

Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering

<http://pig.sagepub.com/>

Low-thrust trajectory design as a constrained global optimization problem

C H Yam, D D Lorenzo and D Izzo

Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering published online 10

August 2011

DOI: 10.1177/0954410011401686

The online version of this article can be found at:

<http://pig.sagepub.com/content/early/2011/08/09/0954410011401686>

Published by:



<http://www.sagepublications.com>

On behalf of:



[Institution of Mechanical Engineers](http://www.imeche.org)

Additional services and information for *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* can be found at:

Email Alerts: <http://pig.sagepub.com/cgi/alerts>

Subscriptions: <http://pig.sagepub.com/subscriptions>

Reprints: <http://www.sagepub.com/journalsReprints.nav>

Permissions: <http://www.sagepub.com/journalsPermissions.nav>

Low-thrust trajectory design as a constrained global optimization problem

C H Yam^{1*}, D D Lorenzo², and D Izzo¹

¹Advanced Concepts Team, European Space Agency, Noordwijk, The Netherlands

²Global Optimization Laboratory, University of Florence, Florence, Italy

The manuscript was received on 29 October 2010 and was accepted after revision for publication on 3 February 2011.

DOI: 10.1177/0954410011401686

Abstract: The design of a spacecraft trajectory can be formulated as a global optimization task. The complexity of the resulting problem depends greatly on the final target planet, the chosen spacecraft intermediate route, and the type of engine and power system available on-board. Few attempts have been made to directly use a global optimization framework to design trajectories that make use of low-thrust propulsion because of the large scale and extreme complexity of the resulting non-linear programming problem. The presence of non-convex constraints, in particular, requires the use of solvers able to deal with such an added complexity. Here, the Sims–Flanagan transcription method is proposed to model the low-thrust trajectory design as a constrained global optimization problem. Then, two different solvers are applied: basin hopping and simulated annealing with adaptive neighbourhood. Both algorithms are hybridized with a local search. Two different interplanetary trajectories are considered: an Earth–Earth–Jupiter transfer with a nuclear electric propulsion spacecraft inspired by the Jupiter Icy Moons Orbiter and a transfer to Mercury inspired by the BepiColombo mission. For both problems, the proposed approach proves to be able to explore automatically the vast solution space, producing a large number of trajectories in a large range of final mass and flight times, proving the possibility to apply global optimization techniques directly to the low-thrust problem.

Keywords: global optimization, low-thrust trajectory, basin hopping

1 INTRODUCTION

The application of global optimization techniques to the design of interplanetary trajectories has received quite some attention in the past years as it becomes increasingly evident that such a framework introduces a high level of automation in a process that is otherwise still heavily relying on expert engineering knowledge. The systematic study of global

optimization algorithms in relation to chemically propelled spacecrafts [1–6] has proved that efficient computer algorithms are able to produce, for these types of spacecrafts, competitive trajectory designs. Thanks to initiatives such as the Global Trajectory Optimization Competition [7] and the Global Trajectory Problem database [8] of the European Space Agency, the attention of communities not traditionally linked to aerospace engineering research [9–11] has increased bringing a beneficial influx of new ideas and solutions, thus advancing the field considerably. While for problem formalizations such as the multiple gravity assist (MGA) and the multiple gravity assist with deep space manoeuvre

*Corresponding author: Advanced Concepts Team, European Space Agency, DG-PF, ESTEC, Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands.
email: chithongyam@gmail.com

(MGA-DSM) [4], the advantages of using these techniques has been proved, no convincing results have been produced so far [12] in the case of the low-thrust multiple gravity assist (LT-MGA) problem. The optimization problem of simple low-thrust trajectories can be solved efficiently by local optimization methods. However, on a large design space, local methods converge to suboptimal solutions or sometimes fail to converge if a good starting guess is not provided. On the other hand, global methods fail to provide a good solution because of the complexity of the resulting non-linear programming (NLP) problem that, unlike the box-constrained MGA and MGA-DSM, has to deal with a high number of non-linear constraints if an accurate spacecraft dynamics has to be accounted for. Constraints in the optimization problem can be handled as an extra penalty term in the objective function [13]. However a suitable value of the weighting factor on the penalty is unknown beforehand. A bad choice on the weighting factor leads to premature convergence on the objective or to infeasible solutions. Some recent works [13, 14] are built upon and a global optimization framework is presented for the LT-MGA problem where non-linear constraints handling is incorporated in the main global optimization loop *via* the algorithm hybridization with a local method. The resulting algorithm explore efficiently the vast solution space of low-thrust trajectories, thus producing a convincing case for using global optimization techniques also in relation to the LT-MGA problem.

2 TRAJECTORY MODEL

The trajectory model that is to be used to transcribe an LT-MGA trajectory optimization into a NLP problem (to be solved by global optimization methods) is crucial to the success of the overall algorithm one wants to produce. Criteria to be accounted for include accuracy in the description of the spacecraft dynamics, computational efficiency in the objective function and constraints evaluation, problem dimension, and number of non-linear constraints produced. Bearing these issues in mind, in this article, a version of the trajectory model proposed by Sims and Flanagan [15] is to be used. Figure 1 briefly illustrates such a trajectory model. Trajectory is divided into legs which begin and end with a planet. Low-thrust arcs on each leg are modelled as sequences of impulsive manoeuvres ΔV_i , connected by conic arcs. The number of impulses (which is the same as the number of segments) is denoted by N . The ΔV_i at each segment should not exceed a maximum magnitude, ΔV_{\max} , where ΔV_{\max} is the velocity change

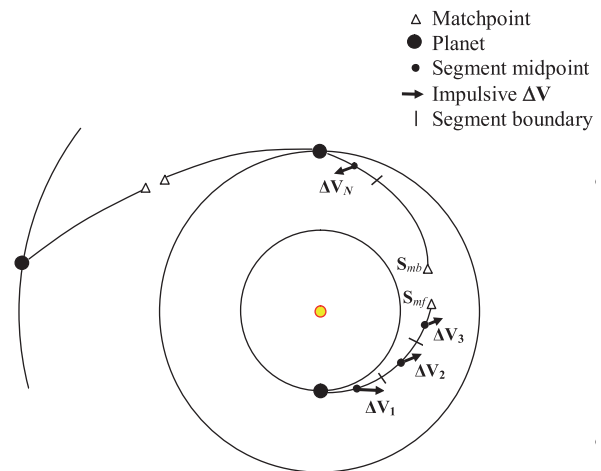


Fig. 1 Impulsive ΔV transcription of a low-thrust trajectory, after Sims and Flanagan [15]

accumulated by the spacecraft when it is operated at full thrust during that segment

$$\Delta V_{\max} = (T_{\max}/m)(t_f - t_0)/N \quad (1)$$

where T_{\max} is the maximum thrust of the low-thrust engine, m the mass of the spacecraft, and t_0 and t_f the initial and final time of a leg. The spacecraft mass is propagated using the rocket equation [16]

$$m_{i+1} = m_i \exp(-\Delta V_i/g_0 I_{sp}) \quad (2)$$

where the subscript i denotes the mass and ΔV on the i th segment, g_0 the standard gravity (9.80665 m/s^2), and I_{sp} the specific impulse of the low-thrust engine.

At each leg, trajectory is propagated (with a two-body model) forwards and backwards to a matchpoint (usually halfway through a leg), where the spacecraft state vector becomes $\mathbf{S}_{mf} = \{r_x, r_y, r_z, v_x, v_y, v_z, m\}_{mf}$ (and similarly for \mathbf{S}_{mb}), where r and v are, respectively, the position and velocity of the spacecraft and the subscripts represent, the Cartesian x, y, z components. The forward- and backward-propagated half-legs should meet at the matchpoint, or the mismatch in position, velocity, and mass

$$\mathbf{S}_{mf} - \mathbf{S}_{mb} = \{\Delta r_x, \Delta r_y, \Delta r_z, \Delta v_x, \Delta v_y, \Delta v_z, \Delta m\} \quad (3)$$

should be less than a tolerance in order to have a feasible trajectory. The patched-conic assumption is employed for gravity-assist trajectories, in which the spacecraft velocity is changed instantaneously by the planet's gravity during a flyby. The angle between the incoming and outgoing \mathbf{V}_{∞} , or the flyby turn angle δ , is given by

$$\sin(\delta/2) = 1/(1 + r_p V_{\infty}^2/\mu) \quad (4)$$

where r_p is the flyby periapsis radius and μ the gravitational parameter of the gravity-assist body.

3 OPTIMIZATION PROBLEM

3.1 The objective and constraints

Under the trajectory model proposed in this article, the optimization of low-thrust trajectories is formulated as an NLP problem, where the objective is to maximize the final spacecraft mass (m_f)

$$\max m_f \quad (5)$$

subject to:

- the equality constraints on the state mismatch in equation (3);
- the inequality constraints that $\Delta V_i \leq \Delta V_{\max}$ (equation (1));
- the inequality constraints that the angle between the incoming and outgoing \mathbf{V}_∞ vectors is less than the maximum turn angle given by equation (4).

Note that the constraints on the ΔV magnitude can be considered as linear constraints, non-linear constraints, or simply bounds on the decision vector variables according to the exact choice of the decision vector encoding.

3.2 The decision vector

The decision vector used contains the following variables:

- the departure epoch t_0 ;
- the departure velocity relative to the Earth V_∞ ;
- the departure mass m_0 ;
- for each leg j and each segment i , the impulse intensity and direction ΔV_{ij} ;
- for each swingby, the incoming and outgoing velocities relative to the planet;
- for each swingby j , the swingby epoch t_j ;
- the spacecraft mass at each swingby m_j ;
- the arrival epoch t_f ;
- the arrival mass m_f .

All the velocities and impulses are expressed with three Cartesian coordinates in the experiments. Since a rendezvous problem is being considered, the arrival velocity to the destination is not included in the set of variables, as it is constrained to be zero relative to the planet.

The resulting NLP will have a dimension equal (or smaller depending on the mission type) to $(10 + 3N)M$, where M is the number of legs considered, and a number of non-linear constraints equal

(or greater according to the decision vector encoding chosen and to other mission details) to $n_{\text{eq}}M$, where n_{eq} is the number of equations considered on the dynamics (e.g. $n_{\text{eq}} = 7$ if a three-dimensional problem with mass is considered).

4 GLOBAL OPTIMIZATION ALGORITHMS

The proposed transcription of the LT-MGA problem is a continuous, constrained, non-linear optimization problem. Since such class of problems usually present local minimizers that are not global, they are often unsolvable using only local optimization algorithms. Practical experience shows that this is usually the case with trajectory optimization problems, regardless of the propulsion type. Thus, local solvers must be used inside a global optimization strategy in order to achieve solutions which are as close as possible to the optimal ones. Here, three different approaches that have been tried on such problem are described. Some procedures that the algorithms will use is first defined.

- $\mathcal{G}()$ is a procedure that randomly generates a starting point. Ideally, the point is to be uniformly distributed on the feasible region, but since the problem's feasible set is of a very small size, and generating a point inside such region is a hard problem by itself, a procedure that uniformly generates points into a reasonable box is used. Containing the feasible region.
- $S(x)$ is a procedure that, given a point x , computes a local minimizer of the objective function, taking x as an initial guess. A software package called SNOPT [17, 18] is for the tests, which implements sequential quadratic programming.
- $Best(x, y)$ is a procedure that, given two solutions x and y , returns the best one according to a fixed rule. Since a constrained optimization problem is being dealt with problem, $Best(x, y)$ chooses the point with the lower constraint violation norm value. In case both points are feasible, then the point with the lower objective function value is chosen instead.

The first and most simple algorithm is called Multistart (MS), which directly optimizes the points obtained by a generator. Multistart can be described by the following pseudo code.

- let $x^* := \mathcal{G}()$
- for $i = 1, \dots, N$
- let $x := \mathcal{G}()$
- let $y^* := S(x)$
- let $x^* := Best(x^*, y^*)$
- end for

Although for N big enough and reasonable choices of \mathcal{G} , \mathcal{S} , and $Best$, MS will eventually converge to the global solution, the extremely slow convergence rate renders this algorithm unfit for the vast majority of global optimization problems. However, because of its simplicity, this algorithm can be effectively used as a baseline to compare other solvers to.

The second algorithm is called Monotonic Basin Hopping (BH), or Iterated Local Search [19]. In addition to the procedures previously used by MS, Monotonic BH needs a procedure $\mathcal{P}(x)$ that, given a point x returns another point randomly generated in a conveniently defined neighbourhood of x . Such algorithm can then be described as follows.

1. let $x_{best} := \mathcal{G}()$
2. for $i = 1, \dots, N$
3. let $x := \mathcal{G}()$
4. let $x^* := \mathcal{S}(x)$
5. let $k := 0$
6. while $(k < MNI)$ do:
7. let $y := \mathcal{P}(x^*)$
8. let $y^* := \mathcal{S}(y)$
9. if $Best(y^*, x^*) = y^*$ then
10. let $x^* := y^*$
11. let $k := 0$
12. else
13. let $k := k + 1$
14. end if
15. end while
16. let $x_{best} := Best(x_{best}, x^*)$
17. end for

In practice, after a local optimization, instead of generating a new point inside the whole feasible region like is done with MS, the generation is restricted inside a small region centre on the current best point. If, as it happens with real-life problems, the good solutions are clustered together, there is a good chance that the series of perturbations and reoptimizations will lead to the best solution contained inside the cluster [20–23]. A new point is then generated from the whole feasible set only when no improvement has been made for a number of times equal to a fixed parameter called Max No Improve (MNI), which has been set to 500 during the runs. The choice of the perturbation function usually determines the BH performance. A typical rule is to choose a new point inside a small box or sphere centred on the current point, but problem knowledge, if available, can be used to better tune the perturbation. As an example, when only two different planets are considered, it can be expected that by shifting a solution in time by a period equal to the synodic period of such two planets, solutions

that are similar (and maybe better) than the current one could be found. Hence, the perturbation rule that has been used is composed by the following two steps.

1. For each variable x_i and its lower and upper bounds l_i and u_i , add to x_i a value uniformly chosen in the interval $[-r(u_i - l_i), r(u_i - l_i)]$ with a small given value of r .
2. With a low probability p , shift the solution in time, either forwards or backwards with equal probability, by a time length equal to the synodic period.

In the experiments, $r = 0.05$ and $p = 0.1$ is used.

Finally, the simulated annealing (SA) with adaptive neighborhood has been used. The algorithm structure is similar to Monotonic BH, with some important differences in key parts of the algorithm, which are quickly described here. For a more complete description, refer to [24, 25].

First of all, the comparison in line 9 is modified to allow for non-monotonicity. The problem is transformed to an unconstrained optimization problem using a penalty function to account for constraints violations. Then, $Best(x, y)$ returns x with probability 1 if $f(x) \leq f(y)$, or with probability $e^{(f(y) - f(x))/T}$ if $f(x) > f(y)$. T is the temperature parameter, which is exponentially decreased every fixed number of iterations.

Furthermore, the perturbation procedure \mathcal{P} is adaptive, meaning that the radius of the perturbation is adjusted at every iteration. Such r value is increased each time the generated point is accepted, and decreased each time the point is refused.

Finally, the local optimization \mathcal{S} is not executed at each step of the algorithm, but just once at the end, starting from the point returned by the SA with Adaptive Neighborhood.

In the tests, the temperature parameter T starts at 10 and is multiplied by 0.8 every 100 iterations. The starting perturbation radius r^0 is equal for each variable to 0.05 times the width of the box, and every K iterations a new value r^{k+1} is obtained from r^k using the following rule. Let ρ be the number of accepted steps in the last K iterations, divided by K . Then

$$r^{k+1} = \begin{cases} r^k \left(1 + \frac{\rho - 0.6}{2}\right) & \text{if } \rho > 0.6 \\ \frac{r^k}{1 + \frac{0.4 - \rho}{2}} & \text{if } \rho < 0.4 \\ r^k & \text{otherwise} \end{cases}$$

$K = 10$ is used, and the maximum number of iterations was set at 2000.

All the code was written in C++, and compiled with gcc on a GNU/Linux environment.

5 NUMERICAL RESULTS

5.1 Nuclear electric propulsion mission to Jupiter

The application of the global optimization framework is demonstrated to perform preliminary design of trajectories for a mission that employs nuclear electric propulsion (Table 1). The spacecraft is assumed to have a thruster with a constant maximum thrust and a constant specific impulse, with similar hardware parameters to the Jupiter Icy Moons Orbiter [26–30] (a cancelled mission originally proposed by NASA in 2003). In the example scenario, a planetary encounter sequences of Earth–Earth–Jupiter (E–E–J) (i.e. one Earth flyby) which rendezvous at Jupiter is considered.

The approach of this study begins with the three global optimization algorithms solving an Earth–Earth–Jupiter rendezvous problem which maximizes the final mass (equation (5)). Ten segments ($N=10$) are used for each leg and the dimension of this problem is 75 with 35 non-linear constraints (as Cartesian coordinates were chosen to encode the ΔV constraint on their magnitude is quadratic and thus increase the number of non-linear constraints). First, the global optimizers are let to run for a fixed time (~ 5 h for each algorithm) on a PC (AMD Turion@2.1 GHz with 3GB of RAM) which produces hundreds of trajectories. Then, feasible solutions that have a norm... vector of less... 10^{-6} is selected than 10^{-6} .

Table 1 Parameters for a nuclear electric propulsion mission

Parameters	Values
Initial mass of the spacecraft	20 000 kg
Maximum thrust	2.26 N
Specific impulse	6000 s
Launch date	01/01/2020 to 01/01/2030
Launch V_∞	≤ 2.0 km/s
Maximum time of flight	10 years
Minimum flyby radius	7000 km

Table 2 summarizes the results found by the three global optimizers. It is noticed that many solutions found by SA and MS fail to converge and therefore, the number of solutions is less than BH. Figure 2 plots the comparison of the three algorithms, which shows that results found by BH always have higher final mass than the other two methods, while SA and MS have similar performance.

Figure 3 plots the x - y projection of a trajectory found by BH with the highest final mass. On the plot, the solid and dashed curves represent thrusting and coasting segments, respectively; while a ΔV is shown as an arrow in the midpoint of a segment. In this example, the spacecraft leaves the Earth on 12 November 2021 with a V_∞ of 2 km/s. It enters a 4:3 resonance orbit with a period of ~ 1.3 years and goes around the Sun for three revolutions, before it flybys the Earth after 3.8 years with an increased V_∞ of 8.5 km/s. After the gravity assist at the Earth, its aphelion increases for the transfer to Jupiter. After 7.4 years of interplanetary flight, the spacecraft rendezvous at Jupiter on 24 March 2029 with a final mass of 17 102 kg.

Besides the value of the objective function (final mass), it is also interesting from a mission design point of view that the optimization process is able to find trajectories that launch on different dates. In the example in Fig. 4, the difference in the final mass is less than 200 kg (or 1 per cent of the initial mass) for most launch periods. The 1 per cent penalty of the final mass gives flexibility to the mission designer to choose a different date in case there is a change in the mission. The process is also able to locate various locally optimal trajectory families, which is interesting from an astrodynamics point of view. From Fig. 5, It is noted that the ‘clusters’ of solutions belongs to different Earth–Earth resonance transfer orbit. For example, 1:1 resonance with flight time ~ 400 days, 2:3 resonance with flight time ~ 800 days, and 3:2 resonance with flight time ~ 1100 days. Unlike the case in the ballistic transfer, in the low-thrust transfer case, the Earth–Earth flight time does not exactly equal some integer multiples of Earth’s

Table 2 Algorithm statistics for the E–E–J mission

Algorithm	Basin hopping	Simulated annealing	Multistart
Best (kg)	17 102	17 019	16 961
Worst (kg)	10 913	11 924	11 900
Mean (kg)	16 235	16 302	15 930
Mean best 10 (kg)	17 039	16 912	16 715
Standard deviation (kg)	1345	1160	1320
Number of solutions above 95% best (16 246 kg)	137	19	11
Number of solutions above 90% best (15 391 kg)	155	22	14
Total number of solutions	183	24	18

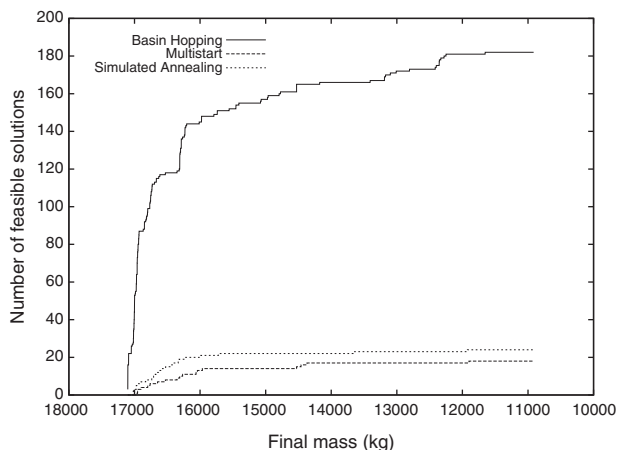


Fig. 2 Cumulative number of solutions for the E-E-J mission

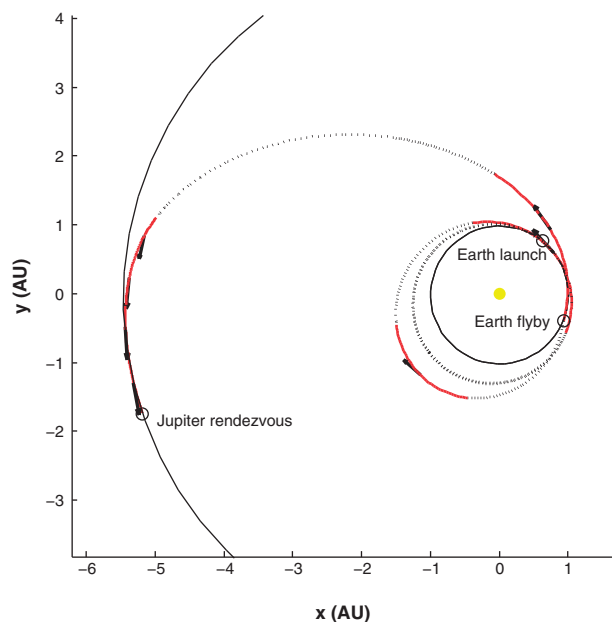


Fig. 3 Trajectory plot of an E-E-J rendezvous mission

orbital period and the spacecraft does not encounter the Earth at the same position from launch. However, the mechanism in the low-thrust case is similar to the chemical case [31], where the V_∞ at the second Earth encounter is increased through some small manoeuvres.

5.2 Mission to mercury

The second test case is a mission to Mercury, inspired from the BepiColombo mission [32]. The planetary encounter sequence is Earth–Venus–Venus–Mercury–Mercury–Mercury (EVMMM) and the mission parameters are presented in Table 3. The mission parameters have been subject to many

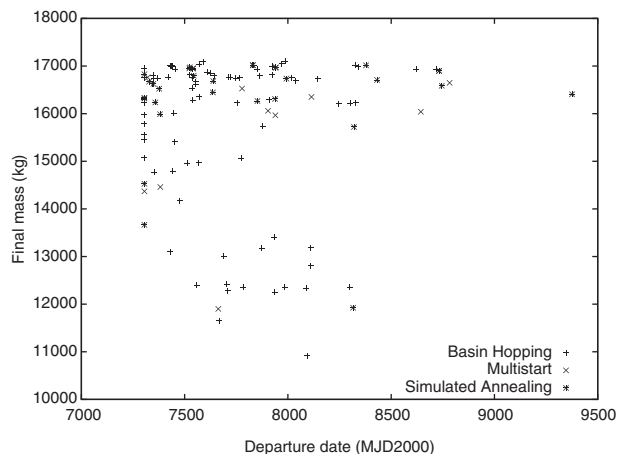


Fig. 4 E-E-J solutions with various launch dates

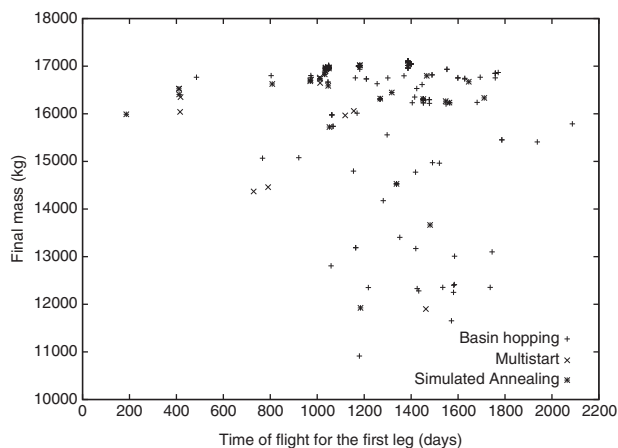


Fig. 5 E-E-J solutions with various Earth–Earth transfer times

Table 3 Parameters for a mission to Mercury

Parameters	Values
Initial mass of the spacecraft	1300 kg
Maximum thrust	0.34 N
Specific impulse	3200 s
Launch date	01/08/2009 to 27/04/2012
Launch V_∞	≤ 1.925 km/s
Arrival V_∞	≤ 0.5017 km/s
Arrival date	No later than 26/11/2021
Minimum flyby radius	$1.1 R_{\text{planet}}$

changes since the original concept. Here, the mission as was described in a 2002 report from the European Space Operations Centre [33] is considered. Unlike the solar electric propulsion (SEP) used in the BepiColombo mission, the SEP engine is not modelled in the test but rather assume the thrust and specific impulse to be constant. In comparison with the

Table 4 Algorithm statistics for a mission to Mercury

Algorithm	Basin hopping	Simulated annealing	Multistart
Best (kg)	1064	988	1052
Worst (kg)	101	111	100
Mean (kg)	724	522	530
Mean best 10 (kg)	1051	917	1032
Standard deviation (kg)	277	253	261
Number of solutions above 95% best (1011 kg)	35	0	1
Number of solutions above 90% best (958 kg)	88	14	31
Total number of solutions	345	94	756

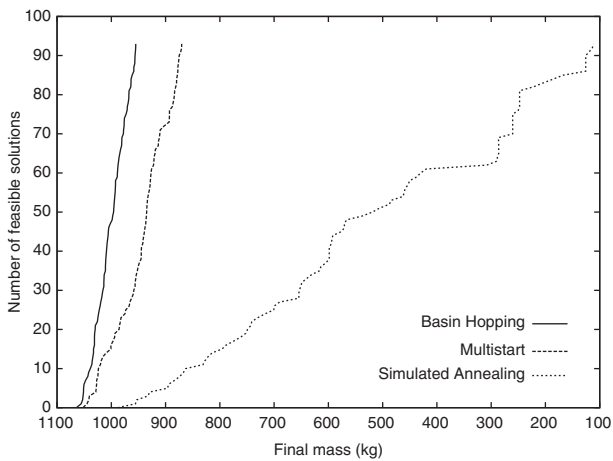


Fig. 6 Cumulative number of solutions for the EVMMM mission

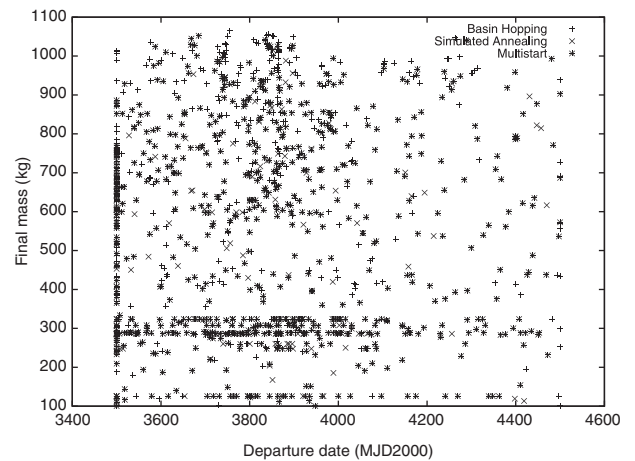


Fig. 7 EVMMM solutions with various launch dates

first case, the number of flybys increases from one to four and the dimension of the problem increases to 222 with 99 non-linear constraints. Also, a rendezvous with Mercury with a V_1 of zero is not required, but the spacecraft is allowed to have a 0.5 km/s speed difference with respect to the planet.

Because of the added complexity, the global solvers were allowed to run for a longer time (~48 h), and a more performant machine (Intel Xeon@2.66 GHz with 8 GB of RAM) is used. The test results are summarized in Table 4. Monotonic BH found more than 300 locally optimal solutions, of which more than 80 are above 90 per cent of the best solution; while MS found double the number of solutions than BH, but with fewer high final mass trajectories (e.g. only 30 solutions are above the best 90 per cent). SA only found about 100 locally optimal solutions and with less good quality solutions than the other two algorithms.

From Figs 6 and 7, it is noted that the first one-third of the solutions have a high final mass (over 900 kg) and it covers many launch opportunities within the search space. If the analysis is not restricted to the solutions that are also locally optimal, then the

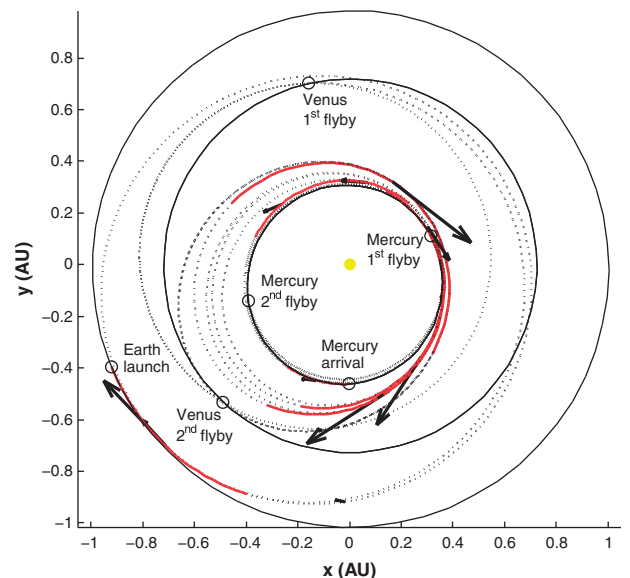


Fig. 8 Trajectory plot of a mission to Mercury

range of launch opportunities increases even more. The trajectory of the 'best' solution (with the highest final mass) is plotted in Fig. 8. Here, the spacecraft leaves Earth on 13 April 2010, performs two flybys at

Venus to lower its orbit, then encounters Mercury twice to lower its V_∞ to 0.5 km/s before it arrives on 5 September 2015. Comparing such trajectory with the one described in the BepiColombo early mission planning stage, it is noted that the results are comparable with those described in [33] (as cited in [34]) with regard to the overall shape of the optimal trajectory, which includes the planetary resonances, the times of flight, and the engine thrust periods, but have been obtained using a completely automated process.

6 CONCLUSIONS

Global optimization techniques are successfully applied to achieve the automated design of two instances of MGA LT interplanetary trajectories within a ten-year wide launch window. The use of the technique described in this study is not limited to the two particular problems here studied as it rests upon a general interface between the Sims–Flanagan trajectory model and a global optimization layer hybridized with a local search as to deal efficiently with the non-linear constraints. The resulting method makes no use of expert knowledge and starts from randomly generated trajectories, thus achieving a completely automated design process.

© Authors 2011

REFERENCES

- 1 Di Lizia, P. and Radice, G. Advanced global optimisation tools for mission analysis and design. Technical Report 03-4101b. European Space Agency, the Advanced Concepts Team, 2004.
- 2 Myatt, D. R., Becerra, V. M., Nasuto, S. J., and Bishop, J. M. *Advanced global optimisation tools for mission analysis and design*. Technical report 03-4101a, European Space Agency, the Advanced Concepts Team, 2004.
- 3 Izzo, D., Becerra, V. M., Myatt, D. R., Nasuto, S. J., and Bishop, J. M. Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories. *J. Global Optimization*, 2007, **38**(2), 283–296.
- 4 Vinko, T. and Izzo, D. *Global optimisation heuristics and test problems for preliminary spacecraft trajectory design*. Technical report GOHTPPSTD, European Space Agency, the Advanced Concepts Team 2008.
- 5 Vasile, M. and De Pascale, P. Preliminary design of multiple gravity-assist trajectories. *J. Spacecraft Rockets*, 2006, **43**(4), 794–805.
- 6 Armellin, R., Di Lizia, P., Topputo, F., Lavagna, M., Bernelli-Zazzera, F., and Berz, M. Gravity assist space pruning based on differential algebra. *Celestial Mech. Dyna. Astron.*, 2010, **106**(1), 1–24.
- 7 Izzo, D. Global trajectory optimization competition portal, June 2009, available from <http://www.esa.int/gsp/ACT/mad/op/GTOC/index.htm>
- 8 Izzo, D., Vinko, T., and Zapatero, M. Global trajectory optimization competition database, June 2009, available from <http://www.esa.int/gsp/ACT/inf/op/globopt.htm>
- 9 Addis, B., Cassioli, A., Locatelli, M., and Schoen, F. A global optimization method for the design of space trajectories. *Comput. Optim. Appl.*, 2009, **48**(3), 635–652.
- 10 Cassioli, A., Di Lorenzo, D., Locatelli, M., Schoen, F., and Sciandrone, M. Machine learning for global optimization. Technical Report 2360, Optimization Online, 2009.
- 11 Schütze, O., Vasile, M., Junge, O., Dellnitz, M., and Izzo, D. Designing optimal low-thrust gravity-assist trajectories using space pruning and a multiobjective approach. *Eng. Optim.*, 2009, **41**(2), 155–181.
- 12 Alemany, K. and Braun, R. D. Survey of global optimization methods for lowthrust, multiple asteroid tour missions. In AAS/AIAA Space Flight Mechanics Meeting, AAS paper 07-211, Sedona, Arizona, January 2007.
- 13 Yam, C. H., Biscani, F., and Izzo, D. Global optimization of low-thrust trajectories via impulsive delta-V transcription. In 27th International Symposium on *Space technology and science*, ISTS paper 2009-d-03, Tsukuba, Japan, 5–10 July 2009.
- 14 Vavrina, M. A. and Howell, K. C. Global low-thrust trajectory optimization through hybridization of a genetic algorithm and a direct method. In AIAA/AAS Astrodynamics Specialist Conference, AIAA paper 2008-6614, Honolulu, Hawaii, 18–21 August 2008.
- 15 Sims, J. A. and Flanagan, S. N. Preliminary design of low-thrust interplanetary missions. In AAS/AIAA Astrodynamics Specialist Conference, AAS paper 99-338, Girdwood, Alaska, 15–19 August 1999.
- 16 Tsiolkovsky, K. E. *Exploring the unknown*, Vol I, 1995, pp. 59–84 (NASA SP-4407).
- 17 Gill, P. E., Murray, W., and Saunders, M. A. SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM J. Optim.*, 2002, **12**(4), 979–1006.
- 18 Gill, P. E., Murray, W., and Saunders, M. A. User's guide for SNOPT version 7, software for large-scale nonlinear programming. Stanford Business Software Inc., February 2006.
- 19 Lourenço, H. R., Martin, O. C., and Stülze, T. Iterated local search. In *Handbook of metaheuristics* (Eds F. W. Glover and G. A. Kochenberger), 2003, pp. 321–353 (Kluwer Academic Publishers, Boston, Dordrecht, London).
- 20 Wales, D. J. and Doye, J. P. K. Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms. *J. Phys. Chem. A*, 1997, **101**(28), 5111–5116.
- 21 Wales, D. J. and Scheraga, H. A. Global optimization of clusters, crystals, and biomolecules. *Science*, 1999, **285**, 1368–1372.
- 22 Leary, R. H. Global optimization on funneling landscapes. *J. Global Optim.*, 2000, **18**, 367–383.

- 23 **Grosso, A., Jamali, A. R. M. J. U., Locatelli, M., and Schoen, F.** Solving the problem of packing equal and unequal circles in a circular container. *J. Global Optim.*, July, 2009.
- 24 **Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P.** Optimization by simulated annealing. *Science*, 1983, **220**, 671–680.
- 25 **Corana, A., Marchesi, M., Martini, C., and Ridella, S.** Minimizing multimodal functions of continuous variables with the ‘simulated annealing’ algorithm. *ACM Trans. Math. Softw.*, 1987, **13**(3), 280.
- 26 **Greeley, R. and Johnson, T.** *Report of the NASA science definition team for the Jupiter Icy Moons Orbiter*. Technical report, NASA Science Definition Team, February 2004.
- 27 **Whiffen, G. J.** An investigation of a Jupiter Galilean Moon Orbiter trajectory. In AAS/AIAA Astrodynamics Specialist Conference, AAS paper 03-554, Big Sky, Montana, 3–7 August 2003.
- 28 **Yam, C. H., McConaghy, T. T., Chen, K. J., and Longuski, J. M.** Preliminary design of nuclear electric propulsion missions to the outer planets. In AIAA/AAS Astrodynamics Specialist Conference, AIAA 2004-5393, Providence, Rhode Island, 16–19 August 2004.
- 29 **Yam, C. H., McConaghy, T. T., Chen, K. J., and Longuski, M. J.** Design of low-thrust gravity-assist trajectories to the outer planets. In 55th International Astronautical Congress, IAC paper IAC-04-A.6.02, Vancouver, Canada, 4–8 October 2004.
- 30 **Parcher, D. W. and Sims, J. A.** Gravity-assist trajectories to Jupiter using nuclear electric propulsion. In AAS/AIAA Astrodynamics Specialist Conference, AAS paper 05-398, Lake Tahoe, California, 7–11 August 2005.
- 31 **Strange, N. J. and Sims, J. A.** Methods for the design of V-infinity leveraging maneuvers. In AAS/AIAA Astrodynamics Specialist Conference, AAS paper 01-437, Quebec City, Canada, 30 July to 2 August 2001.
- 32 **Garcia Yarnoz, D., Jehn, R., and Croon, M.** Interplanetary navigation along the low-thrust trajectory of BepiColombo. *Acta Astronaut.*, 2006, **59**(1–5), 284–293.
- 33 **Katzkowski, M., Corral, C., Jehn, R., Pellon, J.-L., Landgraf, M., Khan, M., Yanez, A., and Biesbroek, R.** *BepiColombo Mercury cornerstone mission analysis: input to definition study*. Technical report, ESA European Space Operations Centre, April 2002.
- 34 **Gil-Fernández, J., Graziano, M., Gomez-Tierno, M. A., and Milic, E.** Autonomous low-thrust guidance: application to SMART-1 and BepiColombo *Ann. N. Y. Acad. Sci.*, 2004, **1017**. (Astrodynamics, Space Missions, and Chaos), 307–327.