/4101. Contract Number: 18138/04/NL/MV, 2004. Available: http://www.esa.int/gsp/ACT/doc/ACT-RPT-ARIADNA-03-4101-Reading.pdf.

[6] Izzo, D., "Efficient Computation of Lambert Interplanetary Arcs", internal report ACT-RPT-4100-DI-ECLI01

[7] Gobetz, F.B., "Optimal transfer between hyperbolic asymptotes", AIAA Journal Vol. 1, No. 9, 1963.

[8] Biesbroek, R., Ancarola, B. "Optimisation of Launcher Performance and Interplanetary Trajectories for Pre-Assessment Studies", IAF abstracts, 34th COSPAR Scientific Assembly, The Second World Space Congress, Houston, TX, USA., p.A-6-07, 10-19 October, 2002.

[9] Riordan, J., Combinatorial Identities, Wiley, 1968.

[10] Storn, R. and Price, K., "Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization, Vol. 11, pp.341-359, 1997.

[11] Holland, J.H., Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.