Global Optimization of Low-Thrust Trajectories via Impulsive Delta-V Transcription

Chit Hong Yam,[1)] Francesco Biscani,[1)] Dario Izzo[1)]

[1)]*European Space Agency, ESTEC, The Netherlands*

Low-thrust trajectories can be modeled as a series of impulses connected by conics. We first describe a new framework for global optimization for the preliminary design of low-thrust trajectories that makes use of the impulsive delta-V transcription model. We then apply two global optimization algorithms, differential evolution and simulated annealing with adaptive neighbourhood, to solve an Earth-Mars rendezvous example problem. Test cases reveal that the two algorithms are able to produce good initial guesses for a local optimizer, which suggests a promising direction of research in automated trajectory design.

Key Words: Global Optimization, Trajectory Optimization, Low-Thrust

Nomenclature

J	:	value of the objective function
N	:	number of impulses
$\mathbf{S}$	:	state vector (position and velocity)
T	:	spacecraft thrust
$\mathbf{V}_\infty$	:	hyperbolic excess velocity
m	:	spacecraft mass
r	:	spacecraft position
t_f	:	encounter time at arrival
t_0	:	encounter time at departure
v	:	spacecraft velocity
$\mathbf{\Delta V}$	:	impulsive velocity change
δr	:	mismatch in position
δv	:	mismatch in velocity
Subscripts		
mb	:	matchpoint from backward propagation
mf	:	matchpoint from forward propagation
x, y, z	:	orthogonal inertial directions

1. Introduction

In recent years, several interplanetary missions[1-4)] have demonstrated the use of electric propulsion (EP) as the main propulsion system of the spacecraft. With a specific impulse (I_{sp}) of about ten times that of a chemical rocket, electric propulsion provides a large saving in propellant mass (and thus mission costs). However, the design of trajectories for electric propulsion missions is often more challenging than missions using chemical propulsion. Acceleration of a spacecraft propelled by EP is very small and the thrust duration can be a significant portion of the total time-of-flight.

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 do not require an initial guess to search for optimal solutions, but can suffer with long computational time when the complexity of the problem is large.

In this paper, we test different meta-heuristics global optimization algorithms to optimize low-thrust trajectories under an impulsive ΔV transcription model. We study the performance of the algorithms on an example mission to demonstrate an automated process of preliminary trajectory design.

2. Trajectory Model

Figure 1 illustrates the trajectory model proposed by Sims and Flanagan[5)]. Trajectory is divided into legs which begin and end with a planet. Low-thrust arcs on each leg are modeled as sequences of impulsive maneuvers ($\mathbf{\Delta V}$), connected by conic arcs. We denote the number of impulses (which is the same as the number of segments) with N. The $\mathbf{\Delta V}$ at each segment should not exceed a maximum magnitude, ΔV_{max}, where ΔV_{max} is the velocity change accumulated by the spacecraft when it is operated at full thrust during that segment:

$$\Delta V_{max} = \left(T_{max}/m\right)\left(t_f - t_0\right)/N \qquad (1.)$$

At each leg, trajectory is propagated (with a two-body model) forward and backward to a matchpoint (usually halfway through a leg), where the spacecraft state vector becomes $\mathbf{S}_{mf} = \{r_x, r_y, r_z, v_x, v_y, v_z\}_{mf}$ (and similarly for $\mathbf{S}_{mb}$). The forward- and backward-propagated half-legs should meet at the matchpoint, or the mismatch in position and velocity:

$$\left\{\delta r_x, \delta r_y, \delta r_z, \delta v_x, \delta v_y, \delta v_z\right\} = \mathbf{S}_{mb} - \mathbf{S}_{mf} \qquad (2.)$$

should be less than a tolerance in order to have a feasible trajectory.

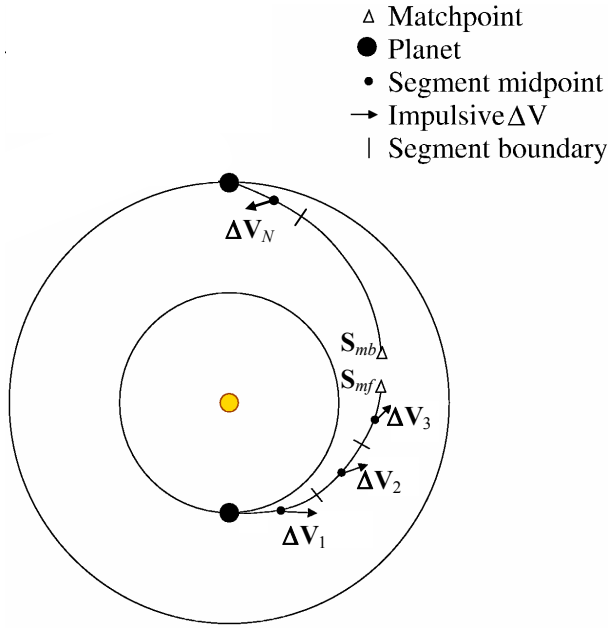

Fig. 1. Impulsive ΔV transcription of a low-thrust trajectory, after Sims and Flanagan[5].

3. Optimization Problem

3.1. The Objective Function

In space mission design, the problem of low-thrust trajectory optimization is usually to maximize the final spacecraft mass (or to minimize the propellant), subject to a set of constraints [e.g. the state mismatch constraints in Eq. (2.)]. In the context of global optimization (GO), we use a penalty function approach to include the constraints in the objective function. This transforms a nonlinear constrained problem into an unconstrained optimization problem, allowing the GO-algorithms to work on a single objective function. The objective function, J (to be *minimized*), is written as a sum of the impulses and the state mismatches:

$$J = \sum_i^N \|\Delta\mathbf{V}_i\| + k_r\sqrt{\delta r_x^2 + \delta r_y^2 + \delta r_z^2} + k_v\sqrt{\delta v_x^2 + \delta v_y^2 + \delta v_z^2} \quad (3.)$$

where k_r and k_v are scaling parameters on the position and velocity mismatches. The variables in Eq. (3.) are expressed in canonical units, where the distance unit for the position mismatches is in AU (astronomical unit); while the velocity unit for the velocity mismatches and the $\Delta\mathbf{V}$ equals to 29.78 km/s ($=\sqrt{\mu_{sun}/AU}$, where μ_{sun} is the gravitational parameter of the Sun).

3.2. The Decision Vector

The objective in Eq. (3.) is a function of a set of variables, known as the decision vector. Under our framework, the decision vector includes the following:

1) the launch date (at Earth),
2) the launch V_∞ magnitude,
3) the launch V_∞ direction parameters (known as u, v),
4) the time of flight,
5) the throttle parameters τ on the $\Delta\mathbf{V}$ magnitudes, and
6) the $\Delta\mathbf{V}$ direction parameters (known as ΔV_u, ΔV_v).

Following the work by Izzo[6], we use *sphere point picking*[7] parameters to represent the direction of a vector. The parameters u and v, ranges from 0 to 1, are related to $\mathbf{V}_\infty$ by the following equations:

$$\theta = 2\pi u \quad (4.)$$
$$\cos\varphi = 2v - 1 \quad (5.)$$
$$\mathbf{V}_\infty = V_\infty[\cos\theta\sin\varphi\ \hat{x} + \sin\theta\sin\varphi\ \hat{y} + \cos\varphi\ \hat{z}] \quad (6.)$$

and similarly for the relationship between $\Delta\mathbf{V}$ and its direction parameters $\Delta\mathbf{V}_u$ and $\Delta\mathbf{V}_v$. The sphere point picking parameters is particularly useful in connection with the implementation of the global optimizers as it allows the optimizers to select a uniformly distributed direction when u and v are randomly chosen within their range [0,1]. The use of u and v ensures that the populations initialized at random by the global optimization algorithms are uniformly distributed in the physical space. On the other hand, if the angles θ and φ are randomly selected within [0,2π] and [0,π] radians respectively, each direction of the $\Delta\mathbf{V}$ would be statistically favoured to be out-of-plane.

The throttle parameter τ, ranges from 0 to 1, relates to the magnitude of the $\Delta\mathbf{V}$ with:

$$\Delta V = \tau \cdot \Delta V_{\max} \quad (7.)$$

For a problem with N segments, there are N $\Delta\mathbf{V}$ vectors and each $\Delta\mathbf{V}$ is represented by three variables (τ, ΔV_u, ΔV_v). Including the other variables, the total number of variables (dimension of the decision vector) is $3N + 5$.

3.3. Global Optimization Algorithms

The algorithms used in our experiments are differential evolution (DE) and simulated annealing with adaptive neighbourhood solver (SA-AN).

Differential evolution is a popular algorithm for the optimization of multidimensional functions belonging to the class of evolutionary optimizers. The important element that sets DE apart from other population-based techniques (such as genetic algorithms) is the mutation mechanism, which is based on the computation of weighted differences between population vectors. DE has been proven to be effective in optimizing chemical spacecraft trajectories, as shown for instance in Refs. 6,8,9). Our version of DE follows the standard implementation presented in Storn and Price[10], using the second strategy with weighting factor F=0.8 and crossover constant CR=0.8.

Simulated annealing[11] is another popular probabilistic metaheuristic for global optimization problems. The algorithm is inspired by annealing in metallurgy, a technique which involves heating and controlled cooling of a material in order to increase the size of its crystals and reduce their defects. The heating provides a mechanism to unstuck the atoms from their initial configuration (i.e., local minima of the internal energy) and to let them wander randomly through states of higher energy, while the cooling increases the chances to locate configurations with energy lower than the initial state. In our implementation of simulated annealing we adopted the variant described in Corana et al.[12]

For both algorithms, the initial points (or population) for each run were generated by picking random values for the

components of the decision vector within fixed boundaries. All variables are bounded between 0 and 1, except for the launch date, the launch V_∞ magnitude, and the time of flight (see Table 1 for their bounds). For each trial we let the optimizers run for a fixed number (2 millions for *N=10*, 4 millions for *N=20*, and 6 millions for *N=30*) of objective function evaluations. In the case of SA-AN, we restarted the annealing process every 10,000 iterations in order to give the chance to the algorithm to perform a more global search.

3.4. Local Optimization

To test the merit of the solutions found by the two algorithms, the final decision vectors found by the two GO-algorithms are used as initial guesses for a local optimizer. Local optimization can refine the solutions by further reducing the constraint violations and improve in the value of the objective function. We solve the local optimization problem using a direct method (also known as nonlinear programming) described as follows.

In the global optimization step, for simplicity, we assume that the spacecraft mass is constant throughout the trajectory. In the local optimization step, however, mass is propagated along the trajectory. The spacecraft mass before and after a $\Delta\mathbf{V}$ are related by the rocket equation[13]:

$$m_{i+1} = m_i \exp\left(-\Delta V_i / g_0 I_{sp}\right) \tag{8.}$$

where g_0 is the standard gravity (9.80665 m/s^2) and I_{sp} is the specific impulse of the spacecraft's thruster.

Unlike the objective function used in global optimization [see Eq. (3.)], the main objective function of the local optimization is treated separately from the constraints. The problem here is to *maximize* the final spacecraft mass

$$J' = m\left(t_f\right) = m_f \tag{9.}$$

subject to the constraints on the mismatches on the position, velocity and mass at the matchpoint to be less than a tolerance.

We use the software package SNOPT[14] to solve the constrained nonlinear optimization problem. SNOPT implements sequential quadratic programming (SQP) to find a *locally* optimal solution to the problem. We can specify some parameters for SNOPT which determines when to end an optimization run. In our experiment, we set the major iteration limit (i.e., maximum number of major iterations) to be 500, the major feasibility tolerance to be 10^{-6}, and the major optimality tolerance to be 10^{-4}. The major feasibility tolerance specifies how tightly the constraints should be satisfied, while the major optimality tolerance specifies closeness to the optimality conditions. For a detailed explanation of these parameters, please see Ref. 15). We note that the framework we present here is similar to that of the trajectory design tools MALTO[16] (developed by the Jet Propulsion Laboratory) and GALLOP[17,18] (developed by Purdue University).

4. Earth-Mars Rendezvous Example

In this section, we demonstrate with an example an automated, end-to-end preliminary trajectory design process described in section 3.

4.1. Mission Parameters

We consider an Earth-Mars rendezvous mission described in Table 1. The spacecraft is assumed to have a thruster with a constant maximum thrust and a constant specific impulse. By rendezvous, we mean that the heliocentric position and velocity of the spacecraft matches with that of the target (i.e., Mars). This rendezvous terminal condition is satisfied by construction under the Sims and Flanagan model (see Fig. 1). In our experiment, we set the scaling parameters of the state mismatch constraints [see Eq. (3.)], k_r and k_v, to 1000 in order to penalize heavily infeasible trajectories.

Table 1. Parameters of an Earth-Mars rendezvous mission.

Parameters	Values
Initial mass of the spacecraft	1500 kg
Spacecraft thrust	$\leq$ 0.135 N
Spacecraft specific impulse	3000 s
Maximum time of flight	1000 days
Launch date	Jan. 1, 2015 to Jan. 1, 2025
Launch V_∞	$\leq$ 3.0 km/s

4.2. Results

We first present the results found by the two global optimizers (DE and SA-AN). Each algorithm generates 100 independent trajectories and evolved for a fixed number of function evaluations stated in section 3.3. Figure 2 shows an example of how the value of objective function (*J*) changes with the number of function evaluations in differential evolution. Each line represents a single run initialized with random seed values (100 in total).

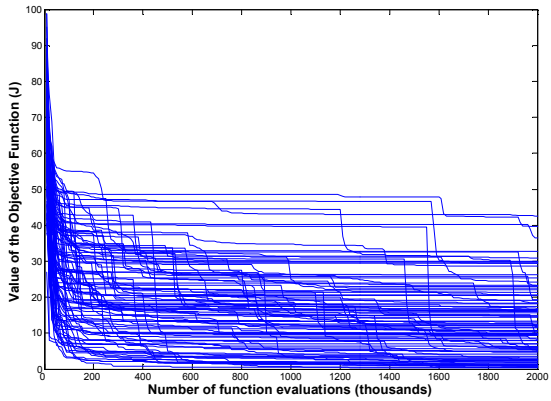

Fig. 2. Evolution of the value of the objective function during the optimization runs (algorithm: differential evolution, $N = 10$).

Table 2 summarizes the results found by the global optimization algorithms for trajectories with 10, 20, and 30 segments, showing the best and the worst solution (in terms of *J*) returned, together with the mean and the standard deviation (S.D.) of all the runs. On average, the performance of differential evolution is quite stable as the mean value does not change with the number of segments (*N*), while simulated annealing only perform better for small *N*. However we also note that for every *N*, the best cases found by SA-AN are better than those found by DE. At the same time, the worst cases found by SA-AN are much worse than those found by

DE. Figure 3 shows the values of the objective function and the launch epochs of the 100 trajectories found by DE and SA-AN on the 10-segment case. We note that many "good" solutions found by SA-AN which have low objective values J, but there are also solutions with high values of J, which explains why the standard deviation is high and implies that J is more "scattered" than those found by DE.

Figure 4 plots the flight time and the launch epoch of the solutions at the end of the optimization run. It is worth noting that solutions found by the two algorithms cover different regions of the design space. The global algorithms also find solutions with repeating patterns on the launch epoch defined roughly by the Earth-Mars synodic period (similar to the case of chemical trajectories), without any prior guidance from the user.

Table 2. Value of the objective function, J (the lower the better), at the end of the global optimization run (100 trajectories on each case).

Algorithms	Best	Worst	Mean	S.D.
DE, N=10	0.191	42.5	12.8	9.9
SA-AN, N=10	0.042	55.9	6.58	13.7
DE, N=20	0.149	39.4	12.5	9.9
SA-AN, N=20	0.104	113.5	22.2	22.5
DE, N=30	0.170	45.8	12.7	10.2
SA-AN, N=30	0.134	361.9	29.1	40.3

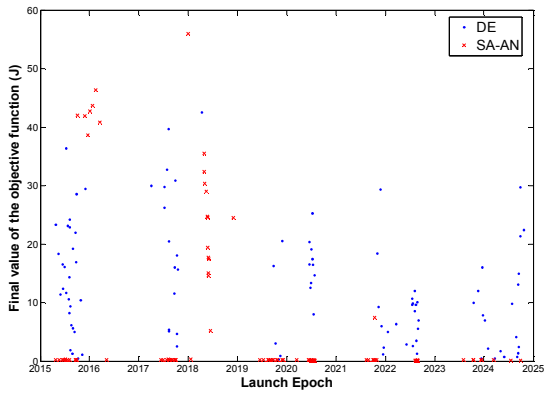

Fig. 3. Global optimization results for an Earth-Mars rendezvous mission (N=10) after 2 million function evaluations (number of trajectories = 100).

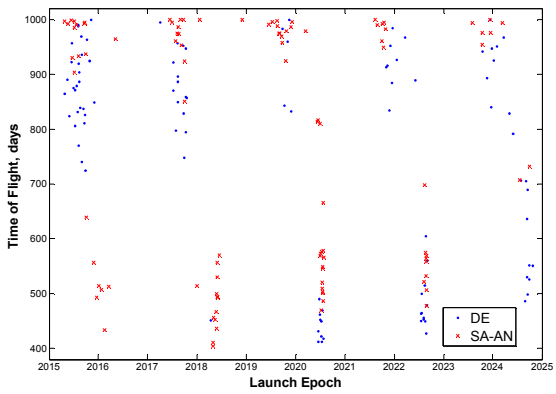

Fig. 4. Global optimization results for an Earth-Mars rendezvous mission (N=10) after 2 million function evaluations (number of trajectories = 100).

As a second step, solutions found by DE and SA-AN are submitted as initial guesses for local optimization (to maximize the final mass) and the results are summarized in Table 3. The best trajectory found by differential evolution on the 30-segment case is plotted on Fig. 5. The spacecraft leaves the Earth with a V_∞ of 3 km/s (the upper bound) and rendezvous with Mars in 974 days. On the plot, a black dot represents the midpoint of a segment and a red line represents the **ΔV** magnitude and its direction. Thrusting and Coasting segments can be distinguished by whether a black dot contains a red line or not. We note that the trajectory shown in Fig. 5 is almost ballistic except for the final rendezvous phase.

Table 3. Final mass in kg (the higher the better) after local optimization.

Algorithms	Best	Worst	Mean	S.D.	Conver-gence rate
DE, N=10	1372.3	1319.3	1361.8	14.2	98 %
SA-AN, N=10	1372.3	1236.5	1355.4	20.1	100 %
DE, N=20	1372.1	1309.5	1363.6	13.1	100 %
SA-AN, N=20	1372.1	1218.2	1354.8	27.4	93 %
DE, N=30	1372.1	1318.4	1361.6	12.9	100 %
SA-AN, N=30	1372.1	1236.3	1356.7	23.0	90 %
Random, N=10	1372.3	1183.8	1324.1	55.2	63 %

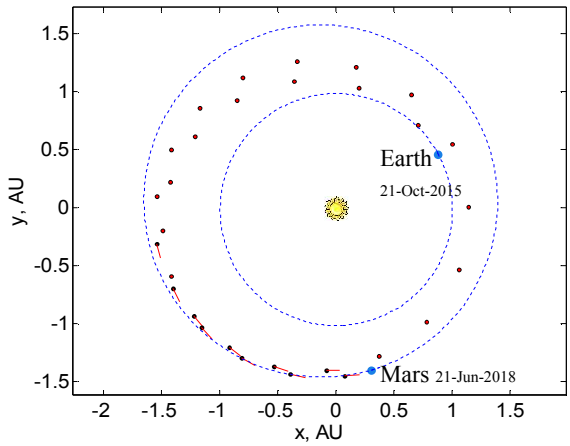

Fig. 5. "Best" trajectory (N=30) found by differential evolution (DE) and later optimized by SNOPT, final mass = 1372.1 kg.

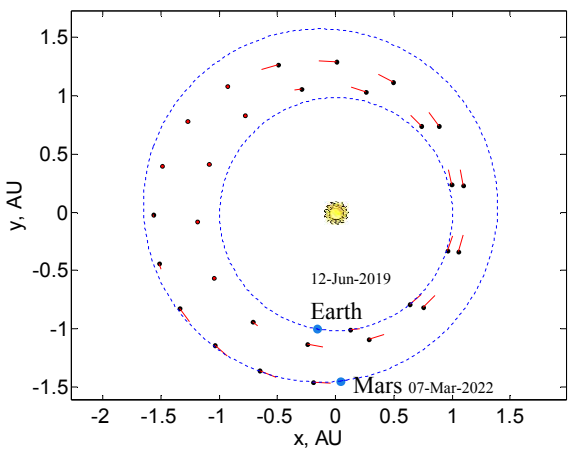

Fig. 6. "Worst" trajectory (N=30) found by simulated annealing (SA-AN) and later optimized by SNOPT, final mass = 1236.3 kg.

The results found by the local optimization (shown in Table 3) are consistent to those found by the global optimizers (shown in Table 2). Differential evolution has a more stable performance than simulated annealing in terms of the mean values, convergence rate, and the worst cases. Even though the "worst" solution found by SA-AN has a low final mass value, it is worth noting from the astrodynamics point of view that this solution belongs to a different family of trajectories from the "best" case. For example, for the 30-segment case, the "worst" trajectory found by SA-AN is plotted on Fig. 6. The spacecraft leaves the Earth with a zero V_∞ and therefore it requires more thrusting ($\mathbf{\Delta V}$) to increase its orbital energy to reach Mars.

In our test cases, the GO-algorithms are able to provide good initial guesses to the local optimizer, SNOPT. As shown in Table 3, the convergence rate for SA-AN is at least 90% and for DE is at least 98%, which is a lot higher than the convergence rate of using random initial guesses (63%). We also note from Table 4 that DE requires fewer number of major iterations in SNOPT (on average) than SA-AN (and random initial guesses), meaning that computational time is shorter for DE to complete the optimization run.

Table 4. Number of major iterations required in SNOPT (see section 3.4 for its stopping criteria).

Algorithms	Mean	S.D.
DE, N=10	172	137
SA-AN, N=10	189	159
DE, N=20	193	122
SA-AN, N=20	267	202
DE, N=30	180	117
SA-AN, N=30	253	159
Random, N=10	292	188

4.3. Discussion

We make the following observations and comments based on the research we have done.

1) In our setup, the two GO-algorithms are able to find many feasible trajectories, which provide good initial guesses for SNOPT. However, most solutions found by the GO do not have the "bang-bang" control structure on the $\mathbf{\Delta V}$, which we expect a locally optimal solution should have. One potential solution to this problem is to assume an "on/off" profile on the $\mathbf{\Delta V}$ and use the switch-on/off times as decision variables[18]. Another possible solution is to employ a "global-local" hybrid approach similar to the work by Vavrina and Howell.[19]
2) An "optimal" choice of the scaling parameters on the state mismatch constraints [see Eq. (3.)], k_r and k_v, is still yet unknown. In principle, the higher these scaling parameters, the more the optimizer would focus on the constraints and ignore the main objective. More research has to been done to study on how does k_r and k_v affect the feasibility and optimality of the solution found by global optimization.
3) The two GO-algorithms studied in this paper seem to be able to explore different regions of the design space (see Fig. 4 for an example). From our experiment, each algorithm has its own strength and weakness: DE is more reliable overall since it finds more good solutions but few of them are "the best"; SA-AN finds the best solutions but at the same time it also finds many bad (and the worst) solutions.
4) A future research topic that can make use of the strength of DE and SA-AN (and other GO-algorithms) is to implement an "island model"[20] which allows for migration and exchange of information between algorithms.

5. Conclusions

A new framework for global optimization of low-thrust trajectories that makes use of the impulsive delta-V transcription model is presented. We apply two global optimization algorithms, differential evolution and simulated annealing with adaptive neighbourhood, to solve an Earth-Mars rendezvous example problem. Solutions found by the global optimizers are used as starting guesses for local optimization. Test cases reveal that the two algorithms are able to produce good initial guesses for their high convergence rates in the local optimizer. Our work suggests a promising direction of research in automated trajectory design.

References

1) Rayman, M. D., Varghese P., Lehman D. H., and Livesay L. L., "Results from the Deep Space 1 Technology Validation Mission," *Acta Astronautica*, **47**, No. 2, 2000, pp. 475-487.
2) Elfving, A., Stagnaro, L., and Winton, A., "SMART-1: key technologies and autonomy implementations," *Acta Astronautica*, **52**, No. 2-6, 2003, pp. 79-86.
3) Kawaguchi, J., Fujiwara, A. and Uesugi, T. K., "The Ion Engines Cruise Operation and the Earth Swingby of 'Hayabusa' (MUSES-C)," 55th International Astronautical Congress, IAC 04-Q.5.02, Vancouver, Canada, Oct. 4-8, 2004.
4) Rayman, M., "Preparing for the Dawn Mission to Vesta and Ceres," 56th International Astronautical Congress, IAC-05-A3.5.B.01, Fukuoka, Japan, Oct. 17-21, 2005.
5) Sims, J. A., and Flanagan, S. N., "Preliminary Design of Low-Thrust Interplanetary Missions," AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 99-338, Girdwood, Alaska, Aug. 1999. Also in Advances in the Astronautical Sciences, Univelt Inc., San Diego, CA, Vol. 103, Part I, 1999, pp. 583-592.
6) Izzo, D., "Advances in Global Optimisation For Space Trajectory Design," Paper ISTS 2006-d-45, 25th International Symposium on Space Technology and Science, Japan, 2006.
7) Weisstein, E. W., "Sphere Point Picking," *MathWorld*, A Wolfram Web Resource.
[URL: http://mathworld.wolfram.com/SpherePointPicking.html]
8) 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," *Journal of Global Optimisation*, **38**, pp. 283-296, 2007.
9) Vinkó, T., Izzo, D., and Bombardelli, C., "Benchmarking different global optimisation techniques for preliminary space trajectory design," 57th International Astronautical Congress, Hyderabad, India, 2007.
10) Storn, R. and Price, K., "Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, **11**, pp.341-359, 1997.
11) Kirkpatrick, S., Gelatt, C. D., Jr., and Vecchi, M. P., "Optimization by Simulated Annealing," *Science*, **220**, pp. 671-680, 1983.
12) Corana, A., Marchesi, M., Martini, C., and Ridella, S., "Minimizing multimodal functions of continuous variables with the 'simulated

annealing' algorithm," *ACM Trans. Math. Softw.*, **13**, pp. 262-280, 1987.

13) Tsiolkovsky, K. E., "Exploration of the Universe with Reaction Machines," (in Russian), *The Science Review*, **5**, St. Petersburg, Russia, 1903.
14) Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal on Optimization*, Vol. 12, No. 4, 2002, pp. 979-1006.
15) Gill, P. E., Murray, W., and Saunders, M. A., "User's Guide for SNOPT Version 7, Software for Large-Scale Nonlinear Programming," Feb. 2006. Available on the Stanford Business Software Inc. website [URL: www.sbsi-sol-optimize.com].
16) Sims, J. A., Finlayson, P. A., Rinderle, E. A., Vavrina, M. A., and Kowalkowski, T. D., "Implementation of a Low-Thrust Trajectory Optimization Algorithm for Preliminary Design," AIAA/AAS Astrodynamics Specialist Conference, AIAA Paper 2006-6746, Aug. 21-24, 2006, Keystone, CO.
17) McConaghy, T. T., Debban T. J., Petropoulos, A. E., and Longuski, J. M., "Design and Optimization of Low-Thrust Trajectories with Gravity Assists," *Journal of Spacecraft and Rockets*, Vol. 40, No. 3, 2003, pp.380-387.
18) Yam, C. H., and Longuski, J. M., "Reduced Parameterization for Optimization of Low-Thrust Gravity-Assist Trajectories: Case Studies," AIAA/AAS Astrodynamics Specialist Conference, AIAA 2006-6744, Keystone, CO, Aug. 21-24, 2006.
19) Vavrina, M. A., and Howell, K. C., "Global Low-Thrust Trajectory Optimization through Hybridization of a Genetic Algorithm and a Direct Method," AIAA/AAS Astrodynamics Specialist Conference, AIAA 2008-6614, Honolulu, HI, Aug. 18-21, 2008.
20) Izzo, D., Rucinski, M., and Ampatzis, C., "Parallel Global Optimisation Meta-Heuristics using an Asynchronous Island-Model," IEEE Congress on Evolutionary Computation (CEC 2009), Trondheim, Norway, 18-21 May, 2009.