PERGAMON

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Acta Astronautica 56 (2005) 705–720

ACTA ASTRONAUTICA

www.elsevier.com/locate/actaastro

Design of Earth–Mars transfer trajectories using evolutionary-branching technique☆

Massimiliano Vasile[a,*,1], Leopold Summerer[a], Paolo De Pascale[b]

[a] *ESA-ESTEC, Advanced Concepts Team, Keplerlaan 1, 2200 AZ, Nordwijk ZH, The Netherlands*
[b] *Politecnico di Milano, via La Masa 34, 2158 Milano, Italy*

Received 11 February 2004; received in revised form 6 August 2004; accepted 7 December 2004

Abstract

In this paper a novel global optimisation approach is used to search for potentially interesting solutions for a mission to Mars. The approach blends the characteristics of evolutionary algorithms with the systematic search, typical of branching techniques. Solutions for a roundtrip to Mars, either direct or via Venus, considering long and short stays on Mars or free return trajectories are analysed, thus providing a comprehensive view of all the opportunities within a wide range of possible launch dates. Finally, electric propulsion options are investigated including the possibility of using Mars' Lagrangian points for a low cost capture. The proposed global search was effectively able to find globally minimal Δv solutions for Earth–Mars roundtrips giving the expected characterisation of the problem under study. Moreover, it will be shown how the method was able to automatically rediscover known solutions along with new ones of practical interest for future Mars exploration.

1. Introduction

In recent years the space community has demonstrated a growing interest in global optimisation techniques as a viable tool for the design of space trajectories [1–5]. However, most of the global methods used so far can be classified as stochastic or heuristic approaches and in particular of the evolutionary [6–8] class, which represents only a portion of all available global methods [9]. In many cases they have proven to explore efficiently the solution space providing even unexpected optimal solutions or a number of good initial guesses useful for a further optimisation with more accurate local optimisation techniques.

Other classes of global methods like deterministic ones, such as branching or branch and bounds approaches, have received less attention even though they have demonstrated to be extremely effective in many other fields [9–12]. A hybridisation of stochastic and deterministic approaches could be beneficial for an

☆ Based on paper IAC-03-A.7.07 presented at the 54th International Astronautical Congress, 29 September–3 October 2003, Bremen, Germany.

* Corresponding author. Tel.: +39 02 2399 8370.
E-mail address: vasile@aero.polimi.it (M. Vasile).

[1] Presently working at the Aerospace Department of Politecnico di Milano.

0094-5765/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.actaastro.2004.12.002

improvement of the effectiveness of both of them at solving space related problems.

In this paper an analysis of a variety of Earth–Mars transfer trajectories has been performed using an innovative global optimisation approach, which combines a stochastic and a deterministic method.

The basic idea of this novel approach is to use a limited set of potential solutions evolving for a small number of generations, according to some specific evolution strategies (the stochastic step), in subregions of the solution space defined by a branching procedure (the deterministic step). On the other hand, the branching rules, i.e. the rules used to partition the solutions space, are functions of the outcome from the evolution step. This technique has been used to conduct an extensive search for families of potentially interesting transfer trajectories from Earth to Mars, and return, in view of future exploration and colonisation missions envisaged by the Aurora programme [13]. Several types of trajectories have been modelled including: multiple impulsive transfers, low thrust propulsion trajectories and indirect transfers, exploiting gravity-assisted manoeuvres or 3rd body dynamics in order to reduce either the transfer time or propellant consumption.

After an introduction of the proposed global optimisation approach, the paper presents some interesting results obtained analysing possible viable Earth–Mars transfers for the Aurora programme. On the one hand, these results represent a proof of the effectiveness of the methodology, on the other hand, they are a set of potentially useful solutions for future Mars missions.

2. General problem formulation

Optimisation problems in trajectory design can be formulated, in their general form, as

$$\begin{aligned}&\min\ f(\mathbf{y})\\&\mathbf{b}^{\mathrm{l}} \leqslant \mathbf{C}(\mathbf{y}) \leqslant \mathbf{b}^{\mathrm{u}}\\&\text{with}\quad \mathbf{y} \in D,\end{aligned} \tag{1}$$

where f is a scalar nonlinear function of a multidimensional vector $\mathbf{y}$ defined within the domain D. The domain D is a hypercube defined by the upper and lower bounds on the components of the vector $\mathbf{y}$:

$$y_i \in [b_i^{\mathrm{l}}, b_i^{\mathrm{u}}]. \tag{2}$$

The vector $\mathbf{C}(\mathbf{y})$ is formed by all nonlinear constraint functions of the vector $\mathbf{y}$. If problem (1) is twice continuously differentiable and presents a single solution, i.e. only one vector $\mathbf{y}$ in the domain D minimises f and satisfies $\mathbf{C}$, a nonlinear programming method like sequential quadratic programming (SQP) can be efficiently used. This means implicitly that the problem must be formulated properly and cannot contain non-differentiable quantities. However, even in this case the problem may present more than one solution within the required search space D.

If the problem is either non-differentiable, i.e. no gradient method can be applied, or more than a solution is expected, a global optimisation method must be considered. The idea is to perform an extensive search of the solution space D looking for possible solutions to problem (1). In this respect the interest could be more in finding a number of good initial guesses for the nonlinear programming solver, rather than finding the global optimum with a high level of accuracy.

Among all global methods two categories are here considered: heuristic methods and systematic methods.

Heuristic methods contain all methods that cannot be proven to find a global optimum with a predictable amount of work. Most stochastic methods are in this class. For this particular class, it is sometimes possible to prove convergence with probability arbitrarily close to 1 but with a number arbitrarily large of function evaluations.

Systematic methods contain all methods that (in exact arithmetic) are guaranteed to find the global optimum with a predictable (deterministic) amount of work. The bound on the amount of work is anyway quite high: exponential in the problem characteristics. The simplest systematic method for bound-constrained problems is grid search (or enumerative search) where all points on increasingly finer grids are tested and the best point on each grid is used as a starting point for local optimisation. The number of grid points grows exponentially with the dimensions of the problem and so does the amount of work. Even though systematic methods are generally more reliable than heuristic ones they need some level of insight into the problem and the structure of the objective function, to be efficient (an exception can be made for methods based on interval analysis [14]). If the problem is represented by a black box then they may not find the global optimum in a reasonable amount of time. This is understandable if we look at the density theorem [15], which states that any method based only on local information, that converges for every continuous f to a global minimiser of f in a feasible domain D must produce a sequence of points y^1, y^2, $y^3, \ldots$ that is dense in D. A well known stochastic method is represented by genetic algorithms (GA) [8] that make use of analogies to biological evolution by allowing mutations and crossing over among candidates for good local optima in the hope to derive even better ones. The original concept of genetic algorithms is to encode a potential solution (individual) of the problem under study, in the form of a binary string in which each binary number represents a chromosome of the "DNA" (or genotype) of the solution (or phenotype). More sophisticated genetic algorithms make use of the data structure of the problem to encode the individual in the more appropriate way [6]. An improvement in standard genetics, used to increase exploration capabilities, is represented by niching-GA [7]. The basic concept is that in nature different species can exploit different niches in the environment. This translates in the formation of subpopulations with each subpopulation specialised at a subtask of the problem or exploring a subregion. Subpopulations can compete as in pure GA or cooperate. In general all methods that resort to some heuristic ideas derived from biological evolution can be defined as evolutionary algorithms.

Among systematic methods there are some that split the solution domain on the basis of some local information. Each time the domain is split, a number of new branches are created, each branch corresponding to a further exploration of the solution space and each subdomain representing a node that can be expanded and explored further. If the diameter of all the nodes converges to zero, convergence of the algorithm is straightforward.

The proposed optimisation approach is composed of a stochastic and a systematic step. The stochastic step is performed using an evolution strategy and is meant to obtain information on the possible presence of optima in a subdomain $D_{l \subseteq} D$. The systematic step is performed through a branching approach and is used to partition the domain D into subdomains D_l, where the presence of an optimum is expected. Each subdomain may or may not contain the global optimum but the systematic exploration of the subdomains, allows finding a number of optima and eventually the global one. This particular hybridisation can be seen as a form of *forced* niching since populations evolving in subregions form different species.

This particular hybridisation is the first difference with respect to usual global techniques. Other peculiarities of this approach rely on the way each individual explores the solution space throughout an environment perception mechanism that will be presented in the next section. For a comparison with known methods the interested reader can refer to [16].

3. Evolutionary-branching approach

3.1. Evolution step

Each solution $\mathbf{y}$ is represented by a string, of length n, containing in the first m components integer values and in the remaining s components real values. This particular encoding allows the treatment of problems with a mixed integer-real data structure. A hypercube $\mathbf{S}$ is associated to each individual $\mathbf{y}$, the hypercube, enclosing a region of the solution space surrounding the individual, is defined by a set of intervals $\mathbf{S} = S_1 x S_2 \ldots x S_n \subseteq D_l$, where S_i contains the value of the component y_i. The solution space is then explored locally by acquiring information about the landscape within each region $\mathbf{S}$ and globally by using a population of individuals $\mathbf{y}$ with their associated intervals. Each individual can communicate its findings to the others in order to evolve the entire population towards a better status.

Evolution is governed by four fundamental operators: mutation, migration, mating and filtering. The mutation operator generates a new individual randomly perturbing an old one. The mating procedure takes two individuals and generates one or two children mixing the genotypes of the two parents. Four schemes are used to mate individuals:

- Single point crossover exchanges part of the genes between the two parents;
- Arithmetic crossover generates a new individual with an interpolation of the two parents.
- Extrapolation generates a new individual on the side of the best individual between the two parents $\mathbf{y}^1$

and $\mathbf{y}^2$ at a distance from the best parents proportional to the vector connecting the two parents:

$$\mathbf{y}^3 = \alpha(\mathbf{y}^2 - \mathbf{y}^1) + \mathbf{y}^2. \tag{3}$$

- Second order interpolation mating generates a child using two parents and the child generated with an extrapolation mating. If $\mathbf{p}$ is the vector difference between $\mathbf{y}^1$ and $\mathbf{y}^3$ and f^1, f^2, f^3 are the fitness values for the three individuals $\mathbf{y}^1$, $\mathbf{y}^2$, $\mathbf{y}^3$ respectively, then a second order one-dimensional model of the fitness function is built and the new child is generated taking the minimum of the resulting parabola (see Fig. 1):

$$\mathbf{y}^4 = \mathbf{y}^1 + \mathbf{p}\chi_{\min}, \tag{4}$$

$$f_{\min} = a(\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3)\chi^2_{\min} + b(\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3)\chi_{\min} + f(\mathbf{y}^1). \tag{5}$$

The mating operator is also used to prevent crowding of more than one principal individual in the basin of attraction of the same solution: if two or more principal individuals are colliding (intersecting their migration regions) and their reciprocal distance falls down below a given threshold, a repelling mechanism is activated which mates the worse individual (between the two colliding) with the boundaries of the subdomain D_l: each component of the selected individual is blended with the value of the furthest bound, projecting the individual into a random point within D_l, according to the following relation:

$$y_i^2 = \alpha b_i + (1 - \alpha) y_i^1. \tag{6}$$

3.1.1. Environment perception and migrations

Each region $\mathbf{S}$ is evaluated using two alternative mechanisms: breeding or *perception* and learning. Breeding generates a subpopulation and selects the best child, if better than the parent. A new region $\mathbf{S}$ is then associated to the child generating a migration of the entire subpopulation towards a place where better resources are expected. For this reason each hypercube $\mathbf{S}$ is here called *migration region*. The subpopulation is generated with the following procedure: a first child is generated, within $\mathbf{S}$, mutating the parent, then an extrapolation mating is performed. The two resulting children and the parent are then used to generate a third child using second order interpolation mating. The procedure is repeated until a number of children equal to the number of coordinates have been generated (see Fig. 1). This procedure can be seen as a way for the individual $\mathbf{y}$ to *perceive* or sense locally the environment to obtain clues about where to proceed with the exploration of the solution space.

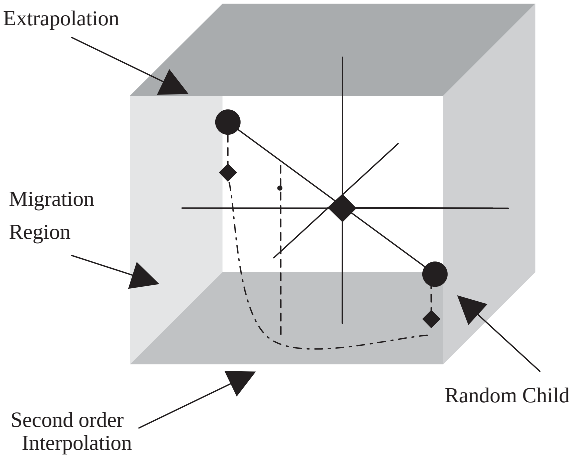

Fig. 1. Perception mechanism.

Learning is performed using two mechanisms: interval analysis and gradient methods. In the first case the inferior value of f within the migration region is associated to the individual $\mathbf{y}$ and perception is then not necessary. In the second case, if the function f is locally continuous and differentiable a number of SQP steps are taken if perception gives no results and the migration region shrinks down to a small value.

The contraction or expansion of each region $\mathbf{S}$ is regulated through a migration radius ρ and depends either on learning or on the perception mechanism. The migration radius is defined as the ratio between the value of the distance from the boundary $\mathbf{b}^j$ of the migration region of the jth individual and the value of the distance from the corresponding boundary $\mathbf{b}$ of the domain D:

$$\rho^j = \frac{b_i^j - y_i^j}{b_i - y_i^j}. \tag{7}$$

If none of the children of the subpopulation is better than the parent the radius is reduced according to:

$$\rho^j = \begin{cases} \max([1e-8, \delta y_{\min}]) & \text{if } \delta y_{\min} \geqslant \varepsilon\rho^j, \\ \varepsilon\rho^j & \text{if } \delta y_{\min} < \varepsilon\rho^j, \end{cases} \tag{8}$$

where ε has been set to 0.5 and $\delta y_{\min}$ is the distance of the best child $\mathbf{y}^*$, among the ones in the migration region, from the parent j, normalised with respect to the dimensions of the migration region:

$$\delta y_{\min} = \sqrt{\sum_{i=1}^{n} \left(\frac{y_i^* - y_i^j}{S_i^j} \right)^2}, \tag{9}$$

where for individual j and for dimension i, S_i^j is the difference between the value of the upper bound and of the lower bound and the summation is over non-zero dimensions. Now, if from generation k to generation $k+1$ the differential improvement Δf^j (the difference between the function f^j at generation k minus f^j at generation $k+1$) increases, the migration radius is recomputed according to the prediction:

$$\rho_{k+1}^j = \rho_k^j \eta \log(e - 1 + j), \tag{10}$$

where η is equal to 2 in this implementation. For integer numbers migration operates in the same way but the migration regions and migration radius are generated and treated differently. In particular if $\rho_{i\,\min}$ is 1 and ρ is defined as

$$\rho_{k+2}^j = \min[\mathrm{int}(\log(2 + j)\Delta f^j), \rho_{i\,\min}]. \tag{11}$$

The migration region is therefore contracted differently for real and for integer variables allowing a better spatial exploration.

3.1.2. Filtering

Instead of traditional selection mechanisms based on fitness here a permanent population of n individuals is maintained from one generation to another. Each individual has a chance to survive provided that it remains inside the filter. The filter ranks all the individuals on the basis of their fitness from best to worst. All the individuals with a fitness worse than a given threshold are either hibernated (i.e. no operator is applied) or mutated while migration is applied to all individuals within the filter. The probability of being mutated or hibernated depends on their ranking. This allows each of the individuals within the filter to evolve towards a different local optimum. Mating is operated between all the individuals who present an improvement after mutation or migration and all the remaining ones.

3.2. Branching step

Even though the filter increases the chances of finding several optima and eventually the global one, convergence is not guaranteed due to the stochastic nature of the process. Therefore, a systematic step is taken on the basis of the output of the evolutionary algorithm. The initial domain $D_0 \equiv D$ is partitioned generating a number of subdomains D_l. Each subdomain is then qualified and explored further according to its qualification.

The partitioning, or branching, process begins by taking the worst individual, which is out of the filter, and cutting D_0 into L subdomains, corresponding to L potentially new branches (or nodes). Each one of the L nodes may or may not contain an individual coming from the previous step of evolution and the volume of the node depends on the position of the cutting point (a safeguard mechanism prevents cuts too close to a boundary moving the cutting point to the middle of the interval). For each node D_l the ratio between the relative number of individuals and the relative volume is computed and the resulting quantity defines how necessary a further exploration of the node is

$$\varpi_{D_l} = \frac{\sum_{D_l} j}{\sum_D j} \Bigg/ \sqrt[n]{\frac{V_{D_l}}{V_D}}, \quad l = 1, \ldots, L, \tag{12}$$

where the volumes V_{D_l} and V_D are computed taking only edges with a non-zero dimension. This quantity is then added to a fitness φ_{D_l} defined as

$$\varphi_{D_l} = \begin{cases} \frac{\frac{1}{J}\sum_{j=1}^{J} f_j - f_{\text{best}}}{f_{\text{worst}} - f_{\text{best}}} & \text{if } J \neq 0, \\ 1 & \text{otherwise}, \end{cases} \tag{13}$$

where J is the number of individuals in domain D_l. If interval analysis is available each subregion is evaluated taking the inferior and superior values, the quantity φ_{D_l} is then defined as

$$\varphi_{D_l} = \inf_{D_l} (f). \tag{14}$$

The node is then qualified by the quantity:

$$\psi_{D_l} = \sigma \varpi_{D_l} + (1 - \sigma)\varphi_{D_l}, \tag{15}$$

where σ is the weighting factor that weights how reliable the result coming from the evolution step is considered. If σ is 0, only the nodes with low fitness are explored because the evolutionary algorithm is considered reliable enough to explore exhaustively the domain D without leaving any region unexplored. On the other hand, if σ is 1 the result from the evolutionary algorithm is considered to be not reliable due to a premature convergence or to a poor exploration of the solution space. Now, every time a node D_l is subdivided into other Q subnodes only the most promising pair is taken into account. If ψ_{D_l} is used to select the most promising ones among all L subdomains, the best pair out of the Q subnodes generated for each subdomain is selected using the following slightly different quantity:

$$\tilde{\psi}_{D_q} = \sigma \left. \frac{\sum_{D_q} j}{\sum_{D_l} j} \middle/ \gamma_q \right. + (1-\sigma)\varphi_{D_q}, \quad q = 1, \ldots, Q, \tag{16}$$

where γ_q is, for each of the subnodes q, the ratio between the length of the edge along which the subdomain D_l is cut and the corresponding edge of D_q. Once a D_q is selected the other subnode of the pair will be the complement $D_{q+1} = D_l - D_q$. For a fast search only nodes presenting a high fitness and large volume are explored further.

In order to avoid the rediscovery of minima already found, the original domain is partitioned using more than one individual. If the worst individual is useful to determine an upper bound on the objective function, converged individuals suggest where a further exploration is unnecessary. Therefore, in the general scheme, all converged individuals are ranked depending on the value of their fitness function. The principal cut is then, as stated above, performed using coordinates of the worst individual, the second cut takes the worst converged individual and so on up to the best converged individual.

3.3. Constraint satisfaction

The algorithm described solves bound-constrained problems but since in most of the cases constraints are nonlinear an extension of the algorithm has been developed to take into account nonlinear inequality constraints.

At each evolution step the population of solutions is divided into two subpopulations and a different objective function is assigned to each one, namely one subpopulation aims at minimising the original objective function while the other aims at minimising the residual on the constraints defined as:

$$\min_{y \in D} f(\mathbf{y}) = \sum_{j=1}^{M} \mathrm{e}^{R_j}, \tag{17}$$

where M is the number of violated constraints and R_j is the residual of the jth violated constraints. The two subpopulations are evolved in parallel and individuals are allowed to jump from one population to the other, i.e. if a feasible individual becomes infeasible it is inserted in the subpopulation of infeasible individuals and assigned to the solution of problem (17), on the other hand if an infeasible individual becomes feasible it is inserted in the population of feasible individuals and allocated to the minimisation of the original bound-constrained objective function *f*. As a result, the final optimal solution is either feasible or minimises infeasibilities.

This procedure does not maintain feasibility for any individual, therefore once a feasible set has been found the perception mechanism is used to ensure that every move maintains the feasible population inside the feasible set. If f^* is the value of the objective function of an individual $\mathbf{y}$ inside the feasible set, the objective function of a new individual generated from $\mathbf{y}$ is then augmented in the following way:

$$\min_{y \in D} f = \begin{cases} f^* & \text{if every } R_j \leqslant 0, \\ f^* + \max R & \text{if any } R_j > 0. \end{cases} \tag{18}$$

The described strategy co-evolving two populations with two different goals, allows a flexible search for feasible optimal solutions: in fact through the described use of the perception mechanism feasibility can be enforced on all feasible solutions or just on the best among the feasible ones. In the former case the exploration of the solution space may be over penalised reducing the convergence rate or leading to a local minimum, on the other hand the latter strategy, while preserving the feasibility of at least the best solution, allows a more extensive search along the boundary of the feasible region.

3.4. Interval analysis and stopping criterions

There are two combined stopping criteria: one for local convergence and one for global convergence.

Both are based on some heuristics and not on any rigorous proof of global converge. Local convergence of each subpopulation is determined by the improvement of each individual and by the migration radius. In a convex problem, both should tend to zero in the neighbourhood of the solution. Since each individual is supposed either to converge to a different minimum or not to converge (letting just the individual with highest rank in the filter to converge) a global stopping criterion for the evolution step is the convergence of the filter. The convergence of the filter is determined by the convergence of all the individuals if they are not clustered, i.e. if their migration regions are not intersecting, and, otherwise, by the convergence of the best individual. Convergence for an individual is reached when its migration radius drops below a given threshold. It must be noticed that when evolution step is used in conjunction with branching the convergence of the filter is not usually necessary since the branching takes care of the global exploration of the solution space. The global convergence of the branching is reached either if all the nodes reduce below a given tolerance or if evolution steps have converged in all subdomains and no improvement is reported after branching, i.e. no new local minima are discovered. Interval analysis, when used, guarantees that the node containing the global minimum is always in the list of explored nodes therefore if the difference between the inferior value of the best node and the best individual contained in that node is below a given tolerance, convergence to the global optimum is achieved.

The process is initialised defining the boundaries of the search space D, setting the value for σ, the number of individuals and the dimension of the filter, then the algorithm proceeds in the exploration until one of the stopping criteria is met or the maximum number of function evaluations is reached.

Human intervention is therefore limited to the initial definition of the search space and of the number of exploring individuals but it must be underlined that the branching step allows a loose definition of the boundaries as opposed to common evolutionary approaches [8]. On the other hand the stochastic nature of the evolution step makes the method robust against landscapes which might deceive systematic methods [10].

The total computational cost of each run depends on the total number of function evaluations and the level of exhaustiveness can be tuned changing σ and the number of levels of branching thus increasing or reducing the total number of function evaluations as desired. Since, here the interest is a characterisation of the solution space more than a quick convergence to the global optimum, σ has been set to 0.9, the maximum number of branching steps has been set to 4, generating a maximum of 81 subdomains and the number of function evaluation for the evolution step has been limited to 6000.

4. Characterisation of Earth–Mars roundtrips

The global search algorithm presented in the previous chapters is now applied to the problem of characterising Earth–Mars transfers.

The first analysis looks for roundtrips from Earth to Mars with minimal total Δv. Roundtrip trajectories are made of an Earth–Mars transfer, departing from either a circular or an elliptical orbit around the Earth and aiming at either a circular or an elliptical orbit around Mars, a certain stay time around Mars and a return transfer to either a circular or an elliptical orbit around the Earth. Depending on the launch date and on the transfer time, for each leg, different families of roundtrips can be envisaged. In order to include even free return trajectories, instead of a braking and a departure manoeuvre, a swing-by of Mars is performed every time the stay time drops below 1 day.

4.1. Problem modelling

Each transfer is computed in a three-dimensional heliocentric ecliptic reference frame, as the solution of a Lambert's problem for the restricted 2-body problem [17]. Each planet is considered as a point mass with no gravity. Ephemeris of the planets are computed analytically as polynomial expansions of the orbital parameters as a function of the modified Julian date (MJD2000).

The resulting global optimisation problem can be formulated as

$$\min_D f = \Delta v_1 + \Delta v_2 + \Delta v_3 + \Delta v_4, \tag{19}$$

where Δv_2 and Δv_3 represent, respectively, the braking manoeuvre and the departure manoeuvre at Mars, while Δv_1 and Δv_4 are, respectively, the departure manoeuvre and the braking manoeuvre at Earth. Each

Δv is computed as the difference between the velocity at the pericentre of the arrival or departure planetocentric hyperbola and the velocity at the pericentre of a planeteocentric orbit with pericentre and apocentre radius (r_p^E, r_a^E) for Earth and (r_p^M, r_a^M) for Mars, respectively. The v_∞ of each hyperbola is derived from the solution of the above mentioned Lambert's problem and from the velocity of each planet on its orbit.

Thus the Δvs are functions of the departure date t_0, the time of flight for the outbound leg T_1 and for the inbound leg T_2, the stay time t_s. Each solution is therefore defined by the following vector:

$$\mathbf{y} = [t_0, T_1, T_2, t_s, r_p^E, r_a^E, r_p^M, r_a^M]^T. \tag{20}$$

And the solution space D contains all possible values of **y.** In case of swing-by of Mars a linked-conic approximation of the gravity manoeuvre is used and the objective function becomes:

$$f = \Delta v_1 + 10C_1^2 + 20C_2^2 + \Delta v_4, \tag{21}$$

where the two constraint violations for the swing-by manoeuvre, C_1 and C_2 are defined as

$$C_1 = v_i^2 - v_o^2; \quad C_2 = \langle \mathbf{v}_i, \mathbf{v}_o \rangle + \cos(2\beta) v_i v_o, \tag{22}$$

where $\mathbf{v}_i$ and $\mathbf{v}_o$ are the incoming and outgoing velocity vectors relative to Mars and β is the deviation angle, function of the modulus of the incoming velocity, of the radius r_p^M of the pericentre at Mars and of the gravity constant μ_M of Mars.

$$\beta = a \cos\left(\frac{\mu_M}{v_i^2 r_p^M + \mu_M}\right). \tag{23}$$

A further analysis of return trajectories via Venus has been done, introducing an additional swing-by in the model and extending the solution vector (and therefore the domain D) as follows:

$$\mathbf{y} = [t_0, T_1, T_2, t_s, r_p^E, r_a^E, r_p^M, r_a^M, \omega, r_p^V, T_3, T_4]^T, \tag{24}$$

where now T_2 is the time of flight from Mars to Venus, r_p^V is the pericentre radius at Venus, ω is the rotation angle of the plane of the hyperbola around the incoming vector with respect to the ecliptic plane [16], T_3 is the time of flight, after the swingby, to a deep space manoeuvre and T_4 is the time of flight from the deep space manoeuvre to the Earth. The new objective function must include the deep space manoeuvre and therefore becomes:

$$\min_D f = \Delta v_1 + \Delta v_2 + \Delta v_3 + \Delta v_4 + \Delta v_5. \tag{25}$$

Table 1
Domain D for the roundtrip problem

Variable	Lower bound	Upper bound
t_0 (MJD)	5479	15775
T_1 (day)	50	700
T_2 (day)	50	700
t_s (day)	0	600
r_p^E (km)	6778	6778
r_a^E (km)	6778	6778
r_p^M (km)	3789	5.7e5
r_a^M (km)	3789	5.7e5

4.1.1. Short and long stay options

At first problem (19) was solved looking for roundtrips with a variable stay time ranging from 0 to 600 days and evaluating the total Δv necessary for each launch opportunity. Then, the search was focused on short stay opportunities including returns via Venus (commonly called opposition class missions). The solution space D for problems (19) and (21) is defined in Table 1 and comprises all the possibilities including free return trajectories. Launch windows from 2015 to 2043 were explored and all the solutions with a total Δv less than 13 km/s have been collected and plotted in Figs. 2 and 3, where diamonds represent short stay and free return options.

Then if the upper limit on total Δv is extended to 15 km/s several short stay trajectories with a return via Venus become feasible. The result has been plotted in Fig. 4. It should be noticed that a return via Venus is not always available and for each solution via Venus it is often possible to find a direct return with a comparable level of Δv. This is true apart from two launch windows in which only a return via Venus allows, for a short stay, a total Δv less than 15 km/s. Furthermore, for two particular launch opportunities (13 years away from one another) an almost continuous range of short stay periods are allowed.

The latter one of the two comprises almost all the returns via Venus since for this date Venus is in a particularly favourable position.

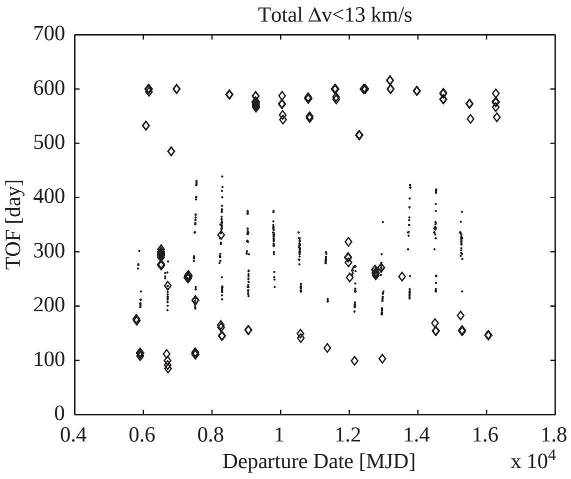

Fig. 2. Departure date vs. TOF.

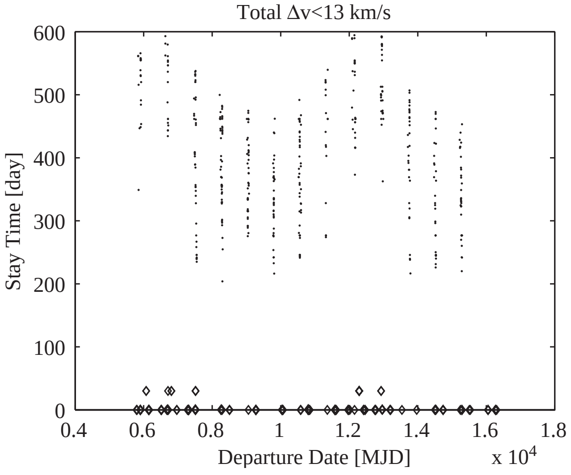

Fig. 3. Departure vs. stay time.

Very short stays (less than 10 days) are also possible both via Venus and via direct return. This class of trajectories can become interesting as abort options in case an immediate return is necessary and a manoeuvre at Mars can still be executed. The most interesting options for a short stay either via Venus are summarised in Table 2 for the period from 2028 to 2037 while optimal solutions for each launch date, in the same period, are summarised in Table 3.

Since total absorbed radiation dose is one of the key issues for human Mars mission design and since the

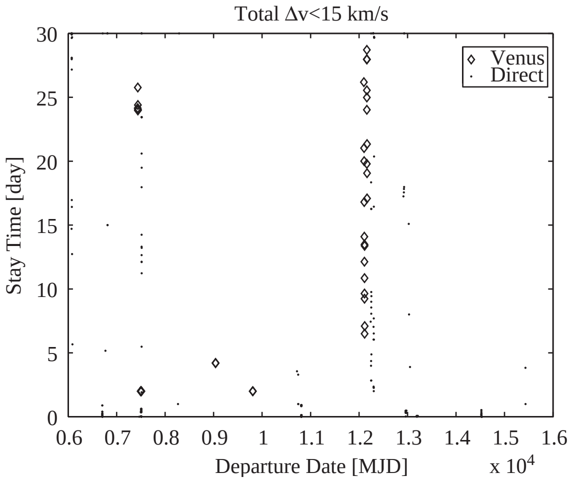

Fig. 4. Return via Venus vs. direct return.

trajectory choices are the most influential parameters, the total dose due to galactic cosmic radiation (GCR) were calculated for all presented options in Table 3. The values are equivalent BFO dose values in *sievert* (Sv). Interplanetary dose values behind 10 g/cm^2 Al are assumed at 0.24 Sv/a. Mars surface levels behind 5 g/cm^2 Al are assumed 0.15 Sv/a. Dose levels for equivalent PE shielding materials are given in brackets.

Due to the stochastic nature of the method, for this case the evolutionary-branching algorithm was run twice to check the exploration of the solution space was sufficiently exhaustive. In both cases the algorithm, implemented in a Matlab code, took about one hour on a 2 GHz Pentium 4 m with 256 Mb of RAM.

4.1.2. *Free return trajectories and cyclers*

As can be noticed in Fig. 3, for each launch window it is possible to find a solution with a stay period below 1 day, which in fact corresponds to a trajectory that departs from Earth, flies by Mars and comes back to Earth ballistically, i.e. without manoeuvres. These trajectories, known as free return trajectories, can be grouped in three main categories (see Fig. 5).

The first one comprises all transfers with a low departure velocity from Earth, a correspondent short transfer arc to Mars and a long arc leading the spacecraft back to Earth again with a low arrival velocity (see Fig. 6). The overall period in space is about two

Table 2
Best short stay options via Venus from 2028 to 2037

Launch Date	E–M (day)	M–E (day)	t_{s} (day)	Δv_1 (km/s)	Δv_2 (km/s)	Δv_3 (km/s)	Δv_4 (km/s)	Δv_5 (km/s)
21/10/2028	252	251	2	4.16	3.63	4.80	5.98	0.193
04/07/2029	529	404	2	4.28	5.28	3.27	4.72	1.579
30/06/2031	460	490	2	4.12	4.56	3.28	6.53	0.644
16/04/2033	199	352	28	3.59	2.43	4.21	3.94	6.15e-4
17/02/2035	236	318	2	4.97	3.02	4.29	3.78	3.13e-4
02/09/2037	225	475	30	4.06	2.03	3.33	4.36	1.369

Table 3
Optimal solutions for direct Earth–Mars roundtrips from 2028 to 2037

Launch Date	E–M (day)	M–E (day)	t_{s} (day)	Δv_1 (km/s)	Δv_2 (km/s)	Δv_3 (km/s)	Δv_4 (km/s)	GCR (Sv)	m_{f}/m_0
23/11/2028	300.1	353.5	344.1	3.589	2.244	2.077	3.9348	0.57(0.43)	0.035
09/08/2029	596.6	320.9	600	4.366	5.067	2.1498	4.123	0.85(0.63)	0.01
24/12/2030	283.1	217.7	497.8	3.663	2.53	1.992	3.749	0.53(0.38)	0.035
17/02/2031	210.7	217.7	514.9	3.806	2.776	1.992	3.749	0.49(0.35)	0.031
16/04/2033	199.7	198.0	553.0	3.587	2.435	2.239	3.589	0.49(0.34)	0.036
26/11/2034	248.1	251.0	30	6.1361	3.9151	3.498	5.1868	0.34(0.28)	0.003
27/06/2035	201.9	267.5	535.6	3.6447	2.07	2.5836	3.689	0.53(0.37)	0.034
02/06/2036	700	284.9	444.0	4.7548	8.347	2.313	3.5439	0.83(0.64)	0.001
15/08/2037	347.5	282.9	358.1	3.9298	2.131	2.313	3.5439	0.56(0.42)	0.035

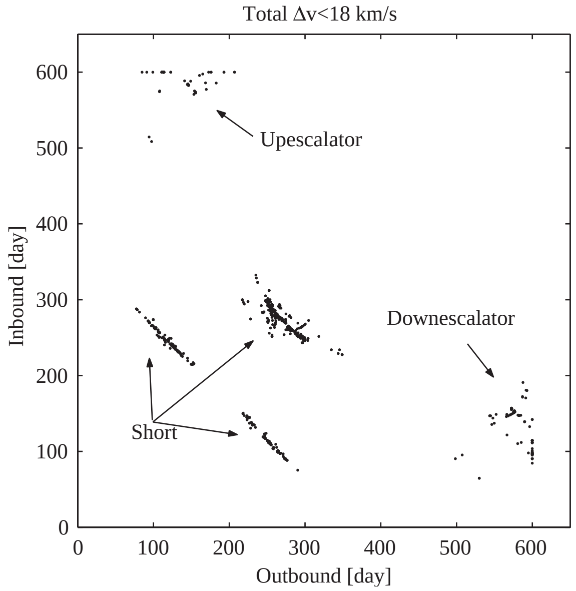

Fig. 5. Free return trajectories.

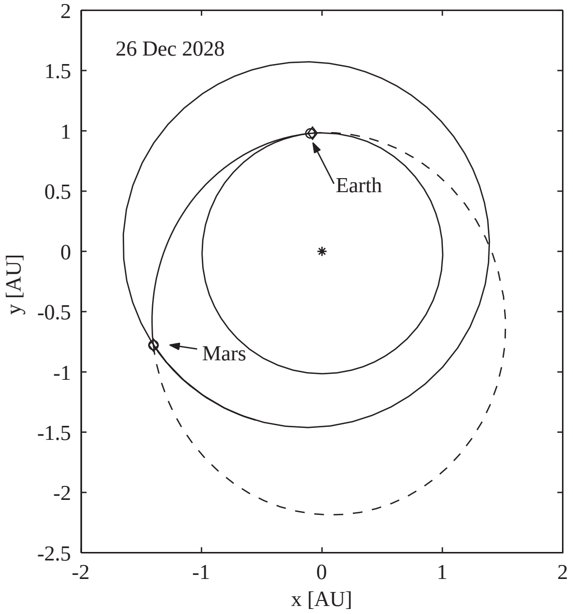

Fig. 6. Up escalator.

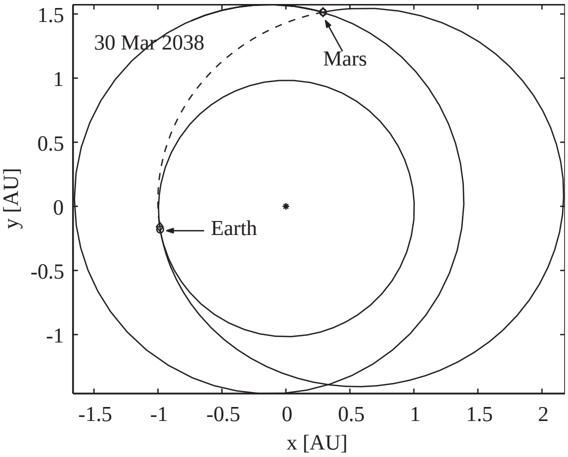

Fig. 7. Down escalator.

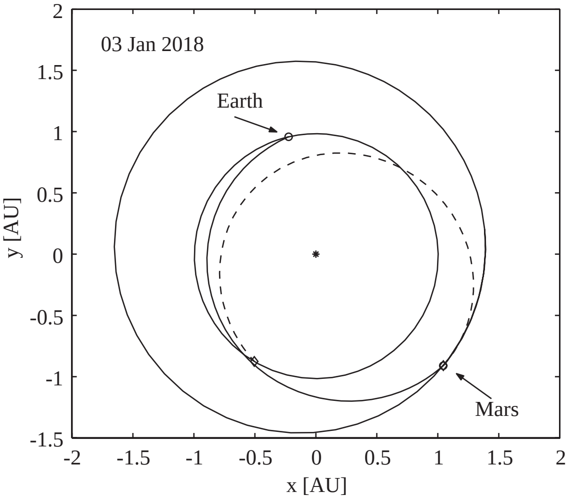

Fig. 8. Short roundtrip.

terrestrial years, therefore, in analogy with Earth–Mars one-synodic-period cyclers, these free return trajectories are here called up escalators. The second category comprises all free return trajectories with an initial long transfer to Mars and a short return leg to Earth, the total time in space is again about 2 years and therefore these trajectories are here called down escalators (see Fig. 7). The third category of free return trajectories presents a relatively short transfer time on both legs either to go or to come back (see Fig. 8).

As can be seen in Fig. 5 short free return trajectories can be subdivided further in three groups depending on the length of each leg. All best free return opportunities for the period from 2028 to 2037 have been summarised in Table 4 where transfer time, infinite velocity at Mars and Δvs at Earth are reported. As can be read, although up and down escalators are appealing for their relatively low Δv at departure they can become prohibitive if the spacecraft has to be inserted in orbit around Mars with a propulsive manoeuvre, due to the high infinite velocity.

On the other hand, some short free return options, although more demanding in terms of Δv at departure have in general a lower velocity at Mars and could be interesting either as nominal trajectories in order to increase safety for manned missions or as abort options.

In fact if a failure, not affecting the propulsion system, forces the mission to be aborted on the way to Mars, in some cases, a deep space manoeuvre, exploiting the whole remaining propellant, could be used to inject the spacecraft on a short free return transfer back to Earth.

Similar results, although for different launch dates and obtained with a fully systematic search method, can be found in [18,19].

4.2. Optimal staging

Even in case cryogenic propellants are considered for propulsion (with an $I_{sp} = 450\,\mathrm{s}$) the mass budget for a roundtrip to Mars could be prohibitive even for minimum Δv transfers (see last column of Table 3). A solution could be to resort to staging in order to improve the payload returned to Earth. Therefore, the natural extension of problem (19) is to introduce staging sequences in the model and to optimise for the final mass m_f instead of the total Δv. The staging model assumed here is fairly simple and does not take into account gravity losses. Furthermore, a constant specific impulse and a constant structural factor of 0.15 has been considered for each stage.

$$m_{i+1} = m_i \mathrm{e}^{-\Delta v/g_0 I_{sp}}, \tag{26}$$

$$m_{i+1} = m_{pl} + m_{p_{i+1}} + m_{s_{i+1}} + m_{s_i}, \tag{27}$$

$$m_s = 0.15 m_p \tag{28}$$

Table 4
Best free return options found from 2028 to 2037

Launch date	E–M (day)	M–E (day)	Δv_1 (km/s)	Δv_4 (km/s)	v_∞ (km/s)	r_p (km)
26/12/2028	141.1	588.4	4.281	4.283	10.77	43389
01/08/2029	582.49	147.4	4.349	4.347	6.961	43301
18/09/2030	272.5	253.82	5.827	8.11	5.253	3789.0
09/02/2031	122.87	600	4.285	4.319	11.653	43389
21/11/2032	252.67	255.98	4.977	6.38	5.1056	3790.7
12/04/2033	99.05	600	4.446	4.876	11.44	13732
02/01/2034	600	90.48	5.103	4.499	10.488	9840.1
25/02/2036	600	113.5	4.504	4.311	11.776	36714
24/01/2037	254.32	262.8	7.2679	5.2115	5.252	3789.0

Table 5
Mass fractions at arrival at Earth with 2 stages

Date	r_a^E (km)	ε_i^E	r_a^M (km)	ε_i^M	m_f/m_0
23/11/2028	5.4e4	0.162	1.9e5	0.03	0.168
24/12/2030	4.3e4	0.195	1.0e5	0.06	0.167
16/04/2033	4.1e4	0.197	4.1e5	0.015	0.168
09/07/2035	4.8e4	0.178	5.0e5	0.01	0.166

the objective function then becomes:

$$\max_{\mathbf{y}\in D} f = -m_f/m_0 \tag{29}$$

where m_0 is the initial mass and each solution is defined by the vector:

$$\mathbf{y} = [t_0, T_1, T_2, t_s, r_p^E, r_a^E, r_p^M, r_a^M, \varepsilon_i^E, \varepsilon_i^M]^T, \tag{30}$$

where ε_i^E and ε_i^M represent for Earth departure and for Mars insertion respectively the ratio between the apocentre of the departure (arrival) orbit r_a^E (r_a^M) and the apocentre radius of the intermediate orbits. The pericentre altitude of the departure (arrival) is constrained to be at 400 km and the initial and final orbits are assumed to be circular with the same altitude. Therefore, if a 2-stage strategy is used for Earth escape the first stage injects the spacecraft from the 400 × 400 km circular orbit into an intermediate orbit with apocentre $\varepsilon_1^E r_a^E$ and the second stage injects the spacecraft into a departure orbit with apocentre $r_{a.}^E$ Furthermore, the number of stages is fixed and equal for each escape or capture manoeuvre. In Table 5 some optimal solutions for the interval [2028, 2037] are reported for a 2-stage strategy.

5. Low-thrust transfers

All the analysis of the previous chapter assumed the use of high thrust engines, however low-thrust propulsion systems may become interesting both for manned and unmanned missions. Therefore, an analysis of direct low-thrust Earth–Mars transfer will follow. A low-thrust trajectory is here modelled using an inverse method: the Cartesian coordinates of the spacecraft are described by means of a set of *pseudo-equinoctial elements* $\boldsymbol{\alpha}$. The set of elements used to parameterise the Cartesian coordinates are here called *pseudo-equinoctial* because they do not satisfy exactly the Gauss'planetary equations unless the thrust is zero. Each element is then developed in form of a parameterised function of the anomaly L. This function is the shape of the pseudo-element.

Once position is defined in terms of the pseudo-elements velocity and accelerations can be computed by differentiation:

$$\mathbf{v} = \frac{d\mathbf{r}}{dt} = \frac{d\mathbf{r}}{dL}\frac{dL}{dt}; \quad \mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{d\mathbf{v}}{dL}\frac{dL}{dt},$$

$$\frac{d\mathbf{r}}{dL} = \sum_{i=1}^{5}\frac{\partial\mathbf{r}}{\partial\alpha_i}\frac{\partial\alpha_i}{\partial L} + \frac{\partial\mathbf{r}}{\partial L}. \tag{31}$$

In order to obtain the set of pseudo-elements that satisfies exactly the conditions at boundaries, the following nonlinear programming problem must be solved:

$$\mathbf{r}(\boldsymbol{\alpha}(L_0), L_0) = \mathbf{r}_0; \quad \mathbf{v}(\boldsymbol{\alpha}(L_0), L_0) = \mathbf{v}_0;$$

$$\mathbf{r}(\boldsymbol{\alpha}(L_f), L_f) = \mathbf{r}_f; \quad \mathbf{v}(\boldsymbol{\alpha}(L_f), L_f) = \mathbf{v}_f; \tag{32}$$

where for low values of the acceleration it is sufficient to solve the easier linear problem:

$$\boldsymbol{\alpha}(L_0)=\boldsymbol{\alpha}_0; \quad \boldsymbol{\alpha}(L_\mathrm{f})=\boldsymbol{\alpha}_\mathrm{f}. \tag{33}$$

For each set of pseudo-elements a different trajectory can be generated, connecting two points in the state space. The control acceleration $\mathbf{a}_c$ necessary to achieve the imposed shape of the trajectory can then be obtained by solving the following system:

$$\mathbf{a}_\mathrm{c}=\mathbf{a}-\frac{\mu}{r^3}\mathbf{r}; \quad m_\mathrm{f}=m_0\mathrm{e}^{-c\int_{t_0}^{t_\mathrm{f}}|\mathbf{a}_\mathrm{c}|\,\mathrm{d}t} \tag{34}$$

with c the inverse of the exhaust velocity and with the additional constraint:

$$t_\mathrm{f}-t_0=\int_{L_0}^{L_\mathrm{f}}\frac{\mathrm{d}t}{\mathrm{d}L}\,\mathrm{d}L. \tag{35}$$

This approach is extremely fast and the computational cost extremely low since no propagation or collocation is necessary.

Of course the thrust profile, though constrainable, is a direct consequence of the shape and must be considered only as a first guess useful for further, more refined optimisation. However, the attempt here is to widely explore the solution space rather than to find an accurate solution. For this reason the design of the low-thrust trajectory has not been written either in the optimal control form (with adjoint equations) or in any direct transcription form (collocation or shooting).

It is anyway expected that as the shape of the pseudo-elements approaches the solution of the corresponding optimal control problem the inverse method will yield the associated optimal control for the thrust. For the analysis conducted in this paper the following shape has been used:

$$\boldsymbol{\alpha}(L)=\boldsymbol{\alpha}_0+\boldsymbol{\alpha}_\mathrm{f}(L-L_0)+\mathbf{p}\,\sin(L-L_0), \tag{36}$$

where $\mathbf{p}=[p_1, p_2, p_3, p_4, p_5]^\mathrm{T}$ is a set of parameters shaping each pseudo-element.

The optimality of the solution found can be seen from the comparison with the optimal solution computed for Mars Exobiology. In the optimised solution the trajectory is characterised by two cost arcs and three thrust arcs and a maximum thrust dependent on the distance from the Sun. On the other hand, the first estimate obtained with the inverse approach does not contain any model for power and thrust and no coast arcs are introduced a priori.

Table 6
Domain D for the low-thrust problem

Variable	Lower bound	Upper bound
n	0	2
t_0 (MJD)	2500	3000
T_1 (day)	500	1e3
θ	$-\pi$	π
ϕ	$-\pi$	π
p_1	-0.1	0.1
p_2	-0.1	0.1
p_3	-0.1	0.1
p_4	-0.1	0.1
p_5	-0.1	0.1

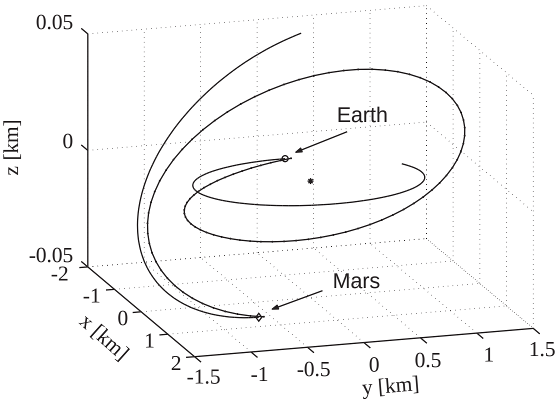

Fig. 9. Mars Exobiology test case.

The optimisation problem then becomes:

$$\min_D -m_\mathrm{f}$$

$$\text{Subject to } t_\mathrm{f}-t_0=0; \quad |\boldsymbol{a}_\mathrm{c}|\leqslant a_\mathrm{max} \tag{37}$$

and the solution vector is:

$$\mathbf{y}=[n,\, t_0,\, T_1,\, \theta,\, \phi,\, p_1,\, p_2,\, p_3,\, p_4,\, p_5]^\mathrm{T}, \tag{38}$$

where n is an integer number representing the number of revolutions, t_0 is the departure date, T_1 the transfer time and θ and ϕ are respectively the inplane and out of-plane (with respect to the Ecliptic) angles of the escape asymptote, defined in a heliocentric interial reference frame. The domain D is specified in Table 6.

The resulting trajectory is represented in Fig. 9 with the associated thrust profile plot in Fig. 10. The main characteristics of the two trajectories are summarised

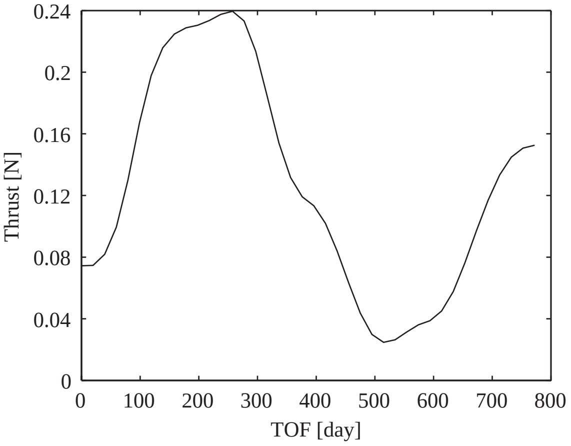

Fig. 10. Thrust profile.

Table 7
Comparison with Mars Exobiology

Sol.	Departure	TOF(day)	m_f/m_0	v_∞ (m/s)
Opt.	18/03/2007	873	0.16	622
FG	09/03/2007	772.1	0.162	622

and compared in Table 7 where FG stands for first guess and represents the solution computed with the inverse approach.

Setting the departure velocity equal to that of Mars Exobiology, the algorithm successfully identified a solution with a low mass consumption comparable with the optimised solution. The transfer time however is quite different due to the selected shape of the orbit. The solution found is expected to have an error in the velocity due to the not exact solution of problem (32). However, this error has been verified to lead, in general, to an equivalent error in propellant consumption which is below 15%. Therefore this solution can be considered acceptable as a first estimate of a possible transfer with low-thrust propulsion, since the error is within the usual margin taken in preliminary mission design.

5.1. Low-thrust transfers with ballistic capture at Mars

The propellant consumption to reach Mars with a low excess velocity, provided by low thrust transfers opens the interesting possibility to exploit lagrangian points of the Mars–Sun system to attempt a low-energy capture in Martian orbit. Maintaining the previous model for low-thrust arcs now the dynamics at arrival is modelled taking into account 3rd body effects. Final conditions at Mars are taken perturbing the state vector at L_1 and propagating backward for a time Δt the following dynamic equations:

Table 8
Low-thrust transfers with ballistic capture

Departure	$a_{\max}.(\mathrm{m/s^2})$	TOF(day)	m_f/m_0	v_∞(km/s)
15/08/2030	2.3e-4	721	0.165	0.2

$$\frac{\mathrm{d}\tilde{\mathbf{x}}}{\mathrm{d}t} = \tilde{\mathbf{F}}(\tilde{\mathbf{x}}, t) = \begin{cases} \tilde{\mathbf{v}} \\ -\frac{\mu_M}{\tilde{r}^3}\,\tilde{\mathbf{r}} - \mu_S\left(\frac{\mathbf{d}}{d^3} + \frac{\mathbf{r}_S}{r_S^3}\right), \end{cases} \tag{39}$$

where μ_S is the gravity constant of the Sun, $\mathbf{r}_S$ is the position vector of the Sun with respect to Mars, $[\tilde{\mathbf{r}}, \tilde{\mathbf{v}}]$ is the state vector of the spacecraft with respect to Mars and $\mathbf{d}$ is the Sun-spacecraft vector.

The solution vector has been extended as follows:

$$\mathbf{y} = [n, t_0, T_1, \Delta t, \delta v_1, \delta v_2, \delta v_3, p_1, p_2, p_3, p_4, p_5]^{\mathrm{T}}, \tag{40}$$

where now δv_1,δv_2 and δv_3 are the three components of the velocity vector at the lagrangian point L_1 defined in the local radial, transversal, normal martian reference frame. The value of the first component belongs to the interval $[-0.12, 0.0]\,\mathrm{km/s}$ while the values of the others belong to the interval $[-0.12, 0.12]\,\mathrm{km/s}$.

The resulting point in deep space at the end of the backward propagation, represents the target of the electric propulsion arc. An example of low-thrust transfer of this kind is reported in Table 8. The arrival at Mars has been plotted in Figs. 11 and 12 for an unpowered and for a powered capture. In the first case no manoeuvres are performed at periapsis and the resulting capture by Mars is only temporary and lasts less than 500 days. In the second case a low-thrust manoeuvre is inserted when the distance from Mars falls down below 1e5 km, the resulting capture is now permanent.

In this case the evolutionary-branching algorithm was not applied to characterise the solution space.

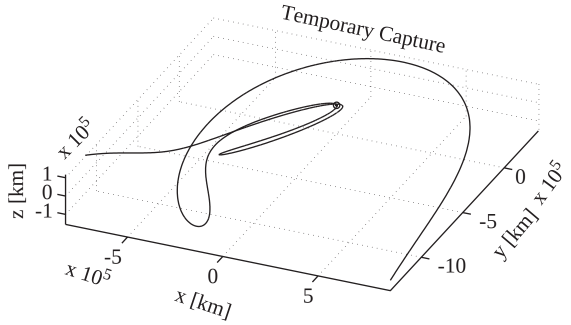

Fig. 11. Temporary capture.

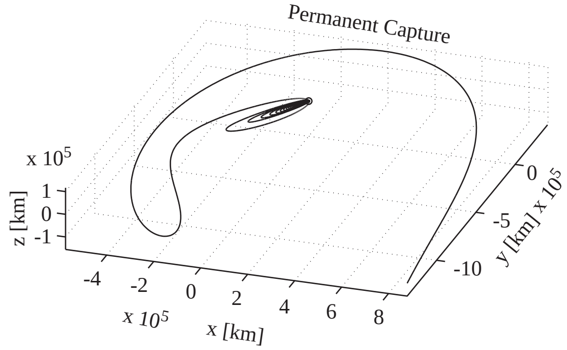

Fig. 12. Permanent capture with low-thrust manoeuvre at periares.

A first interval was computed running evolutionary-branching with the model described in the previous chapter (i.e. without 3rd body effects) then just the evolution step was run, including 3rd body effects in the model and imposing a minimum of 50 days for Δt.

This first result is encouraging and suggests further investigations in this direction.

6. Conclusions

In this paper a combined deterministic-stochastic approach is proposed to solve trajectory design problems in which more than one solution is expected and where not just the global optimum should be obtained. The proposed combination of evolutionary algorithms and branching is suitable for problems characterised by differentiable and non-differentiable functions combining integer and real variables, and demonstrated to be an interesting tool for preliminary mission analysis especially when the objective function is a black-box. In fact, in this respect an ad hoc systematic approach specifically dedicated to solve a certain problem is expected to be more efficient.

The capabilities of this approach have been demonstrated by solving the complex problem of identifying all optimal solutions for a Mars roundtrip in a given time frame. This first analysis has revealed that free return trajectories are always available for each launch window and can be classified in three major groups depending on the length of each transfer leg. Although all of them present the significant drawback of having high velocity either at Earth or at Mars they could represent an option for high specific impulse engines or as abort options. Among nominal transfers in the time frame 2028, 2037 the 2033 launch window seems to offer interesting features since the transfer time for both legs is relatively low with an associated low total Δv and a low cost return via Venus is possible for a short stay. For nominal transfers the analysis of optimal staging sequences has shown how the optimal orbit for departure from Earth is elliptical but with the apocentre almost at the altitude of a geostationary orbit while for Mars the apocentre is much closer to the sphere of influence. Finally, the last analysis presented has opened the interesting possibility to use low-thrust transfers for low-energy planetary capture at Mars. However, this problem and the inverse approach used to design low-thrust arcs, are the subjects of an ongoing more detailed analysis and therefore, the results presented in this paper must be considered preliminary.

The proposed optimisation algorithm appears to be promising for generally complex space trajectory design problems. Despite the effective exploration capabilities demonstrated in the cases presented in this paper, the efficiency of the method has still large margins for improvement and the convergence to the global optimum is still not guaranteed if not in a probabilistic sense or after an infinite number of subdivisions of the branching step. In this respect the proposed use of interval analysis, when actually viable, represents a promising way of guaranteeing and controlling convergence.

An improved version of the algorithm is already under development and will be presented in future works.

Acknowledgements

The authors are grateful to Guy Janin of ESOC/ESA for Mars Exobiology trajectory with electric propulsion.

References

[1] P.J. Gage, R.D. Braun, I.M. Kroo, Interplanetary trajectory optimisation using a genetic algorithm, Journal of the Astronautical Sciences 43 (1) (1995) 59–75.

[2] G. Rauwolf, V. Coverstone-Carroll, Near-optimal low-thrust orbit transfers generated by a genetic algorithm, Journal of Spacecraft and Rockets 33 (6) (1996) 859–862.

[3] V. Coverstone-Carroll, J.W. Hartmann, W.J. Mason, Optimal multi-objective low-thrust spacecraft trajectories, Computer Methods in Applied Mechanics and Engineering 186 (2000) 387–402.

[4] M. Vasile, G. Comoretto, A.E. Finzi, A combination of evolution programming and SQP for WSB transfer optimisation, AIRO2000, Milano, Italy, September 18–21, 2000.

[5] P. Gurfil, N.J. Kasdin, Niching genetic algorithms-based characterization of geocentric orbits in the 3D elliptic restricted three-body problem, Computer Methods in Applied Mechanics and Engineering 191 (49–50) (2002) 5673–5696.

[6] Z. Michaelewicz, Genetic Algorithms + Data Structures = Evolution Programs, third ed., Springer, Telos, 1996.

[7] J. Horn, The nature of niching: genetic algorithms and the evolution of optimal, cooperative populations, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1997.

[8] C.R. Houck, J. Joines, M. Kay, A genetic algorithm for function optimization: a Matlab implementation, ACM Transactions on Mathematical Software, 1996.

[9] C. Bliek, P. Spellucci, L.N. Vicente, A. Neumaier, L. Granvilliers, E. Huens, P. Van Hentenryck, D. Sam-Haroud, B. Faltings, COCONUT deliverable D1. Algorithms for solving nonlinear constrained and optimisation problems: the state of the art, The Coconut Project, June 8, 2001.

[10] D.R. Johns, C.D. Perttunen, B.E. Stuckman, Lipschitzian optimization without the lipschitz constant, Journal of Optimization Theory and Applications 79 (1993) 157–181.

[11] J.D. Pintér, Global Optimization in Action, Kluwer, Dordrecht, 1996.

[12] C.S. Adjiman, S. Dallwing, C.A. Floudas, A. Neumaier, A global optimization method, alphaBB, for general twice-differentiable constrained NLPs-I. Theoretical advances, Computers and Chemical Engineering 22 (1998) 1137–1158.

[13] B. Gardini, F. Ongaro, ESA Aurora exploration programme, 53rd IAC Houston 10–19 October 2002, IAC-02-IAA.13.1.02.

[14] E. Hansen, Global Optimization Using Interval Analysis, Marcel Dekker Inc., New York, 1992.

[15] A. Törn, A. Zilinskas, Global Optimization, Lecture Notes in Computer Science, Vol. 350, Springer, Heidelberg, 1989, 255pp.

[16] M. Vasile, A global approach to optimal space trajectory design, AAS-03-141,13th AAS/AIAA Space Flight Mechanics Meeting, Puerto Rico, 9–13 February 2003.

[17] R.H. Battin, An Introduction to the Mathematics and Methods of Astrodynamics, AIAA Education Series, Vol. XXXII, AIAA Inc., New York, 1987.

[18] M.R. Patel, J.M. Longuski, J.A. Sims, Mars free return trajectories, Journal of Spacecraft and Rockets 35 (3) (1998).

[19] M. Okustu, J.M. Longuski, Mars free return via gravity assist from Venus, Journal of Spacecraft and Rockets 39 (1) (2002).