Paper AAS 03-141

A GLOBAL APPROACH TO OPTIMAL SPACE TRAJECTORY DESIGN

Massimiliano Vasile
European Space Research & Technology Centre (ESA/ESTEC), The Netherlands

13th AAS/AIAA Space Flight Mechanics Meeting

PONCE, PUERTO RICO **9-13 FEBRUARY 2003**

AAS Publications Office, P.O. Box 28130, San Diego, CA 92198

A GLOBAL APPROACH TO OPTIMAL SPACE TRAJECOTRY DESIGN

Massimiliano Vasile
European Space Research & Technology Centre
(ESA/ESTEC),
The Netherlands

Abstract

In this paper a combination of Evolution Programming and Branching is used to solve some typical problems in space trajectory design: finding a number of possible minimum cost transfers, including the global one, between two celestial bodies. The idea is to use a limited population evolving for a small number of generations, according to some specific evolution rules, in subregions of the solution space defined by a branching procedure. On the other hand the branching rules are functions of the outcome from the local evolution optimization. The proposed combined systematic-heuristic global optimization performs quite well on the analyzed cases suggesting the possibility of more complex application in space trajectory design.

INTRODUCTION

A meaningful part of the mission design process consists of designing the trajectory. Traditionally this task has been accomplished using gradient methods, optimal control theory or mathematical tool specifically dedicated to each particular problem. Anyway all this approaches can be generally classified as local optimization methods where the term optimization is intended not just for finding the minimum or the maximum but more in general for finding a solution. Since the problem is generally highly non-linear and not necessary differentiable in all the solution space, a significant part of the job is to formulate appropriately the problem to make it amenable to a solution using local optimization tools and to produce a reasonably good initial guess. In fact it is likely that, despite the global convergence properties of many software tools (like most of NLP solvers available at present), the analysts find a local minimum every time they seek for a solution, eventually finding the global optimum. This is quite a time consuming process that can lead even to a non-satisfactory result. Furthermore due to the relatively poor robustness of some approaches (like indirect methods) a good initial guess is often quite hard to find. Finally a complete and sophisticated formulation of the problem can generally include discontinuities or integer variables or non-differentiable functions impossible to handle with standard gradient methods.

The importance of having an effective and efficient global optimization approach (well known, understood and studied in many other fields like chemistry, biology and electronic engineering[1]) is recently emerging even in the space field with studies on procedure and optimization methods to procure a solution or even just a first guess solution to complex problems as WSB (Weak Stability Boundaries)[2] transfer design, the design of trajectories involving multiple swing-bys or a combination of swing-bys and low thrust propulsion[3,4].

Most of global optimization techniques can be classified in two main groups: systematic and stochastic or heuristic approaches. Systematic methods, like branch and bound approaches, are guaranteed (in exact arithmetic) to find the global optimum with a predictable amount of work. On the other hand heuristic method, like evolution programming, cannot be proved to find the global optimum with a predictable amount of work.

In this paper a mixed approach combining a breaching technique and a particular implementation of evolution programming[5] is proposed to solve some space trajectory design problems. This particular combination presents some novel ideas: a migration operator that guides individuals toward promising areas of the solution space, a filter operator (in place of common selection operators) ranking families of potentially interesting individuals and a particular tunneling[6] technique used to find the global optimum. Moreover EP is used to obtain lower bound information, to select promising branches and to prune non-promising ones. Furthermore the algorithm treats both integer and real variables.

The effectiveness of the proposed algorithm (EPIC) is demonstrated on some typical problems in space mission design.

OPTIMISATION PROBLEM

Optimization problems in trajectory design can be either unconstrained or constrained; in the former case they can be written as:

$$\begin{aligned} &\min f(\mathbf{y}) \\ &\text{with} \quad \mathbf{y} \in D \end{aligned} \tag{1}$$

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 $\mathbf{y}$:

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

In the latter case the problem can be written as:

$$\begin{aligned} &\min f(\mathbf{y}) \\ &\mathbf{b}^l \le \mathbf{C}(\mathbf{y}) \le \mathbf{b}^u \\ &\text{with} \quad \mathbf{y} \in D \end{aligned} \tag{3}$$

Where $\mathbf{C}(\mathbf{y})$ is vector of nonlinear constraint functions of the vector $\mathbf{y}$. If problems (1) and (3) are twice continuously differentiable and present a single solution, i.e. only one vector $\mathbf{y}$ in the domain D minimizes 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 can not contain non-differentiable quantities. However even in this case the problem may present more than one solution within the required domain 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 optimization method must be considered. The idea is to perform an extensive search of the solution space D looking for possible solutions to problems (1) and (3). In this respect the interest could be more to find a number of good initial guesses for the nonlinear programming solver than finding the global optimum with the required accuracy.

Methods for global optimization can be generally classified in three categories: approximation methods, heuristic methods and systematic methods.

Approximation methods transform the original problem by means of suitable approximations into a simpler global optimization problem that is more tractable. Once a solution for the approximated problem is found a local optimization method con be used.

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 of methods. For them it is sometimes possible to prove convergence with probability arbitrarily close to 1 but with a number arbitrarily big of function evaluations. The simplest heuristic method is multiple random start, consisting of picking random starting points and performing local optimizations from these points. Most heuristics can be regarded as techniques devised to speed up this process of local/global search by picking the points more carefully.

Systematic methods contain all methods that (in exact arithmetic) are guaranteed to find the global optimum with a predictable 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 where all points on finer and finer grids are tested and the best point on each grid is used as starting point for local optimization. The number of grid points grows exponentially with the dimensions of the problem and so does the amount of work. More efficient systematic methods generally combine

branching techniques with one or several local optimization procedure, convex analysis, interval analysis and constraint logic. Even though systematic methods are generally more reliable then heuristics they need, to be efficient, some level of insight into the problem and the structure of the objective function. If the problem is represented by a black box then they may not find the global optimum in a reasonable amount of time.

A well known stochastic method is represented bye Genetic Algorithms 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 code a solution 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. More sophisticated genetic algorithms make use of the data structure of the problem to code the individual in the more appropriate way. For example if the problem contains only floating point variables a floating point coding is more accurate and efficient than a binary coding. In general all methods that resort to some heuristics concepts derived from biological evolution can be defined evolution programming methods.

Among systematic methods there are some that split the solution domain on the base of some local information. Each time the domain is split, a number of new branches are created, each branch corresponds to a further exploration of the solution space and each subdomain represents a node that can be expanded and explored further. Therefore a branching scheme generates a sequence of rooted trees of boxes whose leaves cover the feasible set. If the diameter of all the boxes and all leaves converge to zero, convergence of the algorithm is straightforward.

Proposed optimization approach is composed of a stochastic step and of a systematic step. The stochastic step is performed using evolution programming and is meant to obtain information on the possible presence of optima in a subdomain $D_l \subseteq D$. On the other hand 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 and the qualification of each subdomain on the base of the best solution found and the volume of the subdomain, allows to find a number of optima and eventually the global one.

EVOLUTION PROGRAMMING

Present implementation of evolution programming is based on four fundamental operators: mutation, migration, mating and filtering. It should be noticed that all of them operate both on real and integer numbers therefore each individual, represented by a vector **y,** contains in the first m components integer values and in the remaining s components real values.

Mutation. Mutation operates in three different ways: generates a random number, taken from a gaussian distribution, within the domain D or within each subdomain D_l, for each component of **y**; generates a symmetric perturbation of a selected component y_i with respect to its original value within an interval in a neighborhood of **y**; generates an asymmetric perturbation of a selected component y_i with respect to its original value with in an interval in a neighborhood of **y**. A third mutation scheme exchange a random componet of the individual with one of the bounds.

Mating. 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 which simply exchange part of the genes between the two parents
- Blending, also known as arithmetic crossover, which generates a new individual with an interpolation of the two parents: chosen two individuals $\mathbf{y}^1$ and $\mathbf{y}^2$ and a random number α, the resulting child will be given by:

$$\mathbf{y}^3 = \alpha \mathbf{y}^2 + (1-\alpha)\mathbf{y}^1 \tag{4}$$

- Extrapolation generates a new individual on the side of the best individual between the two parents at a distance from the best parents equal to the vector connecting the two parents:

$$\mathbf{y}^3 = \alpha(\mathbf{y}^2 - \mathbf{y}^1) + \mathbf{y}^2 \tag{5}$$

- Second order extrapolation mating generates a child using two parents and the child generated with an extrapolation mating. If **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 Figs. 1 and 2):

$$\mathbf{y}^4 = \mathbf{y}^1 + \mathbf{p}\chi_{\min} \tag{6}$$

$$f_{\min} = a(\mathbf{y}^1,\mathbf{y}^2,\mathbf{y}^3)\chi^2_{\min} + b(\mathbf{y}^1,\mathbf{y}^2,\mathbf{y}^3)\chi_{\min} + f(\mathbf{y}^1) \tag{7}$$

The mating operator is used also to prevent an undesirable effect of migrations: if more than one principal individual is in the basin of attraction of the same solution, it is likely that all of them will move toward the same point with a resulting waste of resources. Therefore if two or more principal individuals are colliding (intersecting their migration regions) a repelling mechanism is activated which mates the worse individual (between two colliding) with the boundaries or 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:

$$y_i^2 = \alpha b_i + (1-\alpha)y_i^1 \tag{8}$$

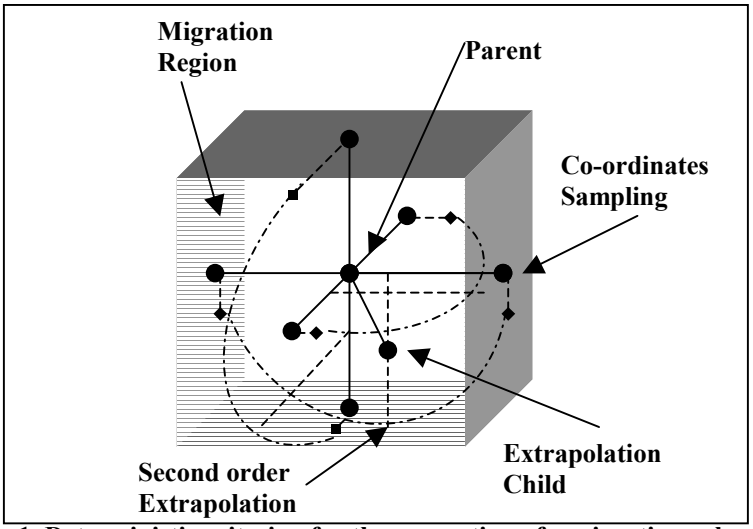

Figure 1. Deterministic criterion for the generation of a migrating subpopulation

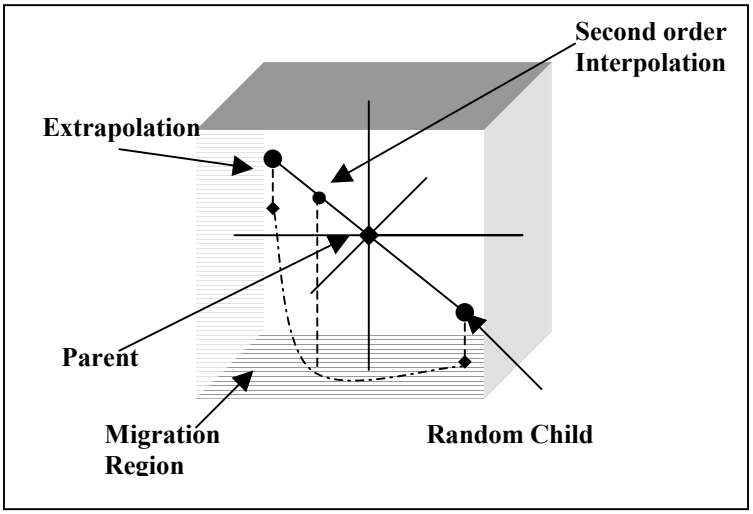

Figure 2. Stochastic criterion for the generation of a migrating subpopulation

Migration

The migration process generates a subpopulation in a neighborhood of an individual **y** the best individual of the subpopulation, if better than the parent, survives to the next generation instead of the parent. The subpopulation is generated using either a stochastic or a deterministic procedure. The deterministic procedure generates a child along the coordinates on the boundary of the migration region. Then an extrapolation mating produces a second child toward the most promising boundary. If the second and the first children are not the same a third one is generated using a quadratic interpolation of the two previous children and the parent. The same scheme is repeated for all the coordinates and all the children generated with the quadratic interpolation are then collected and used to generate an extrapolation child whose components are the components of all the second order children (see Fig.1).

The stochastic procedure samples the migration region generating a child randomly then an extrapolation mating is performed on the side of the best individual. The two resulting children and the parents are then used to build a quadratic model and a third child is generated taking the minimum of the parabola. The procedure is repeated until a number of children equal to the number of coordinates has been generated (see Fig. 2).

The micro population is generated within a migration region in a neighborhood of a principal individual; the migration region is a hyper parallelepiped $\mathbf{S}=S_1xS_2...xS_n \subseteq D_l$, where S_i is a subinterval containing the value of the component y_i . Each subinterval S_i is asymmetric, allowing the migration to depart from one of the boundaries. Furthermore the migration region **S** contracts or expands according to a migration radius ρ whose value depends on the differential improvement of each individual from one generation to another.

The migration radius

If f_k^j is the fitness value associated to a an individual j at generation k and f_{k+1}^j the fitness value associated to the same individual at generation $k+1$, we define the differential improvement as:

$$df^j = f_{k+1}^j - f_k^j \tag{9}$$

The migration radius is defined as the ratio between the value of the bounds of the migration region of the j-th individual $\mathbf{b}^j$ and the value of the bounds **b** of the domain D:

$$\rho^j = \frac{b_i^j}{b_i} \tag{10}$$

During convergence the migration radius is reduced or enlarged depending on the fitness of the individuals inside the migration region and the differential improvement of the principal individual from one generation to another. If none of the children of the subpopulation is better than the parent the radius is reduced. Different from a previous implementation[9], where the migration radius was a function of the differential improvement, in this implementation a slower contraction of the migration region has been implemented computing the radius according to:

$$\rho^j = \begin{cases} \max([1e-8, \delta y_{\min}]) & \text{if } \delta y_{\min} \geq \varepsilon\rho^j \\ \varepsilon\rho^j & \text{if } \delta y_{\min} < \varepsilon\rho^j \end{cases} \tag{11}$$

if from generation k to generation k+1 the differential improvement increases, then the migration radius is recomputed according to the prediction:

$$\rho_{k+1}^j = \rho_k^j \eta \log(e-1+j) \tag{12}$$

where η is equal to 2.1 in this implementation and ρ_{min} has been set to 1e-5. It should be noticed that the value of ρ^j_{k+2} depends on the ranking of the associated individual: the migration radius of a bad individual is therefore larger than the best individual allowing a bigger mutation. In fact if a principal individual is converging to a solution it is not desirable to move it too far from its position.

For integer numbers migration operates in the same way but the migration regions and migration radius are generated and treated differently. In particular ρ_{imin} is 1 and ρ is defined as:

$$\rho_{k+2}^j = \min[\text{int}(\log(2+j)\Delta f_{mig}{}^j), \rho_{i\min}] \tag{13}$$

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

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 base of their fitness from the best to the worst. All the individuals with fitness lower than a given threshold are completely mutated while migration is applied to all individuals within the filter. This allows each of the individuals within the filter to evolve toward a different local optimum. The filter basically operates a simple sorting procedure but, since individuals in the upper part of the filter are strongly mutated, it is likely that they are replaced by quite different new individuals coming from the associated subpopulation and migration.

Not only does the position in the filter determines the level of mutation of the individual but also the mating process is influenced by the ranking. In particular individuals out of the filter are totally mutated at every generation using two operators: a random mutation and a boundary mutation. On the other hand mating is operated on all the individual out of the filter combining them with individuals in the filter. After mating the resulting children survive according to their predicted position in the filter.

Therefore in the pure EP step migration is used to explore locally the solution space and two mechanisms are used for global exploration: mating and mutation of individuals outside the filter. It should be noticed that if several minima are clustered the mixed systematic-stochastic generation of subpopulation should guarantee anyway to find locally the best minimum of the cluster.

COMBINED EVOLUTION BRANCHING

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 evolution algorithm. The domain D is partitioned generating a number of subdomains. Each subdomain is qualified ad explored further according to its qualification. The partitioning, or branching, process begins taking the worst individual, which is out of the filter, and cutting D into q subdomains, corresponding to q new branches (or nodes). Each node may or may not contain an individual coming from the previous step of evolution and the relative volume of the node depends on the position of the cutting point. For each node the ratio between the relative number of individuals and the relative volume is computed and the resulting quantity defines how necessary is a further exploration of the node:

$$\varpi_{D_l} = (1-\upsilon)\frac{\sum_{D_l} j}{\sum_{D} j} \Bigg/ \sqrt[n]{\frac{V_{D_l}}{V_D}} + \upsilon\varphi_{D_l} \tag{14}$$

this quantity is then added to a fitness φ defined as:

$$\varphi_{D_l} = \begin{cases} \dfrac{\frac{1}{J}\sum_{j=1}^{J} f_j - f_{best}}{f_{worst} - f_{best}} & if\ i \neq 0 \\ 1 & otherwise \end{cases} \tag{15}$$

where J is the number of individuals in subdomain D_l. The node is then qualified by the quantity:

$$q_n = (1-\upsilon)\varpi_{D_l} + \upsilon\varphi_{D_l} \tag{16}$$

where ν is weighting factor that weights how reliable the result coming from the evolution step is considered. If ν is 1 only the nodes with low fitness are explored because the EP algorithm is considered reliable enough to explore exhaustively the domain D without leaving any region unexplored, on the other hand if ν is 0 the result from the EP algorithm is considered to be not reliable due to a premature convergence or to a pure exploration of the solution space. It is clear in fact that a node with a large number of individuals with high fitness (at the top of the ranking scale) has a high probability to contain the global optimum if the EP has explored exhaustively the solution space, on the other hand if the volume of the node is large and the number of individuals low it is likely that the EP step was not able to evaluate sufficiently the solutions pace in this region and therefore the node need still to be explored in the future.

It should be noticed however that, if the EP have converged in a given subdomain, nodes not containing any individual, even though they have a large volume, are unlikely to contain the solution. For a fast search, therefore, only nodes presenting high fitness and large volume are explored further.

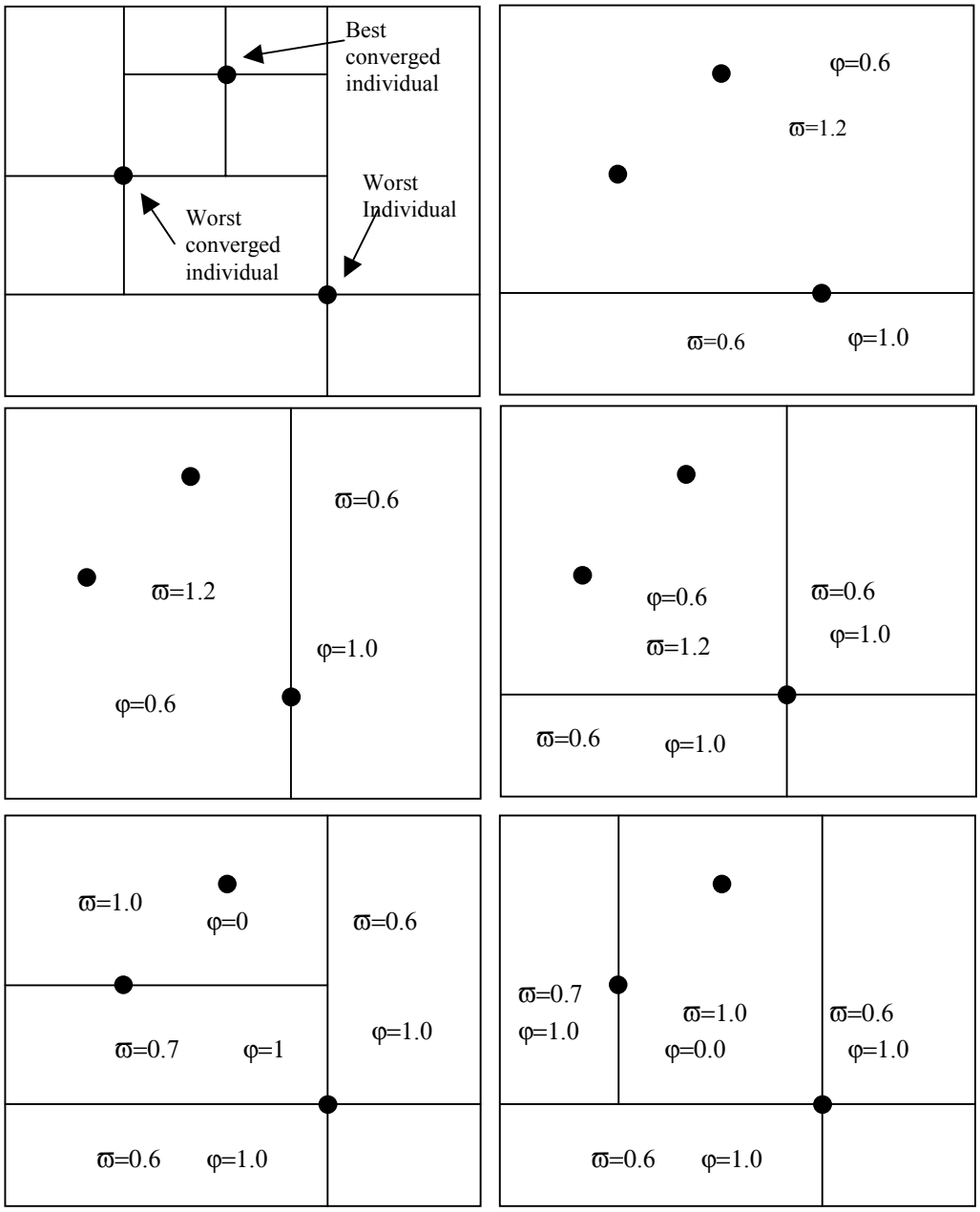

Figure 3. Sketch of the branching procedure

The process is quite effective to explore the entire solution space in great detail but produces often, unnecessary reevaluation of many regions where a local minimum has already been found. The result is a rediscovery of local minima in subdomains getting progressively smaller and smaller with a waste of computational resources. In order to avoid this phenomenon 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 and therefore to cut the solution space, converged individuals suggest where a further exploration is unnecessary. 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. A cartoon of the multi-partition procedure is depicted in figure 3.

In addition the cutting individuals sets an upper bound on the possible value of the objective function therefore all future individuals with a higher value of the objective function are rejected. For all other individuals the cutting one works as a repeller forcing a migration toward more promising areas of the solution space. Repulsion is performed through a tunneling operator. The basic idea is that all the regions of the solution space below certain fitness are flooded and all the individuals that want to survive must migrate toward a dry land. The shape of the territory is changed due to the flooding according to what we called hydrophobic tunneling and the individual, setting the flooding level, works as a repeller.

$$\psi_1 = k\left[1 - e^{-\gamma(f - f_0)}\right] \tag{17}$$

$$\psi_2 = \frac{\sigma}{\left\|\mathbf{r} - \mathbf{r}_0\right\|^{\tau}} \tag{18}$$

Stopping Criterions

There are two combined stopping criterions: one for local convergence and one for global convergence. Both are based on some heuristics and not on any rigorous prove of global converge. Local convergence of each subpopulation is determined by the differential improvement of the principal individual and by the migration radius. In a convex problem both should tend to zero in a neighborhood of the solution. Since each principal 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 EP 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 by the convergence of the best individual otherwise. It must be noticed that when EP are 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 part is based on two ideas: the dimensions of each node and the convergence of EP in each subdomain. If a node reduces below a given tolerance it is discarded and considered converged, therefore if no nodes are left, the algorithm stops, on the other hand if EP have converged in all subdomains and no improvement is reported after branching, i.e. no new local minima are discovered, the algorithm stops since it is likely that all local minima have been already found and no further exploration of the solution space is required.

TEST CASES

Optimal Launch Window Problem

In an ecliptic reference frame centered into the Sun and considering the gravity action of the Sun only, the dynamic of a spacecraft is governed by the following differential system:

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= -\frac{\mu}{r^3}\mathbf{r} \end{aligned} \tag{19}$$

where μ is the gravity constant of the Sun, **r** is the position vector of the spacecraft and **v** is its velocity vector. Now in the hypothesis of Keplerian motion taking two points in space and fixed a time of flight (TOF) T, Lambert's problem consists of finding the transfer arc from one point to the other in the given time. If this is applied to the problem of finding the optimal transfer trajectory from Earth to Mars, an infinite number of trajectories can be generated, each one characterized by a different departure date from the Earth t_0, a different time of flight T and a different departure velocity Δv_E from the Earth and arrival velocity Δv_M at Mars. The arrival and departure velocities can be related to the cost in terms of propellant to transfer a spacecraft from the Earth to Mars, therefore the following objective function can be defined:

$$f = \Delta v_{\ E} + \Delta v_{\ M} \tag{20}$$

which must be minimized with respect to the departure time and transfer time.

If f is plotted with respect to t_0 and T the result can be seen in figures 4 and 5. If an upper limit is imposed on the maximum total Δv allowed for an interplanetary mission, the contour plot 5 shows only regions characterized by a total Δv lower than the require limit. These regions define what are generally called launch windows, i.e. intervals of possible launch dates. For the problem under study t_0 is defined in the interval [3000, 6000] expressed in Modified Julian Day (i.e. number of days from 1st January 2000) while the TOF is defined in the interval [100,400] expressed in days. In the given domain D of launch dates and transfer times, there are at least 8 local minima but actually only one is global with a value of 5.667 km/s. However a second minimum can be found with a value slightly different 5.699 km/s but for a completely different launch date.

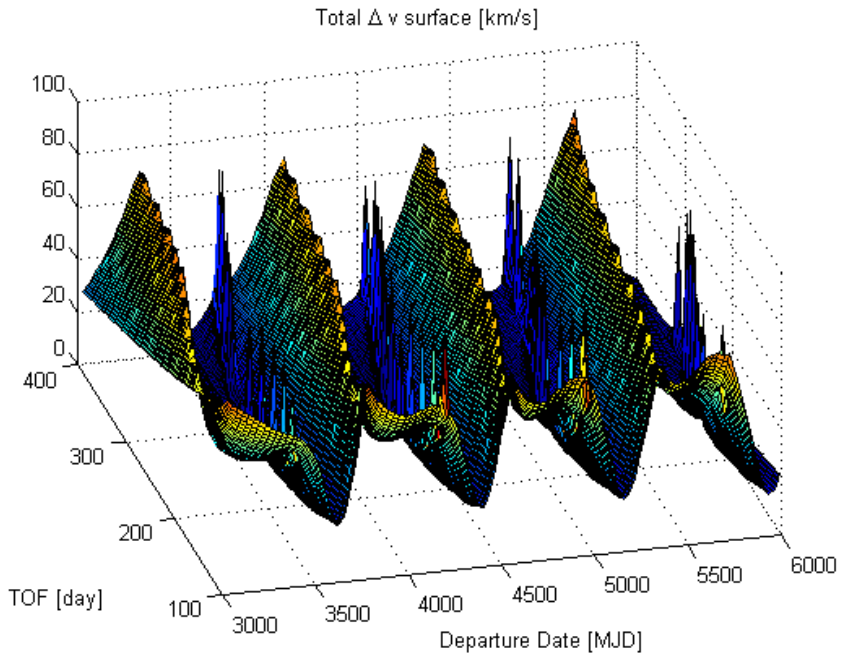

Figure 4 Three-dimensional plot of the total Δv problem: the objective function is the sum of the Δv required to leave the Earth and the Δv required to insert a spacecraft into Mars orbit.

Here an example of the results obtained with the combination of EP and branching is reported. At first only the EP algorithm is tested to verify the effectiveness and efficiency of the new operators. The problem is solved with no branching step running the EP several times and checking the obtained group of minima. The stopping criterion in this case is not the complete convergence of the filter but just of the best individual. A steady population of 10 individuals has been used with a filter containing a maximum of 7 individuals: the individuals outside the filter are therefore strongly mutated. A tolerance of 1e-4 on *df* and a tolerance of 1e-4 on the migration radius have been used for the stopping criterion. Since the nature of the method is stochastic, 20 runs have been performed and the resulting number of function evaluations is the average of all 20 runs. It should be noticed that only three, out of 20, converged to the second better minimum without being captured by the basin of attraction of the best minimum. All the others have the global minimum in the first three positions of the filter and among them, ten have the global minimum as first value.

The accuracy of the outcome form the EP step has been verified with an SQP refinement of all the solutions. An example of a typical run is plotted in Figure 6 and the result is reported in table 1. Notices that the algorithm successfully found 5 minima (among them the global one in the upper left part of the plot) signed with a fat dot, the other individuals, represented by crosses, are values rejected by the filter. It is clear form this example that some minima could not be taken into account by the algorithm and among them there

could be the global one especially if, as in this case, more than one minima have similar values with similar basins of attraction.

As further demonstration of the effectiveness of the algorithm the same problem has been solved using genetic algorithms, with a floating point data structure and encoding, and a pure Branching methods with a Pareto principle for box selection. implemented in the code DIRECT[7]. DIRECT splits at each step all boxes for wich the pair (volume, midpoint value) are not dominated by other such pair. Here *(v,f)* is dominated by *(v',f')* if both $v'<v$ and $f'<f$. In particular, the box of larget volume and the box with the best function value are never dominated and hence always split.

The Genetic Algorithms implementation used here (and freely available in the matlab toolbox GAOT[8]) uses a floating point representation of the genotype. The following genetic operators have been used on a population of 20 individuals: boundaryMutation multiNonUnifMutation nonUnifMutation unifMutation, arithXover heuristicXover simpleXover. A maximum of 500 generations are allowed and total number of function evaluations are computed up to convergence over 20 runs.

In the branching algorithm the number of iterations determines the stopping criterion and the total number of function evaluations. Depending on the initial domain *D* and therefore on the initial sampling of the solution space the number of iterations to reach a given tolerance changes dramatically.

The result of the comparison among the three approaches is summarised in table 1 where the actual global optimum is reported along with the results coming from the three optimisations. Next to the best value obtained for the objective function the percentage of times, over 20 runs, the algorithm was able to find the global optimum is reported. As can be seen the systematic approach is always able to converge to the global minimum with the required accuracy, however the number of function evaluations strongly depends on the domain, i.e. on the initial sampling of the solution space done by DIRECT.

The proposed implementation of evolution algorithms performs quite well, better then both DIRET and GAOT, thanks to the filter and to the recombination and migration operator. Even though, as stated before, the global minimum is always in the filter at the end of the optimisation it is not true that the algorithm is able to recognise it all the times.

Table 1. Pure Evolution step with convergence of the best individual

Value	Global Optimum	GAOT	DIRECT	Epic
J (km/s)	5.6673	5.6673 (40%)	5.6674 (100%)	5.6673 (75%)
t_0 (MJD)	3573.7	3573.5	3573.9	3573.5
TOF (day)	324.05	324.11	323.85	324.34
Function Evaluations	-	1374	(90-)1183	351

The second test includes branching and was used to verify the effectiveness of the branching criterion and to improve the exploration of the solution space.

The first run of EP spans the entire domain finding a number of minima. Some regions of the solution space result however unexplored since the choice of the initial population and of regenerated principal individuals is basically a random process, furthermore it might happen that even though one individual is initially in the attraction basin of a minimum the filter reject the individual, putting it at the bottom of the list. This happens especially when some other individuals are close to convergence. Thus some regions result to be poorly explored because all principal individuals generated do not survive enough to converge toward a local minimum. Figure 7 reports the result of a branching step from a run of the combined systematic-stochastic algorithm. The first cutting point is the worst of the individuals rejected by the filter, this ensures that the resulting branches correspond to either unexplored regions or regions containing some already found minima. Branches containing converged individuals are correctly partitioned using these individuals, and the resulting nodes with a high volume and low density as well as branches with high fitness are evaluated further. After this branching step, however the algorithm stopped declaring convergence since no improvement was found. Using this technique over other 20 runs, the algorithm was always able to find the global optimum plus all the other 7 optima. A summary of the obtained minima for the case represented in Figure 7 is reported in Table 2 where the values found by the evolution branching algorithm (EPIC) are compared to the values computed refining each solution with a SQP algorithm.

Table 2. Summary of minima found by the evolution branching algorithm

Sol.	1	2[(b)]	3	4	5	6	7	8
SQP	3573.7	4330.3	4340.0	3598.82	5088.3	5860.12	5909.5	5123.1
	324.0	306.63	252.11	76.6	295.1	77.06	201.41	221.28
Epic	3573.43	4332.43	4347.4	3599.52	5084.3	5864.9	5909.5	5125.9
	24.1	08.85	254.93	77.58	295.5	292.58	201.82	223.5

(b) This solution is a minimum for the subdomain but it is actually in the basin of attraction of sol 1

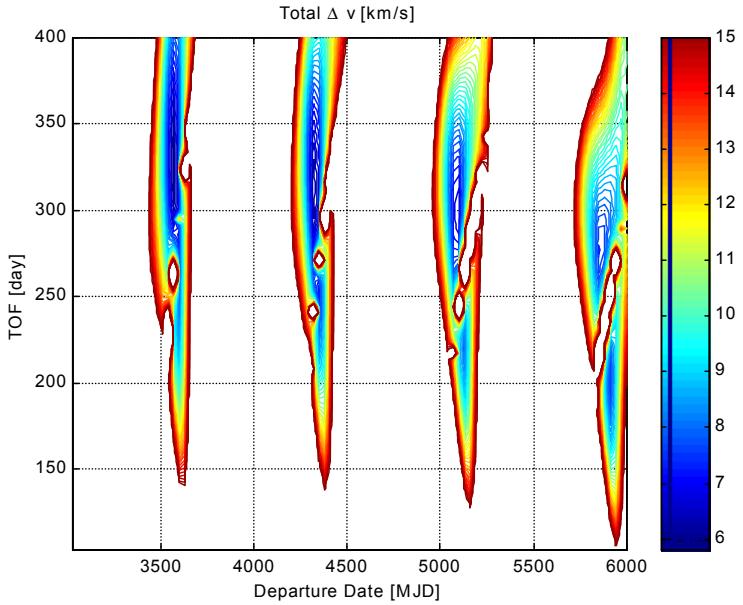

Figure 5 Contour plot of the total Δv problem: blue region have low Δv, red regions have high Δv while white areas between two launch windows have excessive cost higher than 15 km/s.

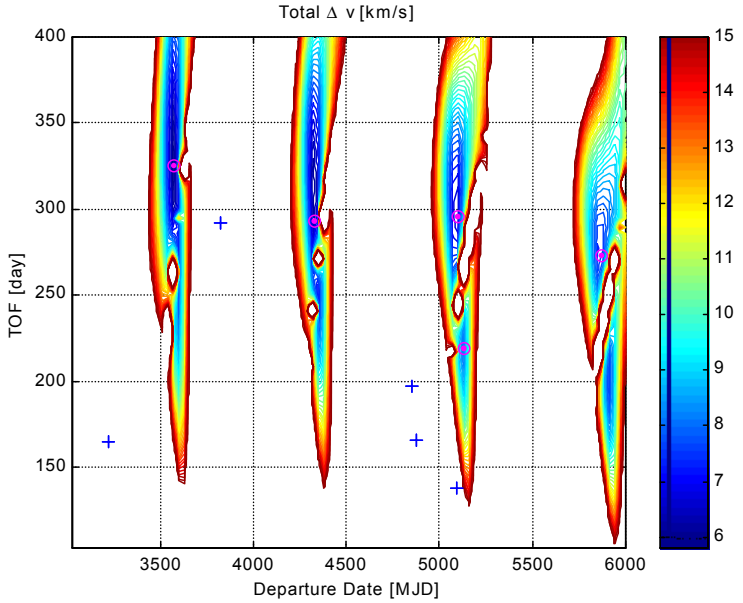

Figure 6 Result from a pure EP step: magenta fat dots are optimal solutions accepted by the filter while blue crosses are individuals rejected by the filter.

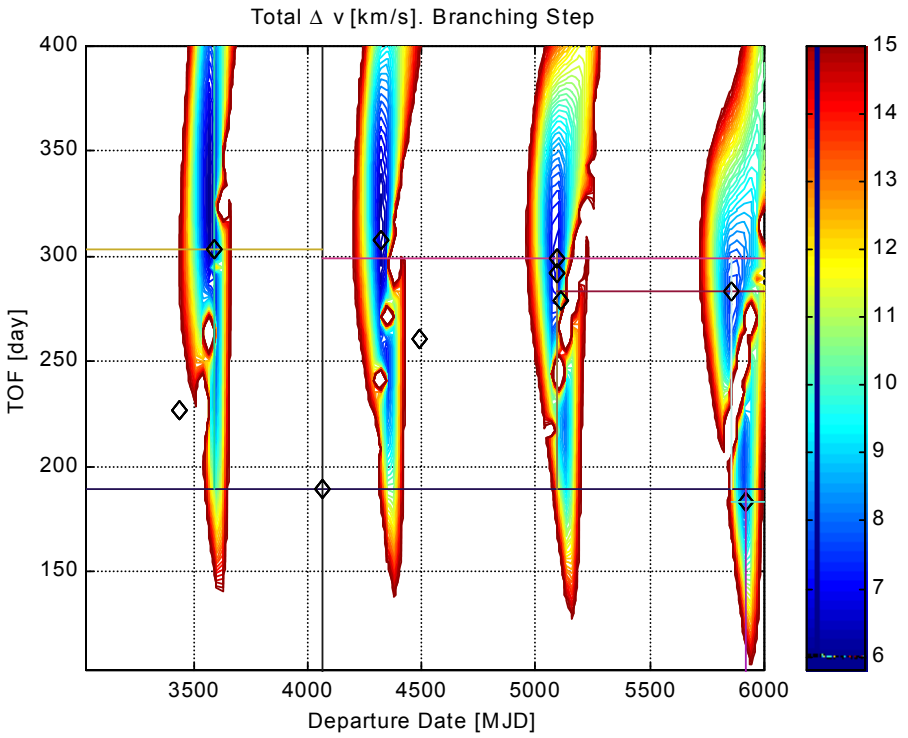

Figure 7 Branching Step

Optimal Earth-Mars Transfer with Electric Propulsion

In this second problem the transfer from the Earth to Mars is performed using an electric propulsion system. The departure C_3 is zero and the objective function is represented by the propellant mass which must be minimized. The engine is continuously on and the thrust direction is defined by a simple shaping law:

$$\xi = \begin{cases} \text{a}\tan(\theta + u_1) + u_2 & \text{if } \xi \le \pi/2 \\ \text{a}\tan(\theta + u_1) + u_2 - \pi & \text{otherwise} \end{cases}$$
$$\lambda = abs(\text{a}\tan(w(\theta - \pi/2 + u_1))) - \pi/2 \tag{21}$$

the dynamic of the spacecraft is therefore governed by the following system of equations:

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= -\frac{\mu}{r^3}\mathbf{r} + \frac{T}{m}\zeta \\ \dot{m} &= -\frac{T}{g_0 I_{sp}} \end{aligned} \tag{22}$$

where T is the maximum thrust available and it was assumed to be 0.3N, while the I_{sp} has been taken equal to 1700s and the initial mass equal to 2000kg .The control vector ζ is defined as:

$$\zeta = \Omega \begin{bmatrix} \cos\xi\sin\lambda \\ \sin\xi\sin\lambda \\ \cos\lambda \end{bmatrix} \tag{23}$$

where Ω is the rotation matrix from the local along-track, cross-track, normal reference system attached to the spacecraft to the three dimensional Cartesian inertial reference frame.
Now if each individual is coded as a real value vector containing the departure date t_0, the transfer time P

and the two parameters characterizing the shape of the control:

$$\mathbf{y} = [t_0, P, u_1, u_2]^T \tag{24}$$

the optimization problem results to be:

$$\begin{aligned} &\min f(\mathbf{y}) = -m(t_f) \quad \text{subject to} \\ &\delta \mathbf{r} = \mathbf{r}(t_f) - \mathbf{r}_M = 0 \\ &\delta \mathbf{v} = \mathbf{v}(t_f) - \mathbf{v}_M = 0 \\ &\text{with} \quad \mathbf{y} \in D \end{aligned} \tag{25}$$

The problem is scaled dividing radius by 384000 (mean Earth-Moon distance), upper and lower bounds defining the domain *D* are summarized in Tab.3. .The fundamental difference with respect to the previous transfer problem consist of the presence of the constraints. Furthermore the constraints are nonlinear end do not present an explicit analytical formulation. One way to proceed is to weight the constraints in the objective function. In this particular case the weights applied have a physical meaning since the two constraints and the objective function do not have the same importance. In fact the first goal is to reach the target planet, among all solutions satisfying this constraint, only the ones with minimum arrival velocity have to be considered and among them the one with minimum propellant consumption is the desired solution. The objective function is then augmented in the following way:

$$f = (m(t_0) - m(t_f))/1000 + \delta v/2 + \delta r \tag{26}$$

Table3. Upper and Lower bounds defining the domain D for EP Earth-Mars transfer

Value	t_0 (MJD)	P (day)	u_1	u_2
Lower bound	4000	500	-π	-π
Upper bound	7000	800	π	π

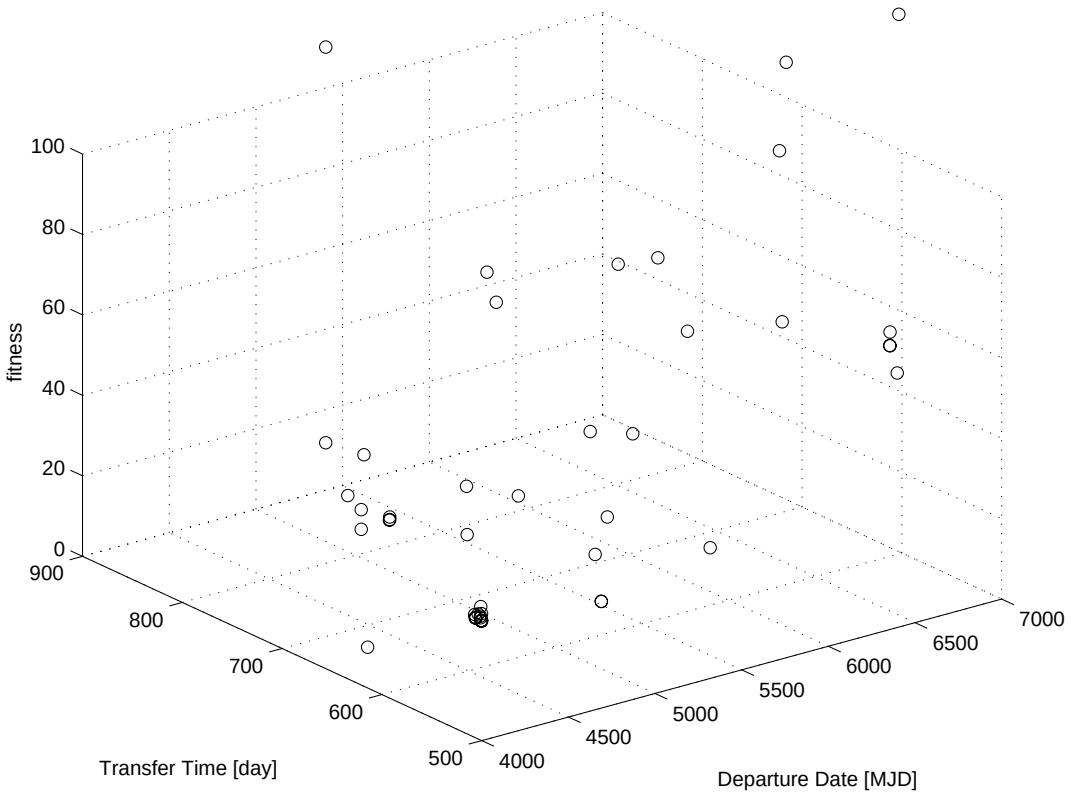

Figure 8 . A group of solutions found by the evolution-branching algorithm

The combined evolution-branching algorithm is then applied to problem and the result is reported in Fig.8 where a group of solutions found have been plotted. In order to check the accuracy of the result all the

solutions have been processed with a gradient method, a comparison can be seen in Tab.4, the best two solutions found have been plotted in Fig. 9. Then the same problem has been solved also using DIRECT and GAOT and the results were reported in Tab. 4 for comparison.

It is remarkable how DIRECT was not able to converge after a considerable number of function evaluations, on the other hand GAOT was able to find a good minimum but just once over 20 runs. In fact the average value for the solutions found by GAOT is comparable with DIRECT.

Table 4. Pure Evolution step with convergence of the best individual

Value	SQP	GAOT	DIRECT	Epic
m_f (kg)	977.1	1089.9	868.68	977.08
δr (km)	6.2e-7	25375	5500800	2181.2
δv (km/s)	1.5997	2.1203	1.2125	1.5987
t_0 (MJD)	4900.7	5673.4	5500	4900.7
u_1	-2.5084-1	-2.7922	1.8617	-2.5085-1
u_2	-2.4875-1	8.796e-1	-2.0944	-2.4879e-1
P (day)	658.14	585.55	727.78	6581.5
Func. Eval.	-	7000[a]	66049	6200

[a]**The number of function evaluations is the mean value per run over 20 runs**

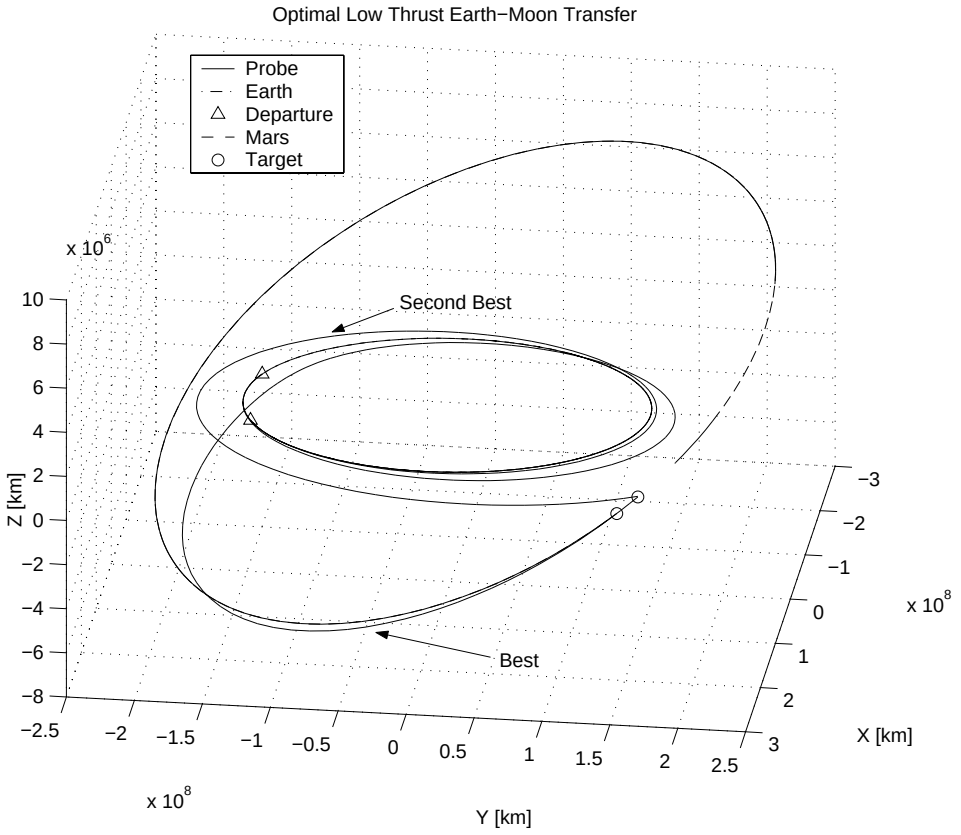

Figure 9. Optimal low-thrust transfer from the Earth to Mars

Multiple swingby Transfer to Pluto

The problem is to find an optimal transfer from the Earth to Pluto passing by a predefined number of intermediate stops (actually swingbys). The propulsion system is chemical, anyway the trajectory, which minimize the overall cost in terms of Δv, is here regarded as optimal for both chemical and electric of propulsion and is used as first guess for a further optimization with a better model of electric propulsion

using the Direct Interplanetary Trajectory Analysis software DITAN [10].

A deep space Δv maneuver has been inserted along the arc connecting two subsequent bodies at an unknown point in time and space, each swingby is modeled collapsing the sphere of influence in one single point linking the transfer arcs before and after the swing by, therefore the following relation must hold:

$$\mathbf{r}_i = \mathbf{r}_o = \mathbf{r}_p \tag{27}$$

where $\mathbf{r}_i$ and $\mathbf{r}_o$ are the incoming and outgoing position vectors respectively and $\mathbf{r}_p$ is the position vector of the planet. Since the swingby is considered un-powered the following relationships between the incoming and the outgoing velocities must hold:

$$\tilde{v}_i = \tilde{v}_o \tag{28}$$

Furthermore, the outgoing relative velocity vector is rotated, due to gravity, by an angle π-2β with respect to the incoming velocity vector and therefore the following relation must hold:

$$\tilde{v}_o^T \tilde{v}_i = -cos(2\beta)\tilde{v}_i^2 \tag{29}$$

where, considering μ the gravity constant of the planet, the complementary angle of rotation of the velocity is defined as:

$$\beta = acos\left(\frac{\mu}{\tilde{v}_i^2 \tilde{r}_p + \mu}\right) \tag{30}$$

All quantities with a tilde are relative to the swing-by planet and $\tilde{r}_p$ is the periapsis radius of the swing-by hyperbola. Constraints given by equation (27) can be explicitly solved while constraints on the velocity require the rotation of the velocity vector $\tilde{\mathbf{v}}_i$ of an angle equal to δ=π-2β in the orbit plane of the hyperbola, which is unknown. Therefore another parameter ω has been introduced, which represents the rotation angle of a plane around the vector $\tilde{\mathbf{v}}_i$

$$\mathbf{n}_\omega = Q(\tilde{\mathbf{v}}_i)\mathbf{n}_i \tag{31}$$

$$\tilde{\mathbf{v}}_o = Q(\mathbf{n}_\omega)\tilde{\mathbf{v}}_i \tag{32}$$

where Q($\tilde{\mathbf{v}}_i$) and Q($\mathbf{n}_\omega$) are the two rotation matrices defined by the quaternions:

$$\mathbf{q}_\omega = \left[\tilde{\mathbf{v}}_i sin\frac{\omega}{2}, \cos\frac{\omega}{2}\right]^T \tag{33}$$

and

$$\mathbf{q}_\delta = \left[\mathbf{n}_\omega sin\frac{\delta}{2}, \cos\frac{\delta}{2}\right]^T \tag{34}$$

respectively and $\mathbf{n}_i$ is the normal to the projection of the incoming vector onto the xy plane (see Fig.12).

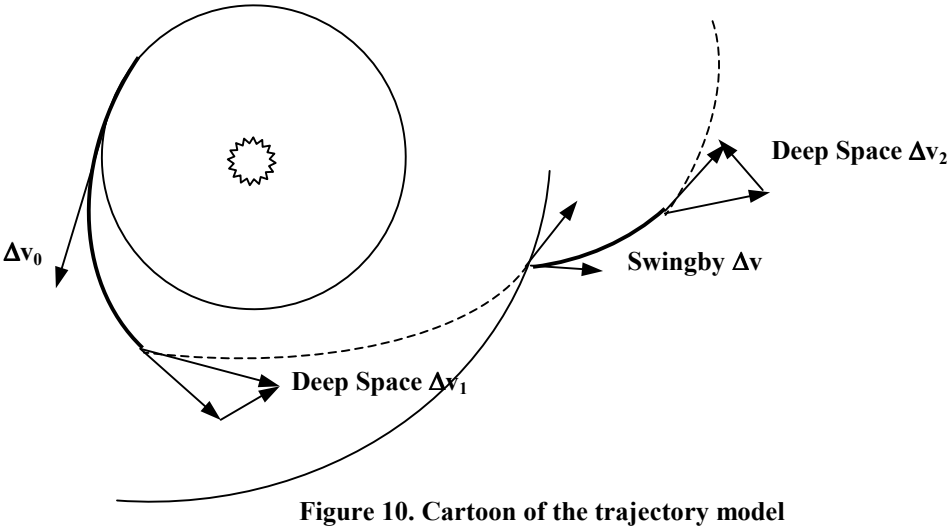

Figure 10. Cartoon of the trajectory model

The outgoing conditions are then propagated for a time t_i up to the deep space maneuver, from that point on a coast arc with period T_i is computed solving a Lambert's problem from the maneuver point to the destination planet (see Fig.10). Therefore, starting from a planet or a generic point in space it is possible to reach a desired point in space passing by a number of swingbys and providing a corresponding number of Δv maneuvers. The problem can then be written in the following form:

$$\begin{aligned} &\min \quad f(\mathbf{y}) = \sum_{i=0}^{N} \Delta v_i \\ &\mathbf{y} \in D \end{aligned} \tag{35}$$

where N is the number of planets after departure and the vector **y** is defined as:

$$\mathbf{y} = [t_0, \Delta v_0, t_1, T_1, \omega_1, \tilde{r}_{p1}, ..., t_i, T_i, \omega_i, \tilde{r}_{pi}, ..., t_N, T_N]^T \tag{36}$$

The problem in this form is amenable to a solution with an algorithm for unconstrained global optimization. First of all the opportunity to use a swingby of the Earth or of another planets of the inner solar system has been investigated. The vector **y** is then extended to include a combination of possible planetary encounters:

$$\mathbf{y} = [p_0, ..., p_i, ..., p_N, t_0, \Delta v_0, t_1, T_1, \omega_1, \tilde{r}_{p1}, ..., t_i, T_i, \omega_i, \tilde{r}_{pi}, ..., t_N, T_N]^T \tag{37}$$

where p_i is the reference number of planet i-th. Now considering a departure from the Earth and two possible encounters before Pluto we take $p_0=3, p_N=9$ (with N=3) ,for the other quantities of vector **y** upper and lower bounds defining domain *D* are summarized in Tab.5.Notice that in this case no intermediate deep space maneuver is inserted for the first transfer and therefore the first Δv is computed just solving a Lambert problem. The swingby altitude h_i is inserted as a ratio between the actual pericenter of the swingby hyperbola $\tilde{r}_{pi}$ and the mean radius of the planet. The final transfer time T_N has an upper limit of 3000 days because a fast transfer to Pluto is required, the launch date has a lower limit 5000MJD because the departure must be in the range [5000,8000]. In this interval there are many different possible launch windows with different characteristics. After 3000 evaluations of the function *f* with the combined evolution-branching algorithms we obtain a number of interesting optima. Three families of solutions found by EPIC are represented in Fig. 11 and the trajectories corresponding to two of the best solutions found are plotted in Fig.13. It is remarkable that most of the optima found have a sequence $\mathbf{p}=[3,3,5,9]^T$ which has been interpreted as if a direct launch to Jupiter was the optimal strategy and no other swingbys were required. Actually an optimal [3,2,5,9] sequence was found with a Δv 219.5m/s higher than the best [3,3,5,9], however the launch window (corresponding to the basin of attraction of the optimum) is apparently smaller and therefore unfavorable. The best solution found for the sequence [3,3,5,9] is reported in table 6 along with the electric propulsion version optimized by DITAN(considering an I_{sp}=6000s and a thrust of 0.2N). For comparison, the best one of a second group of solutions of the family [3,3,5,9] represented in Fig.11, is reported too.

Table 5. Upper and lower bounds defining domain D

Value	Lower Bound	Upper Bound
p_0	3	3
p_1	2	5
p_2	2	9
p_3	9	9
t_0(MJD)	5000	8000
T_1(day)	100	300
ω_1	-π	π
h_1	1	2.5
t_2(MJD)	10	100
T_2(day)	100	600
ω_2	-π	π
h_2	10	100
t_3(MJD)	10	100
T_3(day)	2000	3000

Table6. Earth to Pluto (EJP)Transfer

Value	Best Solution	EP optimized	Best Solution 2	EP optimized
C_3	139.85 km^2/s^2	139.85 km^2/s^2	121.42 km^2/s^2	121.42 km^2/s^2
Launch Date	22/01/2018	22/01/2018	27/12/2016	27/12/2016
Jupiter Encounter	30/03/2019	01/04/2019	11/05/2018	17/06/2018
Swing-by altitude	14.214	12.872	10	10
Δv maneuver	1.37e-3 km/s	-	3.5424 km/s	-
Pluto Arrival	24/09/2027	10/09/2027	15/10/2026	12/12/2026
Mass at Launch	-	2500 kg	-	2500 kg
Mass at Pluto	-	2498.54 kg	-	2362.21 kg

Notice that the propellant spent for the electric propulsion version of the second solution is equivalent (in the simplified hypothesis of the rocket equation) to a Δv of 3.34km/s. Considering that DITAN uses accurate JPL ephemeris while EPIC uses mean elements, first guess solutions computed by EPIC are quite close to the final solution computed by DITAN demonstrating how this approach can be effectively used to provide good initial guesses for a more accurate local search.

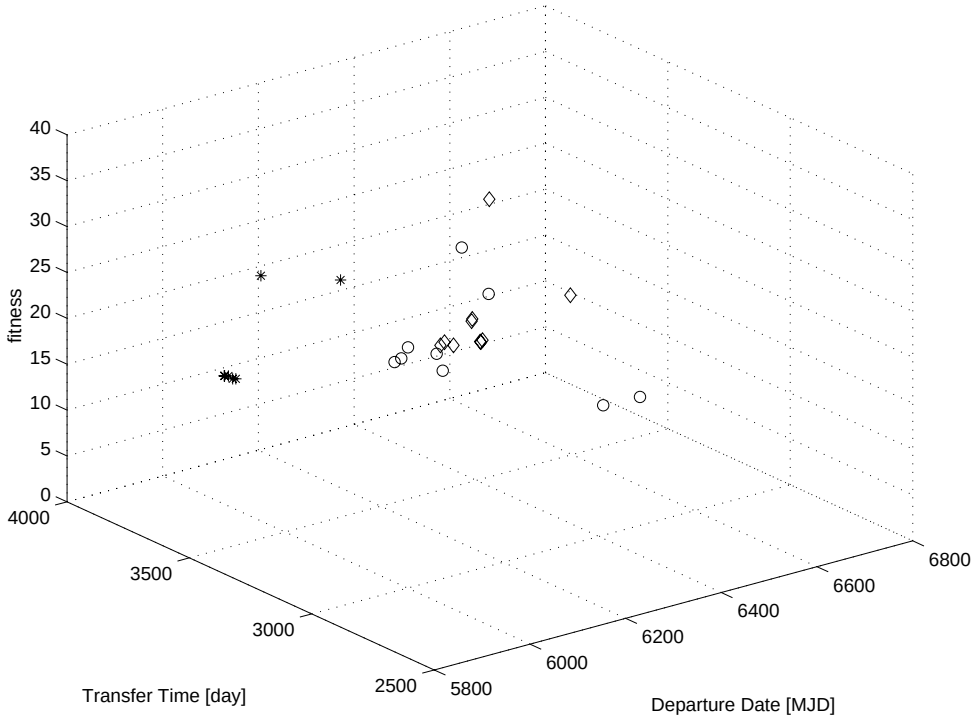

Figure 11. Three families of potential solutions for the Earth-Pluto transfer

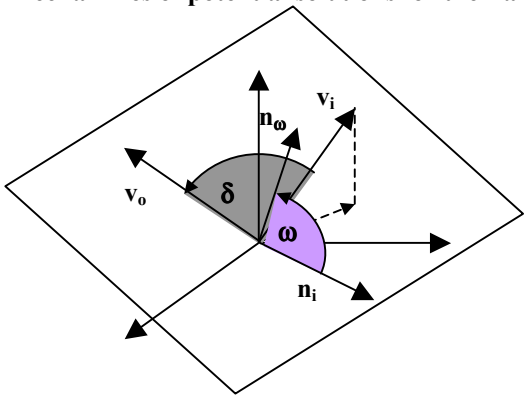

Figure 12. Rotation of the incoming vector to the outgoing vector

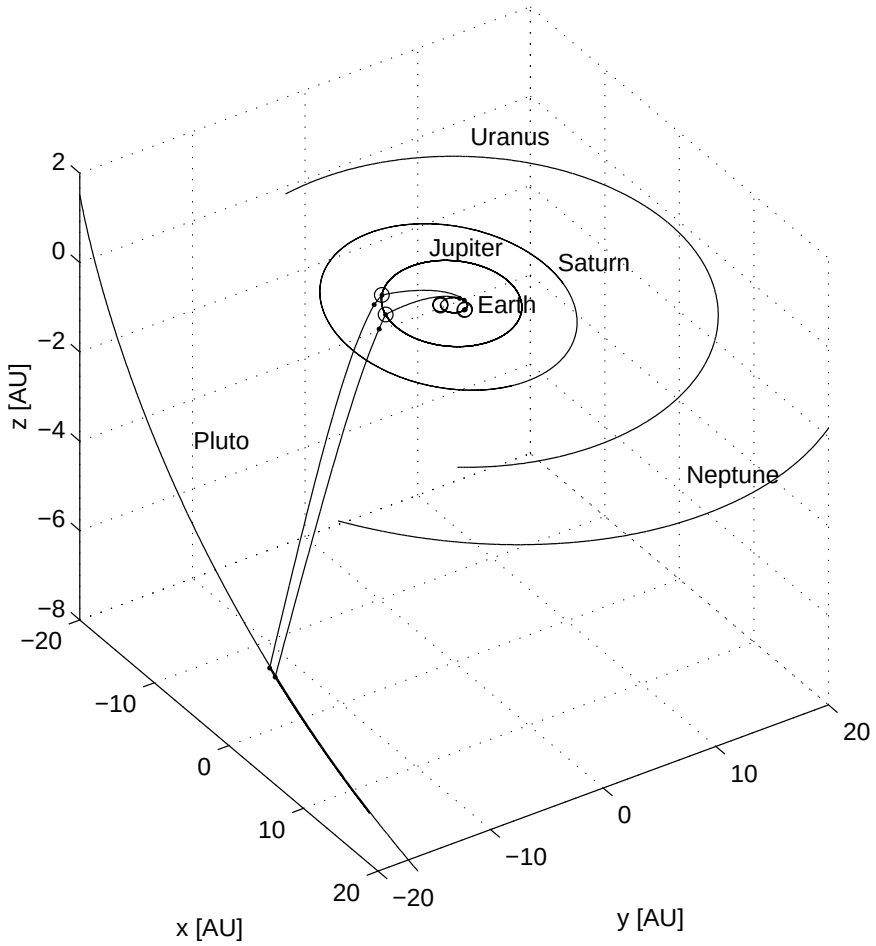

Figure 13. Some EJP optimal first guesses

CONCLUSIONS AND FUTURE WORK

In this paper a combined systematic-heuristic 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 evolution programming and branching is suitable for problems characterized by differentiable and non-differentiable functions combining integer and real variables. The algorithm is based only on local information coming from the evolution of a limited number of individuals in subregions defined by a branching procedure. The outcome of each EP step is used to define new branches and to prune not promising ones. The particular implementation of evolution programming proposed here presents some novel operators like migration and filtering that have given quite good results, compared to Genetic Algorithm or pure branching techniques, on the problems under study providing and independent local convergence toward several local minima. Furthermore the particular mating procedure has demonstrated to be effective to explore widely the solution space avoiding unnecessary clustering of individuals.

Even though the obtained results must be considered preliminary, the proposed algorithm appears to be promising even for more complex space trajectory design problems. The comparison with both pure stochastic and pure systematic approaches demonstrates the effectiveness of the combination of both in particular when the objective function is a black box. In this respect it must be said that an ad hoch

systematic approach specifically dedicated to a certain category of trajectory design problem is expected to be more efficient.

Besides this the algorithm is quite effective as a general tools for problem with bound constraints, at present an extension to treat more complex constraints is under development.

REFERENCES

[1] 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 Optimization Problems: The State of the Art*, June 8,2001.The Coconut Project.

[2]Vasile M., Comoretto G., Finzi A.E. *A Combination of Evolution Programming and SQP for WSB Transfer Optimisation*. AIRO2000, September 18-21 2000 Milano, Italy.

[3] Debban T.J.,McConaghy T.T., Longuski, J.M., *Design and Optimization of Low-Thrust Gravity-Assist Trajetories to Selected Planets*, AIAA 2002-4729, Astrodynamic Spacialist Conference and Exhibit, 5-8 August 2002, Monterey, California USA.

[4] Vasile M. Campagnola S., Bernelli-Zazzera F. *Electric Propulsion Options for a Probe to Europa*. 2nd International Symposium on Low Thrust Trajectories (LOTUS2), Toulouse 18-20 June 2002.

[5] Michaelewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer- Verlag, Telos, 1996, Third Edition.

[6] Barhen J., Protopopescu V., Reister D. *TRSUT: A Deterministic Algorithm for Global Optimization*. Science, 276, 1094-1097 (1997).

[7] Johns D.R., Perttunen C.D. and Stuckman B.E. *Lipschitzian optimization without the lipschitz constant*. J. Optim. Th. Appl., 79:157-181,1993.

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

[9] Vasile M. *A Global Optimization Algorithm for Space Trajectory Design*. ICORD 2002, 27-30 December 2002, Anna University Chennai, Idia

[10] Vasile M., Bernelli-Zazzera F.,Fornasari N., Masarati P. *Design of Interplanetary and Lunar Missions Combining Low Thrust and Gravity Assists*. Final Report of ESA/ESOC Study Contract No. 14126/00/D/CS, ESA/ESOC.