Comput Optim Appl (2011) 48: 635–652
DOI 10.1007/s10589-009-9261-6

A global optimization method for the design of space trajectories

Bernardetta Addis · Andrea Cassioli · Marco Locatelli · Fabio Schoen

Received: 20 November 2008 / Published online: 9 June 2009

Abstract The problem of optimally designing a trajectory for a space mission is considered in this paper. Actual mission design is a complex, multi-disciplinary and multi-objective activity with relevant economic implications. In this paper we will consider some simplified models proposed by the European Space Agency as test problems for global optimization (GTOP database). We show that many trajectory optimization problems can be quite efficiently solved by means of relatively simple global optimization techniques relying on standard methods for local optimization. We show in this paper that our approach has been able to find trajectories which in many cases outperform those already known. We also conjecture that this problem displays a "funnel structure" similar, in some sense, to that of molecular optimization problems.

Keywords Global optimization · Space trajectories · Implicit filter · Black-box · Basin hopping

1 Introduction

A very important problem in mission analysis is the planning of interplanetary trajectories, which consist in launching a spacecraft from a given astronomical body (usually the Earth) along a trajectory which leads to some other astronomical body.

B. Addis
DEI, Politecnico di Milano, Milan, Italy

A. Cassioli · F. Schoen (✉)
DSI, Università degli Studi di Firenze, Firenze, Italy
e-mail: fabio.schoen@unifi.it

M. Locatelli
DI, Università degli Studi di Torino, Torino, Italy

The aim of the mission can be to land on the body or to put the spacecraft into the planet's orbit and the objective of a model is to help trajectory planners in taking the best decision, e.g., on the starting date and other relevant parameters, in order to obtain a "low-cost" mission.

In general, accurate models are computationally very demanding. For this reason simplified models are typically used. Global optimization techniques are usually applied to these simplified models. These might return a single solution (the best detected one) or a set of good solutions. The latter option is usually preferable because it allows for a greater flexibility when the real mission has to be planned.

Once the solution(s) of the simplified models have been obtained, they can be further refined through more accurate and costly models.

In Sect. 2 we will discuss some simplified models, but before doing that we need to give some definitions.

Leg A *leg* is the trajectory followed by the spacecraft between two astronomical bodies (planets or asteroids).

Pericenter radius The *pericenter radius* at an astronomical body is the minimum distance between the trajectory of the spacecraft and the body.

Swing-by A *swing-by* or *gravity assist* maneuver is the result of the gravitational interaction between the spacecraft and the astronomical body: as the spacecraft gets close to the body, such interaction does not change the modulus of the spacecraft velocity but changes its direction. The new direction depends both on the modulus of the velocity and on the pericenter radius.

Deep Space Maneuver A *Deep Space Maneuver* (DSM in what follows) is a change in the spacecraft velocity during a leg (the spacecraft is usually assumed to be able to thrust its engine at most once during each leg).

Lambert's problem and arc Given two points in space and the time of flight between them, the trajectory followed by the spacecraft between the two points can be calculated by solving a *Lambert's problem*, which basically consists in the solution of a second order ordinary differential equation with boundary conditions. The resulting trajectory is called a *Lambert's arc*. Lambert's problems usually have multiple solutions but, exploiting some problem knowledge (and, actually, accepting the risk of excluding good solutions), we can introduce further assumptions which reduce the number of solutions to one (we refer, e.g., to [15] for more details).

After a short survey of existing literature in Sect. 3, details on the GO algorithm we used to tackle the space trajectory planning will be given in Sect. 4. Finally in Sect. 5 we will show and discuss numeric results on a selected test set.

2 Problem definition and analysis

In the problems discussed in this paper we have a sequence of $n + 1$ astronomical bodies $B_0, \ldots, B_n$ (B_0 is usually the Earth). The bodies are not necessarily distinct. We would like to visit the sequence in such a way that the overall mass consumption is minimized. Note that in what follows it is assumed that the sequence is *fixed*, but we may also think of models where the sequence of bodies is part of the decision problem, thus implying the introduction of a discrete choice in the models.

2.1 MGA model

The first model we consider is the Multiple Gravity Assist (MGA in what follows) one. In this model the variables are:

- t_0, the starting date of the mission;
- $T_i,\ i = 1, \ldots, n$, the time of flight along leg i (joining body B_{i-1} with body B_i).

Given the values of these variables, we are able to identify the positions p_{i-1} and p_i respectively of body B_{i-1} at time $t_0 + \sum_{j=1}^{i-1} T_j$, and of body B_i at time $t_0 + \sum_{j=1}^{i} T_j$. Therefore, the solution of the corresponding Lambert's problems allows us to identify the Lambert's arcs (which are conic arcs, either part of an ellipse or of an hyperbola) along all legs. It is then possible to compute the velocities at the end of each leg i and at the beginning of the following one $i + 1, i = 1, \ldots, n - 1$. In order to transfer from one leg to the next one, the spacecraft needs to provide a single impulse, a single change of velocity denoted by Δv_i. In the initial leg the spacecraft will have to provide a single impulse Δv_0 to leave the starting planet's orbit and reach the starting velocity at the initial leg. Similarly, in the final leg the spacecraft will have to provide a single impulse Δv_n to move from the final velocity in the last leg to the velocity of the target astronomical body B_n. Each impulse causes a mass consumption proportional to the modulus of the change of velocity. Therefore, in order to minimize the overall mass consumption, we are led to the following objective function:

$$\sum_{i=0}^{n} \|\Delta v_i\| . \tag{1}$$

Usually, MGA models also include constraints on the pericenter radius r_i at each intermediate body B_i, $i = 1, \ldots, n - 1$, which typically require r_i not to fall below a given threshold $r_i^{\min}$: the spacecraft needs to be far enough from body B_i in order to be able to leave the planet orbit and avoid a "forced" landing, due to the gravitational force. This constraint is also meant to prevent high radiations from the planet.

2.2 MGADSM model

The second model allows for the introduction of DSMs and will be denoted by MGADSM. Such model is more flexible but also more complex. In particular, it requires the introduction of new variables besides those already discussed for the MGA model, in order to take into account the DSMs:

- The modulus and the direction (defined by two angles) of the spacecraft relative velocity at the initial body B_0 (V_∞, u, v).[1]
- The time instant in which each DSM maneuver takes place; these are formulated through the introduction of variables $\eta_i \in [0, 1]$, $i = 1, \ldots, n$, which define at

[1] For the sake of precision, in GTOP database, although related to angles, variables u and v are constrained to belong to the interval [0, 1]: they represent linear transformations of the polar coordinates of the spacecraft relative velocity at the initial body B_0 (see [15] for details).

which portion of leg i the DSM maneuver occurs. More precisely, during leg i a DSM is performed at time

$$t^i_{\mathrm{DSM}} = t_0 + \sum_{j=1}^{i-1} T_j + \eta_i T_i.$$

This way, instead of having a unique Lambert's arc during leg i, we have two of them, the first one joining the position (at time $t_0 + \sum_{j=1}^{i-1} T_j$) of body B_{i-1} with the position of the spacecraft at time t^i_{DSM}, and the second one joining the position of the spacecraft at time t^i_{DSM}, with the position of body B_i at time $t_0 + \sum_{j=1}^{i} T_j$.

- The pericenter radii r_i, $i = 1, \ldots, n-1$, at the intermediate bodies are now independent variables.
- Finally, at each intermediate body B_i, $i = 1, \ldots, n-1$, we need to choose an angle b_i. The spacecraft's incoming velocity and the orbit's eccentricity of the Lambert arc joining the position of the spacecraft at time t^i_{DSM}, with the position of body B_i at time $t_0 + \sum_{j=1}^{i} T_j$, allow to define a cone, along whose surface the spacecraft's outgoing velocity lies: the value of angle b_i uniquely identifies the direction of such velocity along the cone's surface.

Each DSM requires a change of velocity and this implies a mass consumption which has to be included in the objective function.

All these models can be considered as box-constrained black-box ones and, like we did in this paper, no information on the problem, like, e.g., analytical derivatives, can be exploited.

3 Survey of existing literature

In this section we briefly review the existing literature about space trajectory design problems. Different papers (see, e.g., [1, 8, 16, 22]) propose Genetic Algorithms (GA) for this kind of problems. Evolutionary strategies, in particular Differential Evolution (DE), turn out to be a valid alternative (see, e.g., [20]). Hybrid methods have also been proposed. In these methods either the problem structure knowledge (see [15]), or the results of some optimization methods (see [23–26]) are exploited to evaluate portions of the feasible region and, consequently, to discard them or, alternatively, to intensify the search within them. Particularly relevant are recent studies carried on to compare the performance of different GO algorithms on different benchmark problems. The tested algorithms in these studies (see [5, 13, 14, 20, 26–28]) include, besides GA and DE, also Particle Swarm Optimization, Adaptive Simulated Annealing, GLOBAL, COOP, Multilevel Coordinate Search, DIRECT. It is worth to note that, during last years, people at ESA ACT (Advanced Concept Team) carried on a considerable effort to make standard benchmark problems available to the optimization community,both in C/C++ language and in MATLAB (see http://www.esa.int/gsp/ACT/inf/op/globopt.htm). Our hope is that the availability of such common test-bed with the related best known results, will enable to extend the

comparison and encourage people working in optimization to test their own algorithms on these problems. To this aim the annual "Global Trajectory Optimisation Competition," first proposed by the ESA ACT in 2005, could be quite stimulating. This is a one-month contest in which teams are called to find the best solution for a realistic mission proposed by the organizers (the previous edition winners), modeled as a possibly non smooth black-box problem. Information about past editions can be found at http://www.esa.int/gsp/ACT/mad/op/GTOC/index.htm.

The currently available studies show that DE often performs quite well. However, a recent study [27] reveals that a basic version of Monotonic Basin Hopping (MBH, see, e.g., [17, 18]) is able to outperform some other algorithms, including DE, on some benchmark problems. This fact led us to propose a method based on the MBH approach. In Sect. 4 we will discuss the general framework of the proposed method and some possible definitions of its components. In Sect. 5 we will report on the results of extensive numerical tests which have led us to obtain some of the currently best known trajectories for many of the problems proposed by ESA ACT.

4 A new global optimization algorithm for space trajectories planning

Global Optimization (GO) methods can all be viewed as an appropriate mix of *local* and *global* exploration phases. For smooth problems local exploration can be performed by any of the local optimization methods available in the literature (like, e.g., conjugate gradient, Newton, quasi-Newton methods). Global exploration can be performed in a number of ways. In one of the simplest GO algorithms, Multistart, global exploration is simply guaranteed by random sampling over the feasible region, while local exploration is performed through local searches starting at the sampled points. Unfortunately, it is well known that Multistart is deemed to failure when applied to highly multimodal problems. In this paper we consider an approach close to Multistart, but where we introduce a *middle* exploration phase between the local and global one. Such phase can be viewed as a local search over the set of local minimizers. Basically, we first define a neighborhood structure over the set of local minimizers, and then perform a search over the resulting graph (more details can be found in [18]). Following the terminology introduced in [29], local minimizers are organized in *funnels*. A funnel is a subset of local minimizers from each of which we can start a descending sequence over the graph of local minimizers ending at a common local minimizer, called *funnel bottom*. The middle exploration phase takes care of detecting funnel bottoms, whose number is usually much lower than that of the local minimizers. Note that a global minimizer is always a funnel bottom. A Multistart approach where standard local searches are replaced by middle searches returning funnel bottoms, has proved to be very effective for problems like molecular conformation (see [7, 17, 29]) and disk packing (see [2]) ones. In particular, the middle search is often carried on through the Basin Hopping or Monotonic Basin Hopping approaches (see, e.g., [17, 29]). This approach is quite similar to the well known Iterated Local Search method [19] and can even be seen as a special case of VNS [12] in which, however, the neighborhood is not variable. In our case the overall procedure is reported in Algorithm 1.

```
while Multistart stopping criterion is not fulfilled do
    generate a new point X
    while MBH stopping criterion is not fulfilled do
        Z ← Pert(X)
        W ← LS(Z)
        if f(W) < f(X) then
            X ← W
        end
    end
end
```

Algorithm 1: GO algorithm for space trajectories planning

In this algorithm we have that:

- $Pert(X)$ is a perturbation routine generating a point in the neighborhood of the current local minimizer X.
- $f(\cdot)$ is the objective function to be minimized.
- $LS(X)$ is a local search started at point X.

In the above algorithm some components need to be specified. Different choices are possible.

For what concerns the initial points X for the middle exploration phase, they are usually uniformly generated over the feasible region (which guarantees global exploration). However, in some cases initial points can be selected ad-hoc, for instance reading starting points from a repository of known solutions. In the latter case the selection does not aim at favoring a global exploration, but rather at intensifying the search close to the ad-hoc selected point.

Perturbation *Pert*, which allows to explore the neighborhood of the current local minimizer, can be defined in a number of ways. One possibility is to introduce a random perturbation, whose width is at most δ, of all the variables in the current local minimizer. The value of parameter δ allows to control the size of the neighborhood and its choice is quite relevant for the effectiveness of the middle search: if δ is too small, than the middle exploration phase reduces to the local one (we are unable to move from the current local minimizer), while if δ is too large, the middle exploration reduces to the global one. Another possibility is that of randomly perturbing only a (possibly randomly selected) subset of the variables. Finally, when possible, one could even think about some ad-hoc perturbations, exploiting the problem structure.

In MBH the acceptance criterion is a monotonic one (accept W if its function value is smaller than that of X). We mention that also a Metropolis acceptance criterion has been tested in the literature (see, e.g., [29]).

Since MBH moves through local minimizers, an essential component is the local search method *LS*. Here any local search method available in the literature can be employed. However, even after having selected a method, we can further enhance the performance of MBH by employing two-phase local searches. These are local searches where the first phase drives the search towards promising portions of the feasible region (the result may not even be a local minimizer), while the second phase

is a refinement one, actually leading to a local minimizer. In molecular conformation problems the first phase is defined through the addition of geometric penalization terms to the objective function (see, e.g., [7]), but such terms are strictly problem dependent and can not be generalized to other GO problems. Therefore, in this paper we explore a different definition of the first phase based on the concept of Implicit Filter. We describe the details below in the following Sect. 4.1.

4.1 Local optimization inspired by implicit filtering

The idea of two-phase local searches is indeed simple: we perform a first local search which possibly avoids many local optima and quickly leads to a point with low function value; then a second phase refines the result of the first one by performing a more accurate, traditional local optimization.

The idea we exploited for the first-phase local search is the following: even if almost everywhere smooth, space trajectory problems usually display, even for small instances, a function landscape which is characterized by an enormous number of local optima: the objective function looks like a smooth one, perturbed by some sort of noise which generates many local optima. Examples can be seen for example in [20], in which bi-dimensional plots for the objective function in the case of Earth-Jupiter-Saturn, Earth-Mars and other missions, show the presence of many local optima often clustered in a periodic structure.

An interesting technique, called *implicit-filtering algorithm* (IF), has been introduced by Kelley [11, 21] for the case of box-constrained problems for which the objective function can be thought as $f(x) = f_s(x) + \phi(x)$, where $f_s(x)$ is smooth and $\phi(x)$ is not differentiable but such that $f_s(x) \gg \phi(x)$. The basic idea is to substitute the exact gradient with a finite difference estimate and use it inside a standard descent algorithm for smooth optimization. The approximated gradient is computed by forward or centered finite differences, with a step h which, starting from a relatively high value, is gradually decreased during the iterations; this way convergence properties can be derived (see [4]). The kind of problems we planned to tackle seems to fit quite well with the assumptions of Implicit Filtering and resembles for example [3, 4]. However we decided not to actually implement an IF algorithm but to mimic in some sense its behavior. We chose to use the well known SQP code SNOPT [9]. SNOPT is a very effective SQP solver which makes available a finite difference support in which either forward or central differences are used adaptively, applying the following formulae:

4.1.1 Forward scheme

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(\hat{x}) - f(x)}{h(1 + |x_i|)} \quad \text{where } \hat{x}_j = \begin{cases} x_j & j \neq i, \\ x_i + h(1 + |x_i|) & j = i. \end{cases} \tag{2}$$

4.1.2 Central scheme

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(\hat{x}^+) - f(\hat{x}^-)}{h(1 + |x_i|)} \quad \text{where } \hat{x}_j^{\pm} = \begin{cases} x_j & j \neq i, \\ x_i \pm \frac{1}{2}h(1 + |x_i|) & j = i. \end{cases} \tag{3}$$

More precisely, the finite difference approximation scheme used in SNOPT acts as follows [10]: forward differences are used by default, using a step length parameter h which can be chosen by the user; SNOPT switches from forward to central difference when the current point is close to a stationary point. Moreover, the algorithm reduces the step length to ensure feasibility with respect to linear and box constraints. The main advantages of this strategy are the following:

- Forward differences require half function evaluations with respect to centered ones.
- Step length reduction occurs only to ensure feasibility or close to the solution.

SNOPT with finite difference is not an IF-like algorithm, as the strategy used is that of performing very precise derivative estimation, trying to avoid excessive function evaluations. However, as previously suggested, we can separate the local search in two phases: during the first one we carry on an "imprecise" search by choosing a large step h; during the second phase we refine the result by switching to a much smaller h value. Such two-phase local search resembles the behavior of an IF algorithm.

4.2 Other algorithmic issues

We conclude this section by discussing some additional choices which turned out to be quite beneficial both in terms of effectiveness and of robustness.

Variable scaling According to our experience, confirmed also by other works like, e.g., [3], it is important to scale variables in order to reduce numerical problems: each ESA ACT test problem is composed by variables with widely different scales: variables in fact include angles, velocities, departure dates, and so on. In our method we scaled each variable to the interval $[-0.5, 0.5]$.

Periodic variables Many test problems contain variables representing angles constrained to lie in $[0, 2\pi]$. Clearly we can discard such limitation w.l.o.g. and let variables be free, but, to comply with the previous issue, before any local optimization phase variables are scaled back to the original interval. This way we obtain the following advantages:

- The local optimizer can explore the solution space more freely when close to the former boundary.
- During the perturbation step we have not to deal with feasibility of the perturbed variables.

5 Numerical results

We performed a large set of numerical experiments with two objectives in mind: first we wanted to obtain good trajectories, i.e. solutions which were comparable or possibly better than those deposited at the ESA ACT web site. Second, we wanted to check which of many possible variations in our algorithm were the most successful and the most robust ones. By this last term we mean that one of our aims has been that of proposing an algorithm which was capable, in many cases, to produce a set

Table 1 Variables for box constrained ESA MGADSM problems

Name	Meaning	#
t_0	departure time	1
V_∞	dep. vel. modulus	1
u	dep. vel. angle1	1
v	dep. vel. angle2	1
T_i	time of flight	n
η_i	time of DSM i	n
rp_i	pericenter radius at swing-by i	$n-1$
b_i	outc. vel. angle at swing-by i	$n-1$

Table 2 Box constrained ESA MGADSM problems. E: Earth, V: Venus, J: Jupiter, S: Saturn, Me: Mercury, M: Mars, 67P: Comet 67P/Churyumov-Gerasimenko, Pi: generic planet chosen in the set $\{E, V, M, J\}$

Problem name	Variables	n	Planet sequence
Cassini	22	5	E V V E J S
Messenger	18	4	E E V V Me
Rosetta	22	5	E E V E E 67P
Tandem	18	4	E P1 P2 P3 S

of good solutions. This robustness is a particularly required characteristics in this, as well as in many other problems, as aerospace engineers would like to have a bunch of, e.g., good starting date possibilities in order to be able to choose with some degree of flexibility.

Extensive numerical tests have been performed on the GTOP test set freely available from the ESA ACT web site: it is a collection of problems (coded as black–boxes) which have been specifically designed for testing purpose and made available to the scientific community in order to create a shared benchmark test suite for space trajectory problems.

We decided, for what concerns the numerical evidence reported in this paper, to present our results only for MGADSM problems characterized by the presence of box constraints only. We made a few experiments also on constrained problems, but in order to have a more homogeneous set of problems to analyze, we choose not to present results on problems with other kinds of constraints. The meaning of each variable is briefly summarized in Table 1, while problem characteristics (number of variables and sequence of astronomical bodies visited) are listed in Table 2. Note that Tandem is not a single problem but a set of 24 problems, each corresponding to a different sequence of planets from the Earth to Saturn.

Tests have been performed using different machines and compilers, depending on the availability, so it is not easy to give technical details about CPU timing or memory requirements. We summarize our testing environment in the list below:

- Machines used:
  - Intel®Pentium®4 CPU 3.40 GHz, 2048 KB, cache 1 GB ram;
  - AMD Athlon™ XP 2800+, 512 KB cache, 512 MB ram;
  - AMD Athlon™ XP 3000+, 512 KB cache, 512 MB ram.

- Software is mainly coded in C++ (except for the SNOPT library written in Fortran77 and the ESA ACT black-box implemented in C), compiled with GNU `g++`.
- SNOPT version 7.2.

All tests, if not otherwise stated, have been performed with the following stopping criteria:

- Multistart performed $\texttt{NRuns} = 1000$ steps, with uniformly generated starting points.
- At each Multistart step, MBH was executed with a stopping rule calling for stopping as soon as no improvement was observed in the last $\texttt{MaxNoImprove} = 500$ iterations.

We report our results plotting for each algorithm a function $f(t)$ such that for each $t \geq 0$, $f(t)$ represents the percentage of times (out of 1000 independent trials) in which the algorithm obtained a final value which was within $t\%$ of the currently known putative optimum. In other words, if for this problem the global minimum value is estimated to be $f^\star$, and the algorithm produced function values equal to $f_1, f_2, \ldots, f_{\texttt{NRuns}}$, then

$$f(t) = \frac{|\{i \in 1..\texttt{NRuns} : f_i \leq f^\star \times (1 + t/100)\}|}{\texttt{NRuns}}.$$

In practice, at $t = 0$ we can read the percentage of times (if any) the method obtained the currently known global minimum, at $t = 100$ the fraction of runs giving a value which is at most twice the optimum, i.e. no more than 100% worse, and so on. This way it is quite easy to read from the figure which algorithm gave the best approximation to the optimum and which was capable of producing a larger quantity of good results. These graphical representations are closely related to performance profiles (see [6] for details), from which they differ mainly by the fact that here we compare different independent runs of some algorithms on a single problem, while performance profiles are usually employed to show the behavior of single runs of different methods on different test problems.

5.1 Results for the Tandem mission

The huge amount of data generated by our tests is not easy to be summarized, so having observed very similar behavior in all missions we tested, we prefer to present a few pairwise comparisons between different algorithmic options, in order to ease data analysis. As the principal test-bed, we choose the Tandem mission, a box constrained problem with 18 variables. At the ESA ACT web site, 24 instances of box constrained missions to Saturn are proposed, differing on the particular sequence of swing-bys performed. In the following figures we will represent computational profiles in particular for what concerns the mission with highest estimate of the global maximum, i.e. mission 6 (starting with Earth, with 3 swing-bys at Venus, Earth and Earth again). This problem is formulated as a maximization one, as the objective is a function of the final mass of the aircraft.

The first trials we performed were devoted to understand which kind of perturbation was "optimal" during the execution of MBH. In particular, as there seemed

to be some evidence that some variables in the problem like, e.g., starting times and times of flight, were in some sense easier to choose than other ones, or, at least, that, once well chosen, they were quite stable, we tried to check whether perturbations involving only a few variables at a time were more successful than perturbations in which every variable is randomly displaced. We choose the following characteristic parameters for our experiments:

- At each step of MBH, the current solution was perturbed in the following way: for algorithm labeled `MBH1PPertSome` between 1 and 4 coordinates were randomly chosen and uniformly perturbed in an interval of radius equal to 5% of the box containing the variable; for algorithm labeled `MBH1PPertAll` *every* coordinate was uniformly perturbed in an interval whose radius is 5% of the box.
- Numerical derivatives were computed using a parameter

$$h = 10^{-5}$$

  in formulae (2)–(3).
- A single phase of local optimization was performed.

In Fig. 1 we report the results obtained running the two versions of this method, with the graphical representation introduced above.

It is quite evident from Fig. 1 that perturbation of all coordinates is preferable: although both methods find similar global optima, the method based on the perturbation of all variables is significantly more robust (the corresponding curve is significantly "higher" than that of the competing algorithm).

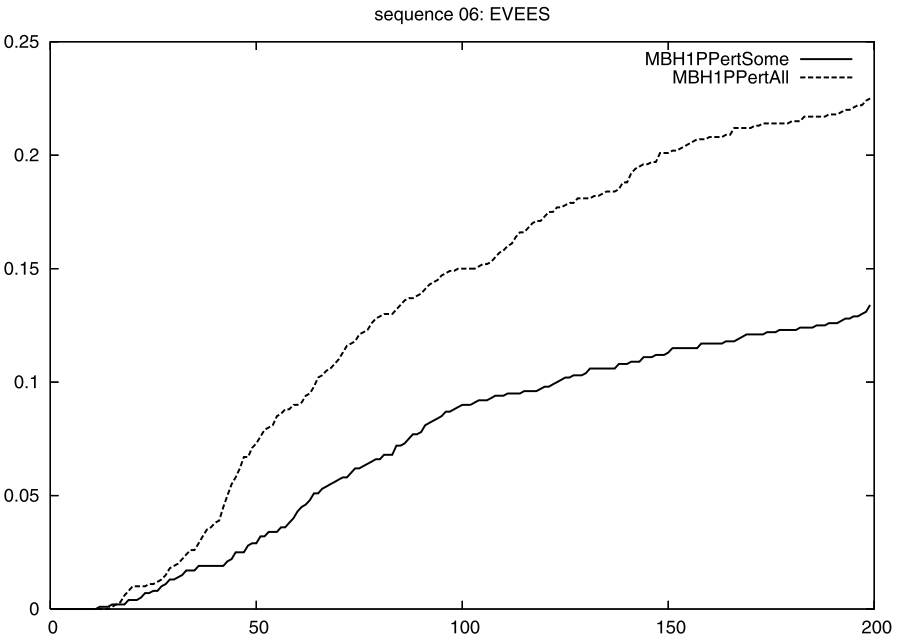

Fig. 1 Comparison of two different perturbations: all variables (*dotted line*) vs. just a few ones (*solid line*)

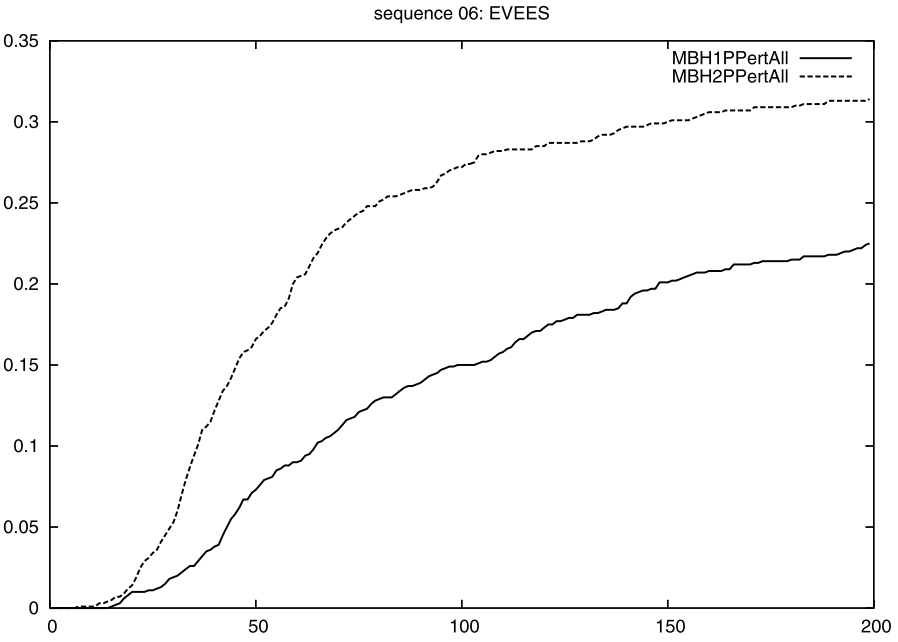

Fig. 2 Comparison between 1-phase (*solid line*) and 2-phase (*dotted line*) algorithms

The second set of experiments was aimed at checking the efficiency of our two-phase approach versus the single phase one. In a two-phase approach first a local search is performed on a "perturbed" problem and then, from the optimum obtained in this way, a "regular" local optimization is started. In the context of the experiments reported in this paper, during the first-phase optimization we simply used a larger step in numerical derivatives. In details, with respect to the scheme outlined above, during the first-phase we let $h = 10^{-2}$ in the formula for numeric derivatives. When the local optimization method, SNOPT, called for stopping, we started a second optimization with the usual parameter $h = 10^{-5}$. Apart from this, we maintained the other parameters unchanged. In Fig. 2 we report the comparison between one- and two-phase optimization.

Again it is evident that using two phases is extremely beneficial both in terms of precision and of robustness. In Fig. 3 we report the comparison between two 2-phase experiments in one of which we changed the maximum perturbation from $\delta = 5\%$ to $\delta = 4\%$. Here the situation is less clear: the two performance profiles intersect and there is no clear winner, although perturbation 5% seems to be slightly preferable. Note that, although not explored here, an adaptive update of the perturbation radius might enhance the performance of the algorithm.

A final experiment was carried on in order to check whether MBH was indeed useful. In order to check whether this was the case, we counted, for the 1000 experiments made, the total number of two–phase local searches performed, which resulted to be 950 046. We then ran the same number of (two-phase) Pure Multistart iterations and checked the obtained result. By the term "Pure Multistart" we mean the

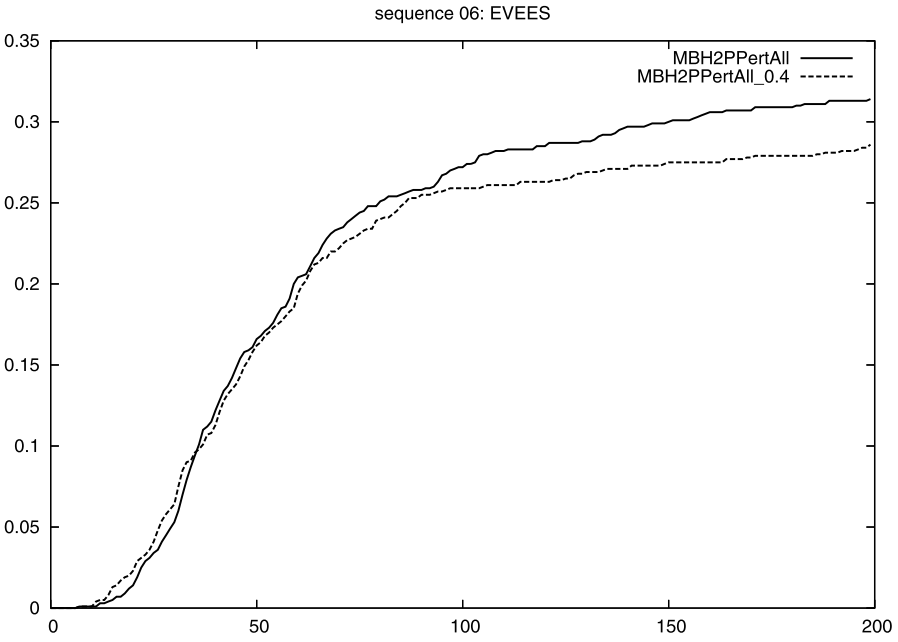

Fig. 3 Comparison between standard (5%) perturbation (*solid line*) and 4% perturbation (*dotted line*)

classical Multistart method in which points are uniformly generated in the feasible set and a local search is started from each of them; the unique variation here is the fact that local searches are indeed composed of two phases. In Fig. 4 we report the comparison between MBH and Multistart; in the figure, for what concerns Multistart, we choose the 1000 best results and compared them with MBH—it is clear that this way the behavior of Multistart is artificially much improved; nonetheless the superiority of MBH is striking. This fact might lead us to conjecture that, similarly to problems in molecular conformation, also space trajectory optimization possesses a "funnel" structure, in which minima are clustered together. The fact that trajectory planning problems might possess a funnel structure apparently was never noticed or conjectured in the literature.

It is worth mentioning that repeating the same kind of analysis on all the 24 tests for the Tandem mission gave roughly the same behavior, so that we can safely conclude that a MBH approach with two-phase local searches seems to be particularly well-suited to solve these problems.

5.2 Results for other box-constrained missions

We also ran experiments on the other box-constrained tests proposed by ESA ACT, namely those related to Rosetta, Messenger and Cassini missions. Again, the same kind of algorithm was particularly effective in all of those tests.

In particular, we consider particularly interesting the fact of having been able, for what concerns the Messenger mission, to discover a competitive solution which

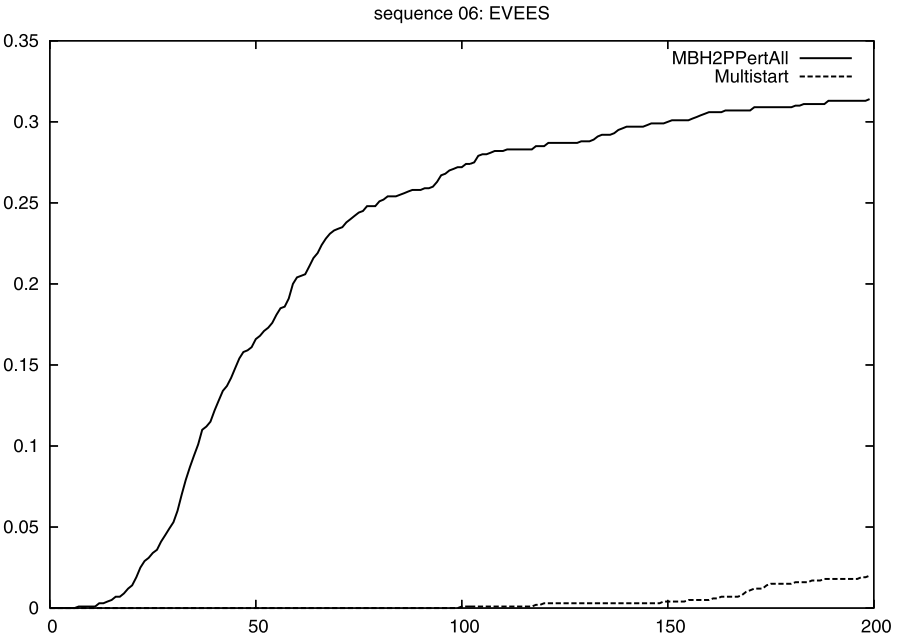

Fig. 4 Comparison between Multistart and MBH

corresponds to starting the mission years before the starting date of previously known solutions (as well as of the real space mission).

We would like to remark that most of the novel putative global optima we found are truly new solutions, i.e. they cannot be considered as refinement of previously known ones. As an example, we plot in the following figures a trajectory assumed to be optimal for Rosetta mission on April 2008, with objective function (representing total mission variation in velocity) equal to 1.417 km/s, and the one we found (and later improved) in May 2008, with objective 1.3678. Although the variation in the objective is not particularly impressive, the trajectories widely differ.

For what concerns other experiments, it is difficult to summarize all of them in this paper. We can trace some of the most relevant improvements. For the Cassini mission, the ESA ACT web site reports solutions ranging from a value of 8.9240 km/sec (starting on October 1997) to our record with value 8.4057 km/sec (starting on November 1997); the real Cassini-Huygens mission started on October 15th, 1997. For the Messenger mission (actually launched in August 2004), the first solution reported by ESA has a starting date at June 2006, with objective function value 8.70257 km/s, while the solution we found starts at March 2003, with an objective value 8.630832 km/s. For the Rosetta mission, started in March 2004, all of the results at ESA ACT site correctly predict launch in the same month. The first solution by ESA was at 1.4491 km/s, while our current record is 1.3437 km/s. An interesting fact is that even in their relative simplicity, the models used in these test problems generate quite reasonable trajectories. In our first attempts to solve a constrained version of the Tandem mission (which is still to be launched), according to ESA ACT we could find

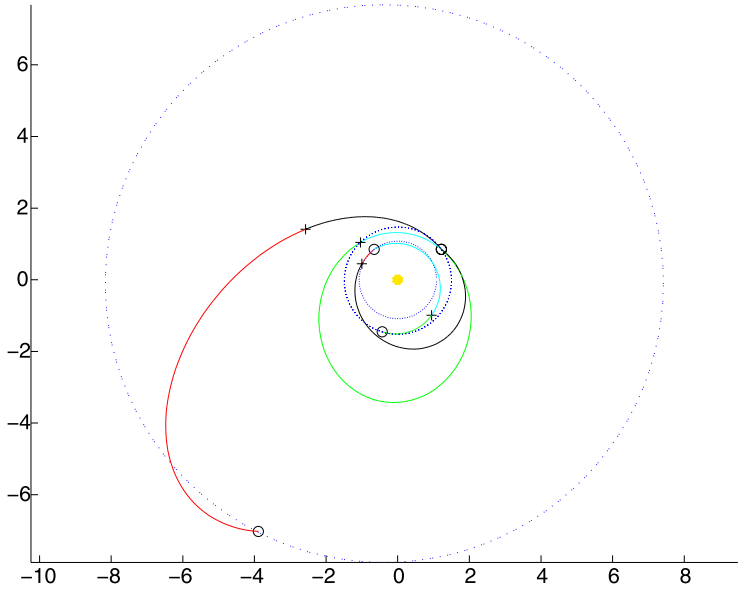

Fig. 5 Rosetta mission, $\Delta_V = 1.417$

both the swing-by sequence and the mission parameters which are very close to those of the planned mission.

As a final remark, we would like to add that, given the apparently enormous number of local optima of these problems, it proved to be particularly useful, after the termination of the runs, to launch short instances of MBH just to refine the best solutions we found. In particular, after each set of runs, we took the 10 best trajectories and, using them as starting points, we performed single-phase MBH runs with higher precision in the estimation of numerical derivatives (we used $h = 10^{-6}$); in these runs we performed a total of 100 Multistart iterations (10 for each starting trajectory), with `MaxNoImprove` $= 300$ and perturbation radius 3%, applied to a randomly chosen subset of 1 to 4 variables. Invariably, these short runs produced significantly improved trajectories, which are now deposited at ESA ACT website (aside we remark that this final refinement is also the reason why in all the experiments represented in Figs. 1–4 the best solution we found is never reached).

6 Conclusions

Trajectory planning is a hard global optimization problem, which is made even more difficult as the analytical expressions of the objective function or the constraints usually are not available. Moreover, even simple bi-dimensional cases display an enormous number of local optima. In this paper we attacked this problem by means of quite standard tools, like a robust local optimizer, an implicit-filter-like scheme for avoiding getting trapped in high level minima, perturbation tools which take into account the periodicity of some variables. These tools are used within a Basin Hopping

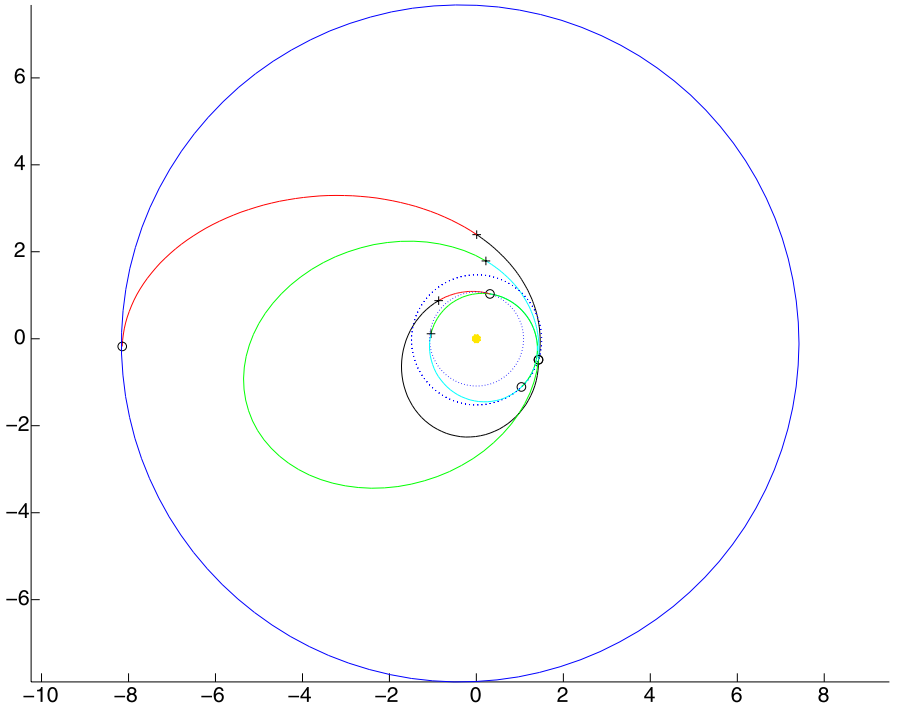

Fig. 6 Rosetta mission, $\Delta_V = 1.3678$

scheme, i.e. within a method which is known to be particularly effective in problems, like those usually found in molecular conformation studies, in which local minima are arranged in clusters known as "funnels." Using this funnel-descent strategy we could significantly improve most of the previously known optima and propose new ones for all of the box-constrained missions available at the ESA ACT web site. We are quite confident that the approach used in this paper can be quite easily extended to more difficult problems with explicit and/or black-box constraints: some preliminary tests have already been performed and the results we obtained are quite encouraging, as it can be checked on the ESA ACT web site. Besides the capability of discovering very good optima for space trajectory planning, our method displayed an important tendency towards generating a whole bunch of alternative, very good, solutions: this fact can be used in order to assess some kind of robustness measure for the proposed trajectory (by checking whether there are many neighborhood good solutions).

Future research will be devoted to extending the range of applicability of Basin Hopping to space trajectory problems. As an example, in [27] an algorithm based on mixing Differential Evolution with a Basin Hopping scheme is proposed.

Acknowledgements This research has been partially supported by Progetto PRIN "Nonlinear Optimization, Variational Inequalities and Equilibrium Problems." The authors would like to express many thanks to ESA ACT group for their commitment to improve global optimization, thanks to their efforts in making test problems available to the scientific community. Among them, our deep and sincere thanks go in particular to Dario Izzo, who contaminated us with his enthusiasm and who has always been available and

willing to give us many explanations of concepts in mission analysis. We would like also to thank, for similar reasons, Massimiliano (Max) Vasile, University of Glasgow, for many discussions, in particular during GTOC3 competition.

References

1. Abdelkhalik, O., Mortari, D.: N-impulse orbit transfer using genetic algorithms. J. Spacecr. Rockets **44**, 456–459 (2007)
2. Addis, B., Locatelli, M., Schoen, F.: Disk packing in a square: a new global optimization approach. INFORMS J. Comput. **20**, 516–524 (2008)
3. Batterman, A., Gablonsky, J.M., Patrick, A., Kelley, C.T., Kavanagh, K.R., Coffey, T., Miller, C.T.: Solution of a groundwater control problem with implicit filtering. Optim. Eng. **3**(2), 189–199 (2002)
4. Choi, T.D., Kelley, C.T.: Superlinear convergence and implicit filtering. SIAM J. Optim. **10**(4), 1149–1162 (2000)
5. Di Lizia, P., Radice, G.: Advanced global optimization tools for mission analysis and design. Technical report 03-4101b, ESA (2004)
6. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
7. Doye, J.P.K., Leary, R.H., Locatelli, M., Schoen, F.: Global optimization of Morse clusters by potential energy transformations. INFORMS J. Comput. **16**, 371–379 (2004)
8. Gage, P., Braun, R., Kroo, I.: Interplanetary trajectory optimization using a genetic algorithm. J. Astronaut. Sci. **43**, 59–75 (1995)
9. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP algorithm for large-scale constrained optimization. SIAM J. Optim. **12**, 979–1006 (2002)
10. Gill, P.E., Murray, W., Saunders, M.A.: User's guide for SNOPT version 7: Software for large-scale nonlinear programming. Systems Optimization Laboratory, University of Stanford (2007)
11. Gilmore, P.C., Kelley, C.T.: An implicit filtering algorithm for optimization of functions with many local minima. SIAM J. Optim. **5**(2), 269–285 (1995)
12. Hansen, P., Mladenović, N., Moreno Pérez, J.: Variable neighbourhood search: methods and applications. Q. J. Oper. Res. **6**(4), 319–360 (2008)
13. Izzo, D.: Advances in global optimisation for space trajectory design. In: 25th International Symposium on Space Technology and Science. Japan Society for Aeronautical and Space and ISTS (2006)
14. Izzo, D.: 1st ACT global trajectory optimisation competition: Problem description and summary of the results. Acta Astronaut. **61**(9), 731–734 (2007)
15. Izzo, D., Becerra, V.M., Myatt, D.R., Nasuto, S.J., Bishop, J.M.: Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories. J. Glob. Optim. **38**, 283–296 (2007)
16. Kim, Y., Spencer, D.: Optimal spacecraft rendezvous using genetic algorithms. J. Spacecr. Rockets **39**, 859–865 (2002)
17. Leary, R.H.: Global optimization on funneling landscapes. J. Glob. Optim. **18**, 367–383 (2000)
18. Locatelli, M.: On the multilevel structure of global optimization problems. Comput. Optim. Appl. **30**(1), 5–22 (2005)
19. Lourenc, H., Martin, O., Stützle, T.: Iterated local search. In: Glover, F., Kochenberger, G. (eds.) Handbook of Metaheuristics. Kluwer Academic, Dordrecht (2002)
20. Myatt, D.R., Becerra, V.M., Nasuto, S.J., Bishop, J.M.: Advanced global optimisation for mission analysis and design. Technical report 18138/04/NL/MV, ESA (2004)
21. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, New York (2006)
22. Rauwolf, G., Coverstone-Carroll, V.: Near-optimal low-thrust orbit transfers generated by a genetic algorithm. J. Spacecr. Rockets **33**, 859–862 (1996)
23. Vasile, M.: A global approach to optimal space trajectory design. Adv. Astron. Sci. **114**, 629–647 (2003)
24. Vasile, M.: Design of Earth-Mars transfer trajectories using evolutionary-branching technique. Acta Astronautica **56**, 705–720 (2005)
25. Vasile, M., De Pascale, P.: Design of Earth-Mars transfer trajectories using evolution-branching technique. In: 54th International Astronautical Congress (2003)
26. Vasile, M., Locatelli, M.: A hybrid multiagent approach for global trajectory optimization. J. Glob. Optim. (2008, to appear)

27. Vasile, M., Minisci, E., Locatelli, M.: On testing global optimization algorithms for space trajectory design. In: Proceedings of AIAA 2008 (2008)
28. Vinkó, T., Izzo, D., Bombardelli, C.: Benchmarking different global optimisation techniques for preliminary space trajectory design. In: 58th International Astronautical Congress. International Astronautical Federation (IAF) (2007)
29. Wales, D.J., Doye, J.P.K.: Global optimization by Basin-Hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. J. Phys. Chem. A **101**(28), 5111–5116 (1997)