Acta Futura 8 (2014) 45-55
DOI: 10.2420/AF08.2014.45

Acta Futura

GTOC5: Results from the European Space Agency and University of Florence

Dario Izzo [*], Luís F. Simões, Chit Hong Yam, Francesco Biscani
European Space Agency, Advanced Concepts Team

David Di Lorenzo, Bernardetta Addis, Andrea Cassioli
University of Florence, Global Optimization Laboratory

Abstract. This paper describes the methodologies used to tackle the problem of the 5th Global Trajectory Optimization Competition within the team composed by the Advanced Concepts Team of the European Space Agency and the Global Optimization Laboratory of the University of Florence. The method pursued is powered by two innovative approaches: a linearized model of the 'self fly-by' aiding a first broad tree search of chemical propulsion options and the use of global optimization techniques (Monotonic Basin Hopping, in this case) applied directly to the low-thrust trajectory model.

1 Preliminaries

In the context of the Global Trajectory Optimization Competitions we consider the problem of the fifth edition (GTOC5), released by Moscow State University scientists, winners of the 4th GTOC edition. The exact mathematical formulation of the problem is described in [1] and we will assume the reader is familiar with it. In short, starting from the Earth, a spacecraft equipped with low-thrust propulsion capable of delivering a constant thrust of 0.5 N, needs to visit as many asteroids as possible from a given list of $N = 7075$ possible targets. Each visit is a rendezvous. After an asteroid has been visited, an optional fly-by can be attempted (at any time) to deliver a penetrator to the asteroid surface, and thus increase the mission value. The primary mission objective is thus to visit (and revisit) as many asteroids as possible within the mission constraints. A simple calculation shows the complexity of the problem. The possible asteroid sequences (considering only subsequent rendezvous) is $\frac{N!}{(N-n)!}$, where n is the sequence length. As soon as n grows larger, the problem complexity grows and soon a simple grid search becomes unsuitable. The situation is actually much worse as asteroid re-visits are also possible. As the problem complexity is overwhelming we start by introducing one simplifying assumption in the attempt to map the problem into an easier case. We assume that the spacecraft performs a fly-by delivering the penetrator immediately after each asteroid rendezvous. Under this assumption

[*]Corresponding author. E-mail: dario.izzo@esa.int

we may state the GTOC5 problem as

$$
\begin{array}{rll}
\text{find:} & \mathbf{seq} \subseteq \mathcal{A}^n, \mathbf{mjd} \in \mathbb{R}^{n+1} & \\
\text{to maximize:} & n = \text{cardinality of } \mathbf{seq} & \\
\text{subject to:} & m_f(\mathbf{seq}, \mathbf{mjd}) \geqslant 500 \text{ kg} & \\
& mjd_{n+1} - mjd_1 \leqslant 15 \text{ years} & \\
& mjd_{i+1} > mjd_i \quad i \in \{1, \dots, n\} & \\
& 57023 < mjd_1 < 61041 & \\
& A_i \neq A_j \qquad i \neq j \quad i \in \{1, \dots, n\} &
\end{array} \tag{1}
$$

where $\mathcal{A}$ is the set of all 7075 asteroids, **mjd** is a vector containing the launch epoch and all the rendezvous epochs (dimension $n+1$) and $\mathbf{seq} = [A_1, A_2, A_3, .., A_n]$ is a vector containing the asteroid encounter sequence. The function m_f is the result of a global low-thrust optimization where the spacecraft dynamics is accounted for. We assume here that infeasible choices of (**mjd**,**seq**) (from the dynamical point of view, e.g. trajectories that cannot perform the requested rendezvous/fly-bys in the selected epochs) result in $m_f < 500$ kg. In order to solve the problem stated in (1) we follow a two step approach which is common in trajectory design and that proved quite valuable to tackle problems in previous editions of the GTOC (see for example [2, 3]). We first compute $m_f(\mathbf{seq}, \mathbf{mjd})$ assuming the spacecraft equipped with a chemical engine capable of delivering instantaneous ΔV changes. This allows to compute m_f with a greatly reduced computational effort, and thus to explore the search space exhaustively. In a second step, we map the solutions found in this "chemical version" of the problem back to the original one. The rest of the paper is organized as follows: in Section 2 we study the first trajectory leg (i.e. Earth-Asteroid transfer) for all possible asteroids, in Section 3 we explore the structure of low-thrust self-fly-by trajectories and we propose a simple chemical approximation for such legs. Exploiting these results we are able to define, in Section 4, the chemical version of our search problem and to propose, in Section 5 a few techniques to explore the vast solution space. In Section 6 we map the solutions obtained back to the full low-thrust problem. Finally, in Section 7 we discuss the experience we gained from the application of the proposed methods.

2 The first low-thrust leg

Any trajectory that is a solution to the problem stated in (1) needs to start with an Earth-Asteroid leg. Thus, we study, separately, the optimal control problem for the first leg with the intention to further simplify the overall problem. In particular, we study the low-thrust transfer from the Earth to a generic asteroid $A_1 \in \mathcal{A}$ allowing (from the GTOC5 problem description) as a starting range $57023 < mjd_E < 61041$ and constraining the starting velocity gain delivered by the launcher to $\Delta V \leqslant 5$. Because of the launch window width, more than time one launch possibility for any considered asteroid is to be expected. In order to locate them efficiently, we use, directly on the low-thrust model, the global optimization approach described in Yam et al. [4]. In short, we transcribe the Optimal Control Problem (OCP) into a Non Linear Programming Problem (NLP) using a method similar to the Sims-Flanagan transcription [5] and we then consider the resulting NLP as a global non-linearly constrained optimization problem which we solve using the Monotonic Basin Hopping (MBH) technique [6, 7]. This procedure allows us to find good launch possibilities in a reasonably short time, since we are able to compute good launch possibilities to all asteroids, first with time as objective function and then with mass. Overall we are able to solve, for two different objectives, a total of 70,750 non linearly constrained global optimization problems (ten per asteroid) determining good launch opportunities to reach the asteroids in the list. We denote the best launch window for each asteroid A_1 in the sequence with $mjd_E(A_1)$. It is to be noted that this first optimization was run in parallel over 16 CPUs and completed overnight (the open-source PaGMO optimization library was used for this purpose [8]). As the amount of results from this optimization effort is huge, in Table 1 we report, as an example, only some of the best launch possibilities found for both minimal time and minimal mass ranked with respect to the objective function. Launch occasions used in later optimization runs (of the full trajectory) are also included. Note that out of the 7075 asteroids, roughly 3000 result into unfeasible trajectories (for the considered constraints), and thus are not selected in the following searches as starting asteroids. To allow for greater speed, a rather coarse mesh grid was used as a precise computation of each optimal trajectory was, at this point of our solution strategy, not a priority.

3 The self-fly-by

To formalize the GTOC5 problem in the form given by (1), we assumed that the spacecraft always revisits an asteroid immediately after it has performed the rendezvous with. This creates a trajectory leg (or phase)

 DOI: 10.2420/AF08.2014.45

Time optimal				Mass optimal			
Rank	Asteroid	m [kg]	t [days]	Rank	Asteroid	m [kg]	t [days]
1.	2001 GP2	87.47	53.88	1.	2009 QR	41.56	700.95
2.	2007 XB23	98.54	66.44	2.	2006 QV89	47.15	907.44
3.	2009 QR	99.24	67.24	3.	2006 XY	48.76	711.62
...	...	...	....	...	...	...	...
8.	2006 XY	127.14	98.91	6.	2007 UN12	51.13	942.19
15.	2006 QV89	138.39	111.67	50.	2007 XB23	77.18	771.57]
20.	2007 UN12	146.89	121.32	103.	2008 JL24	93.08	1009.81]
43.	1991 VG	175.33	153.60	105.	2001 GP2	94.11	918.56]
50.	2008 ST	179.92	158.81	110.	1991 VG	95.61	1079.69
113.	2008 JL24	216.41	200.22	129.	2008 ST	101.32	827.77

Table 1. *Some opportunities found with the global optimization of the first trajectory leg*

having some unusual boundary conditions, namely a zero departure relative velocity $\Delta V_{dep} = 0$ and an arrival relative velocity $\Delta V_{arr} \geqslant \overline{V} = 0.4$ km/s, with the arrival and departure object being the same (i.e. the asteroid). We call this leg a self-fly-by leg. Representing such a leg as a ballistic arc, for each given transfer time, there exists a Lambert solution (the one that matches exactly the asteroid orbit). Following this representation we could estimate the total ΔV as the velocity mismatch at arrival between the Lambert solution (i.e. the asteroid orbit) and the next phase (i.e. $\Delta V = \overline{V} = 0.4$ km/s) but we would have no indication on the necessary transfer time as such a ΔV does not depend on it. We thus develop an alternative simplified model for this peculiar leg by considering, in one dimension, the motion of a point mass subject to an external acceleration. The point mass position will be indicated by s, its velocity by v and its acceleration by a. We assume that the point mass (our spacecraft) will accelerate at its maximum capability $\overline{a} = T_{max}/m_0$ away from the origin (asteroid) for a time t_1 and then it will accelerate back towards the origin for a time t_2. Starting from $s = 0$ and $v = 0$ we get:

$$\begin{cases} s(t_1 + t_2) = \frac{1}{2}\overline{a}t_1^2 + \overline{a}t_1t_2 - \frac{1}{2}\overline{a}t_2^2 \\ v(t_1 + t_2) = \overline{a}(t_1 - t_2) \end{cases}$$

if we also require that $s(t_1+t_2) = 0$ and $v(t_1+t_2) = \overline{V}$ (which accounts for the terminal boundary conditions at the fly-by) we conclude:

$$\begin{cases} t_1 = \frac{2}{\sqrt{2}}\frac{\overline{V}}{\overline{a}} \\ t_2 = \frac{\overline{V}}{\overline{a}}\left(\frac{\sqrt{2}}{2} + 1\right) \end{cases}$$

hence,

$$\begin{cases} \Delta T_{sfb} = t_1 + t_2 = m_0\frac{\overline{V}}{T_{max}}\left(1+\sqrt{2}\right) \\ \Delta V_{sfb} = \overline{a}\Delta T = \overline{V}\left(1+\sqrt{2}\right) \end{cases} \quad (2)$$

This crude model of a low-thrust self-fly-by leg is valid under the assumption of an infinite specific impulse (this hypothesis can be removed by propagating also the mass equation, leading to more complex algebraic formulas) and can be considered as a zero-th order approximation of the spacecraft motion relative to the asteroid. The advantage is that, by using (2), we can put a quick estimate on the propellant consumption and on the time of flight for such a leg. In particular, for the specific mission data, the ΔV_{sfb} is estimated to be equal to $\Delta V_{sfb} = 960$ m/s for all self fly-by legs, while the time of flight varies from $\Delta T_{sfb} \approx 150$ days, at the beginning when the spacecraft weights 4000 kg, to $\Delta T_{sfb} \approx 18$ days at the end when the spacecraft weights only 500 kg. In order to test the validity of such a model in the context of the GTOC5 problem, we performed the global low-thrust optimization of self fly-by legs selecting at random 10% of the asteroid population and letting the starting epoch free in the initial launch window. We used an initial mass of $m_0 = 4000$ kg and we let the velocity direction of the spacecraft relative to the asteroid free at the fly-by. By doing so, we let the optimizer select the best starting orbital anomalies to take advantage of orbital mechanics effects in order to shorten the self fly-by duration.
The results are shown in Figure 2. We plotted the ΔT_{sfb} and ΔV_{sfb} resulting from the low-thrust global optimization against semi-major axis and eccentricity of the target asteroid. We observe how the predictions coming from (2) are quite crude, still they represent an improve-

DOI: 10.2420/AF08.2014.45

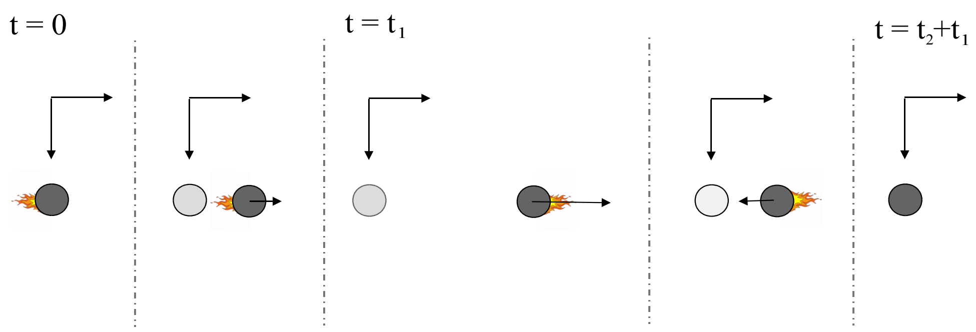

Figure 1. *Visualization of the simple model for the low-thrust, self-fly-by leg*

ment with respect to a simpler purely ballistic approximation, furthermore they systematically overestimate the values resulting from the solution of the time optimal self fly-by problem. We also see that only a weak correlation exists between the asteroid semi-major axis and eccentricity and the accuracy of the prediction.

4 The chemical problem

Using the results derived in Sections 2-3 we transform the problem stated in (1) into a simpler one where the spacecraft trajectory is built by putting together Lambert arcs, self fly-by arcs (described by (2)) and the first leg resulting from the global optimization described in Section 2. We state the following new problem, which we refer to as the chemical version of the full-problem:

$$\begin{array}{rl} \text{find:} & \mathbf{seq} \subseteq \mathcal{A}^n \\ \text{to maximize:} & n = \text{cardinality of } \mathbf{seq} \\ \text{subject to:} & m_f^*(\mathbf{seq}) \geqslant 500 \text{ kg} \\ & T_f^* = mjd_n - mjd_E \leqslant 15 \text{ years} \\ & A_i \neq A_j \end{array} \tag{3}$$

Formally, the difference with respect to (1), is that the continuous part of the decision vector (i.e. the modified Julian dates) has disappeared as it now depends, via the Lambert solution, solely on the asteroids selected. The final mass constraint takes a different form (this reflects the change in the trajectory model) and the starting epoch is not in the decision vector as it depends only on the choice of the first asteroid A_1 (for each asteroid we considered the best launch opportunity found. In later searches we also considered the second and third bests in an attempt to make our search less greedy). Note that a number of different options can actually be considered when computing $m_f^*(\mathbf{seq})$ (e.g., number of revolutions in the Lambert solution, or Lambert arc duration). In these cases, we always perform an inner search to greedily select the best option.

4.1 Computing $m_f^*(\mathbf{seq})$

Consider $\mathbf{seq} = [A_1, A_2, ..., A_n]$. We denote the epoch and the spacecraft mass at the i-th asteroid with respectively t_{A_i} and m_{A_i}. Consider now the transfer $A_i \xrightarrow{fb} A_i \xrightarrow{rndv} A_{i+1}$: an asteroid to asteroid rendezvous with self fly-by. We estimate its costs using the expressions:

$$\begin{array}{l} m_{A_{i+1}} + m_{pld} = \\ = \left[m_{A_i} \exp\left(-\frac{\Delta V_{sfb}}{I_{sp} g_0}\right) - m_p\right] \exp\left(-\frac{\Delta V_{i(i+1)}}{I_{sp} g_0}\right) \\ t_{A_{i+1}} = t_{A_i} + \Delta T_{sfb} + \Delta T_{i(i+1)} \end{array} \tag{4}$$

where $\Delta V_{i(i+1)}$ and $\Delta T_{i(i+1)}$ are the estimates of the $A_i \xrightarrow{rndv} A_{i+1}$ leg, m_p is the penetrator mass left at the fly-by and m_{pld} is the payload mass delivered at each asteroid. Starting from m_{A_1} and t_{A_1} as returned by the global optimization of the first leg, we may use the above equation to sequentially to obtain $m_f^* = m_{A_n}^*$ and $T_f^* = t_{A_n} - t_E$. The estimates for $\Delta V_{i(i+1)}$ and $\Delta T_{i(i+1)}$ are the only data needed. In order to compute them, we consider all the multiple revolution Lambert's problem solutions for the transfer from A_i to A_{i+1} at time $t_{A_i} + \Delta T_{fsb}$ for different values of the transfer times $\Delta T_{i(i+1)}$ (we consider all values in [100,700] days with bins of 10 days) and we select the best one in terms of minimal velocity change (evaluated as the mismatches with the asteroid velocities, after having subtracted $\overline{V}$).

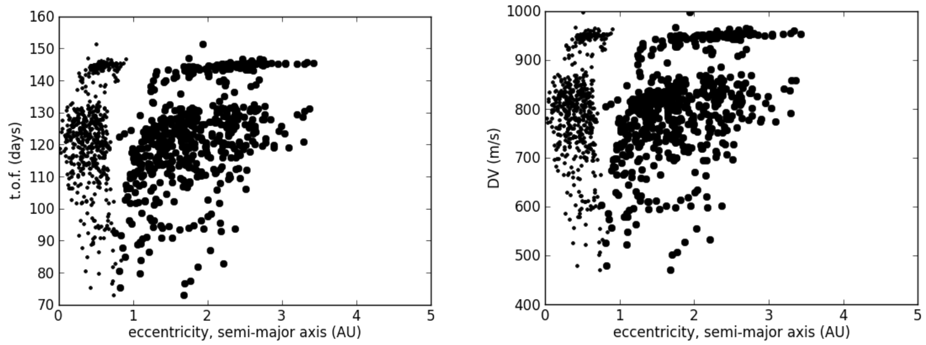

FIGURE 2. ΔT_{sfb} *and* ΔV_{sfb} *distributions for the time optimal self fly-by legs. The zero-th order approximation has* $\Delta T_{sfb} = 150$ *days and* $\Delta V_{sfb} = 960$ *m/s, and can thus be considered an upper bound.*

5 The tree search

Let us now consider problem (3). The non linear constraints m_f^* and T_f^* share a monotonicity property with respect to n. Given an allowed sequence $\mathbf{seq} \subset \mathcal{A}^n$ of cardinality n, and a possible new asteroid, the new allowed sequence $\mathbf{seq}' \subseteq \mathcal{A}^{n+1}$ of cardinality $n+1$, obtained by adding such asteroid, satisfies

$$m_f^*(\mathbf{seq}) \leqslant m_f^*(\mathbf{seq}')$$
$$T_f^*(\mathbf{seq}) \geqslant T_f^*(\mathbf{seq}')$$

This property, specific of the given problem, allows us to explore the search space of feasible sequences incrementally, starting from short length sequences and adding asteroids when possible. We implemented two approaches that exploit such a structure, a branch a prune recursive tree search, and an order based Genetic Algorithm using hidden genes.

5.1 Branch and prune

The branch and prune algorithm we developed is, essentially, a recursive search over the tree of possible chemical representations of the full trajectory. The algorithm is built around the basic recursive function described in Algorithm 1.

The choices of the pruning criteria and the list of asteroid candidates to consider at each step are crucial for a successful application of such an algorithm. A bad selection of these functions can result in pruning good branches. Within the context of the GTOC competition there is no way of knowing how to choose these functions, nor of studying the impact of a certain choice on the resulting trajectories. So expert knowledge is necessary and discriminates success from failure. The list of possible candidates considered as reachable from a given asteroid is formed by considering the first 512 out of the 7075 asteroids ranked using the simple Edelbaum approximation [9]. We list below the pruning criteria we used:

Algorithm 1 Recursive procedure for the Branch and Prune

bp_step($[A_1, A_2, ..., A_n], m_{A_n}, t_{A_n}$)
  $\tilde{A} \leftarrow$ list of possible candidates for the next asteroid
  for A_{n+1} in $\tilde{A}$ **do**
    Compute $m_{A_{n+1}}, t_{A_{n+1}}$ using (4)
    if prune($N, m_{A_{n+1}}, t_{A_{n+1}}$) **then**
      log and continue
    else
      call bp_step($[A_1, ..., A_{n+1}], m_{A_{n+1}}, t_{A_{n+1}}$)
    end if
  end for

1. At each recursive step we define $a_{max} = \eta T_{max}/m_{A_n}$ and we prune if $\Delta V_{n(n+1)}/\Delta T_{n(n+1)} \geqslant a_{max}$. The coefficient η is set to 0.9;
2. We prune if $m_{A_{n+1}} \leqslant 400$ kg;
3. We prune if $T_{A_{n+1}} \geqslant 16$ years;
4. At each recursive step, we define a tolerated variation on the mass as $dm_{tol} = 600 - 200(n+1)/N_t$ kg, where n_t is the maximum length found so far for a feasible sequence plus one. We then prune if $[4000 - 3500(n+1)/n_t] - m_{A_{n+1}} \leqslant dm_{tol}$;

DOI: 10.2420/AF08.2014.45

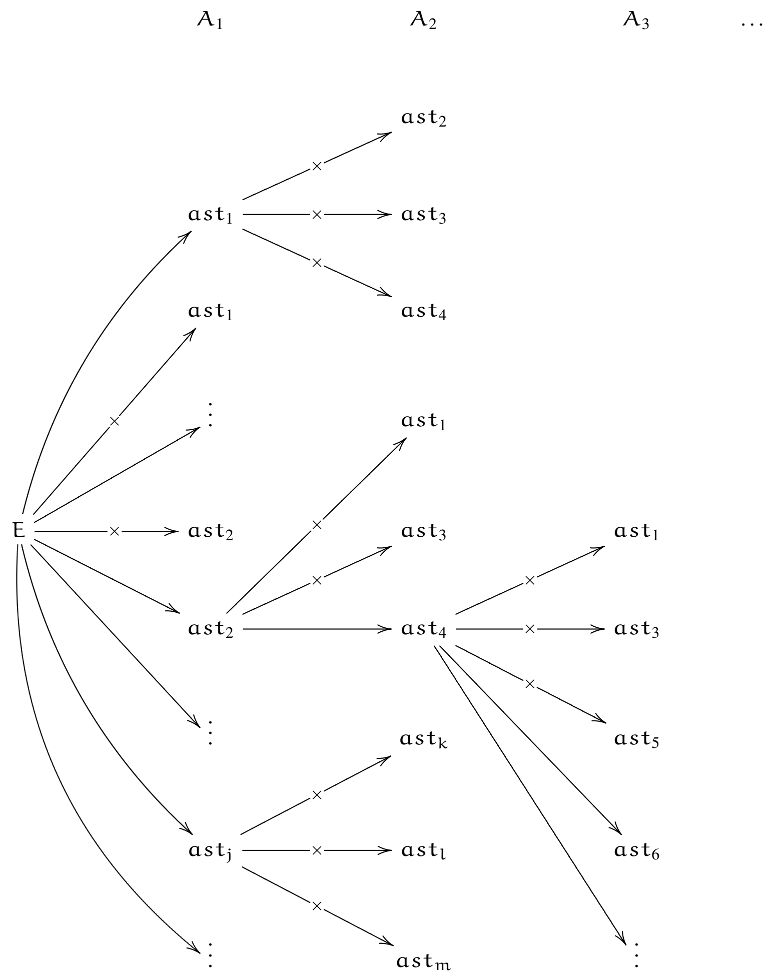

FIGURE 3. *Visualization of the branch and prune process over a small part of the search tree. Starting from the Earth (left) only the first three asteroid selections are shown and only a subsets of the transfer opportunities between asteroids are considered. Most branches are here pruned helping the visualization.*

 DOI: 10.2420/AF08.2014.45

5. At each recursive step, we define a tolerated delay on the time as $dt_{tol} = 1000 - 500(n+1)/N_t$ days, where N_t is the maximum length found so far for a feasible sequence plus one. We then prune if $(t_{A_{n+1}} - t_E) - 15(n+1)/n_t \leqslant dt_{tol}$.

The last two pruning criteria are used to prune trajectories that accumulate too much delay (either in mass or time) with respect to a nominal linear schedule bringing in $n_t + 1$ encounters the mass from 4000 kg to 500 kg and the time from 0 years to 15 years. A visualization of the resulting search process is shown in Figure 3. We run such an algorithm starting from all launch opportunities found performing the global optimization of the first Earth-Asteroid leg (see 2). The algorithm, run in parallel over 8 CPUs, completed the entire tree search in roughly a week, returning millions of asteroid sequences and their encounter dates. Remarkably, in all the computed sequences longer than 16, the first asteroid A_1 was one of 1991 VG, 2007 UN12, 2008 JL24 and 2008 ST. We found several sequences of length $n = 17$ (which we later could not transform into a feasible low-thrust trajectory), but no solution was found with a sequence length $n = 18$.

5.2 Order-based Genetic Algorithm

The main risk with a branch and prune approach lies in the fact that in order to obtain a tractable search space, strict pruning criteria need to be enforced, so as to discard vast swaths of the full search space. Branching and pruning criteria are greedy, in the sense that they demand performance to be above certain thresholds at every step, and their enforcement thus prevents the identification of solutions that, in order to reach excellent performance levels at some point in the future, first need to "pay" high costs.

The second approach followed for the identification of good asteroid sequences used an order based Genetic Algorithm (GA), in which a population of complete candidate solutions (chromosomes) is evolved towards maximization of a given fitness function. Chromosomes are here ordered subsets of the set of asteroids available for the mission, and crossover and mutation operations are defined on them, that always generate valid permutations. The goal with pursuing this second approach was to conduct an exploration on the much larger search space resulting from a relaxation of the pruning criteria used in the branch and prune tree searches. Exhaustive enumeration of the possible solutions was then no longer feasible, but we assumed this approach could succeed in identifying good solutions possibly discarded by the branch and prune's greedier approach.

A population of 5000 chromosomes was used. Each chromosome in the initial population was randomly generated by uniform sampling without replacement of 20 asteroids from the full set of 7075 asteroids. No transition constraints were imposed on the asteroid sequences (by contrast with the branch and prune approach, where an asteroid branches only into the 512 asteroids expected to represent the best transfers). The only imposed positional constraint was that, as mentioned in Section 4.1, A_1 had to be one of the asteroids for which a good launch possibility had previously been identified in the global optimization of the first leg. After generating a random sample of asteroids, the sequence would then be rotated until one of those asteroids would occupy A_1. This constraint is preserved by the crossover and mutation operators described below.

Given a sequence of asteroids in a chromosome, a trajectory is built by sequentially expressing the encoded asteroids, up to the transfer that breaks one of the constraints: $m_{A_{n+1}} \geqslant 500$ kg, and $T_{A_{n+1}} \leqslant 15$ years. The remaining asteroids in the chromosome are not expressed in the trajectory, and do not therefore contribute to the chromosome's fitness.

The costs of transfers between asteroids in a chromosome were determined as described in Section 4.1, with the difference that the transfer time $\Delta T_{i(i+1)}$ was chosen from amongst the 156 values in the range of 1 to 776 days (approximately 2.1 years), in bins of 5 days, and was the one with minimal velocity change that did not cause the maximum acceleration a_{max} to be exceeded (being a_{max} determined using a coefficient η of 1.0 – see Section 5.1, pruning criterion 1). By comparison with the branch and prune tree searches, these changes enlarge the search space with a great number of trajectories containing legs that take too long to complete or require greater accelerations. Doing so in the context of the Genetic Algorithm, however, increases the rate of discovery of good relative orderings between asteroids, that can then be used as building blocks in the construction of better trajectories.

Fitness assignment takes into account a chromosome's trajectory score, as well as the resources (mass and time) needed in order to achieve it. The best of two chromosomes is defined as being the one with greatest score. In case of a tie, it is then the one with greatest resource savings rating $r(\mathbf{x})$. A generalized mixture operator [10] aggregates the attribute vector $\mathbf{x} = [(m_f^*$ -

DOI: 10.2420/AF08.2014.45

500)/3500, (15 - T_f^*)/15], in which the final amounts of mass and time are transformed into the ratios of the available budgets for the mission left unused by the trajectory:

$$r(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x}) x_i$$
$$w_i(\mathbf{x}) = 1 - \frac{x_i}{\sum_{j=1}^{n} x_j}$$

The linear weight generating function $w_i(\mathbf{x})$ dynamically adjusts the weights assigned to each attribute, as a function of the degree to which each is being satisfied. Specifically, it implements a preference for trajectories that allocate more evenly the mission's resources. As both mass and time are critical resources that need to be made available for additional asteroids to become reachable, $w_i(\mathbf{x})$ favors improvements to the least available resource.

A steady-state population update scheme was used, which alternates between steps of parent selection, mating, and insertion of the generated offspring back into the population. Tournament selection, with a tournament size of 2, is used for selecting 2 parents for reproduction. The selected parents always undergo a crossover operation, in which a single offspring is generated. That offspring then undergoes mutation. Finally, the offspring replaces the worst solution in the population, if it has a better fitness, otherwise it is discarded.

A uniform scanning adjacency-based crossover operator was implemented. It is a variation on the operators in [11]. One offspring chromosome is sequentially constructed, by selecting asteroids from parents, in a way that explores variations on the relative orderings they encode. Throughout the construction, one index is kept per parent, pointing to the asteroid in its chromosome vector that the parent advocates as being the adequate choice to follow up from the asteroid last added to the offspring sequence. At every step, the offspring randomly selects one of the parents, and takes the asteroid it is currently pointing to. Initially, those indexes point to the first asteroid in each parent. Whenever one asteroid is added to the offspring sequence, each parent updates its index, so as to point in its chromosome to the positionally nearest asteroid to the one just added, that is not yet present on the offspring sequence. Example: considering asteroid 2 was just now added to the offspring, the parent [3, 7, 2, 6, 5, 1, 4] would set its position marker to the first asteroid in the following sequence, that is not already part of the offspring: [6, 7], [5, 3], 1, 4 (scanning forward and back from asteroid 2, with no wrapping; should two asteroids at the same distance from 2, such as 6 and 7, be viable choices, one of them is chosen at random). Should an added asteroid not occur in one of the parents, that parent keeps its position marker at its previous state.

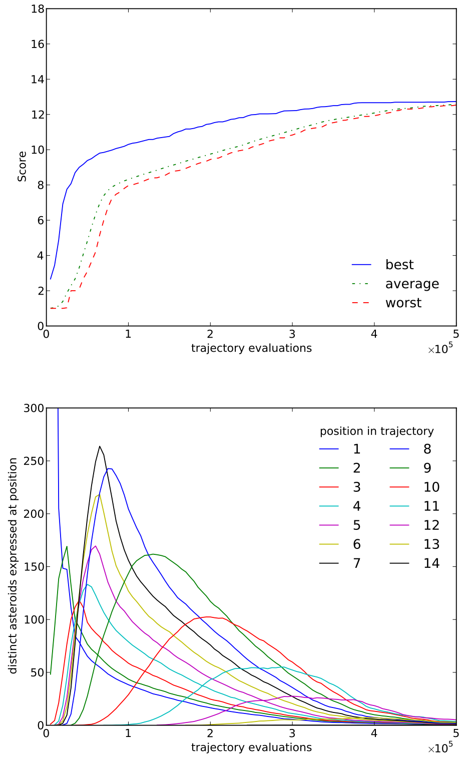

Figure 4. *Performance of the order-based Genetic Algorithm: scores (top) and diversity (bottom) of asteroid sequences in populations throughout the optimization (values averaged over 30 runs).*

The implemented mutation operator swaps two consecutive asteroids, chosen uniformly at random from among the 19 pairs in the sequence. Should the first pair be chosen, the second asteroid only takes the place of the first if it too is a valid starting asteroid.

Figure 4 shows the typical performance of a GA run (averaged over 30 independent runs) searching among the complete set of available asteroids, for a sequence with maximal score. Figure 4(top) shows the progress in the population's scores throughout the search. The average score of the best found asteroid sequence in a run was 12.7 (st. dev.: 0.6), with the best score achieved with this setup being 14.2. As individual GA runs completed on average in approximately 17 hours (running in a single processor), the GA achieved on average 75% of the quality of solutions identified in the branch and

 DOI: 10.2420/AF08.2014.45

prune tree search (where sequences of length 17 were found), in roughly 1.3% of the CPU time, while searching in a much larger search space.

Figure 4(bottom) shows the number of distinct asteroids being expressed at each position A_i across the population's 5000 trajectories. We can observe that, following the appearance of chromosomes in the population which are able to express a greater number of asteroids (increase the score), over what had previously been identified, there is an explosion in the number of asteroids being identified as viable choices for those legs. At the same time, the gradual identification of the best asteroids for a given position continually pushes for a convergence on the set of asteroids that are deemed appropriate choices at the preceding positions. As the crossover and mutation operators only make use of the asteroids contained in parents' chromosomes, it is from chromosomes' non-expressed genes that novel combinations are generated. Though non-expressed genes do not contribute to an individual chromosome's fitness, the crossover operator is able to use it as a "genetic memory" of good relative orderings that were expressed at some point in the chromosome's ancestors. Non-expressed genes are therefore equally undergoing evolution, increasing the likelihood of successful transfers being generated.

6 Back to the low-thrust world

The results from the tree search come in the form of pairs [**seq**, **mjd**] containing asteroid sequences and their encounter dates. The large number of returned trajectories is prohibitive to perform a full optimization for all of them. Thus, we further screen the solutions. This is done considering the global optimization for single asteroid-to-asteroid legs with self fly-by and trying to transform incrementally the trajectory returned by the chemical search into a real low-thrust trajectory. During the optimization, an upper bound on the arrival time of a given leg is set to the departure time of the next leg to ensure the 'continuity' of the whole trajectory. After this preliminary screening process, we obtained a list of feasible low-thrust trajectories, ranked by the number of asteroid encounters, flight time, and final mass.

Next, we attempt to optimize selected sequences as a whole trajectory by removing the bounds on the arrival and departure times. Due to the complexity of the problem, the optimizer has problems in converging in a reasonable amount of time if we optimize the whole trajectory from start. Instead we try to assemble a few (two or three) asteroid to asteroid legs at a time by removing the bounds on the middle times but keeping the bounds at the start and the end.

In the last step the full trajectory is considered as one single nonlinear programming problem. We use a low-thrust trajectory model similar to that described in [5] which is the basis of the widely used GALLOP/MALTO trajectory optimization tool [12]. In our own implementation (available in the open source python library called PyKEP), the possibility of switching between Keplerian and low-thrust propagation is present, so we first use a Keplerian propagator to exploit its speed. Only at the end we switch to a low-thrust Taylor propagation [13], roughly 5 times slower than the Keplerian, where the whole trajectory is numerically integrated to satisfy all the constraints up to the requested accuracy. In Table 2 we report a summary of the best trajectory found.

7 Lessons Learnt

The trajectory summarized in Table 2 ranked as fourth in the GTOC5 competition, a result one can be very proud of, but that has to be analyzed in the context of the overall methodology used. In particular we must remark that solutions with $N = 17$ and $N = 18$ exist for this problem, as proved by the other solutions returned by other teams, but were not found by our methodology. While we did find solutions with $N = 17$ during the chemical tree search, we were not able to transform them into low-thrust solutions. No solutions with $N = 18$ was found by our branch and prune exhaustive tree search. While it is difficult to say exactly what part of our methodologies failed, we may look to them critically as to locate their main limitations and determine good research directions.

1. The assumption that a chemical trajectory can approximate a low-thrust one is very strong and, whenever possible, should be avoided. This requires the development of more and more efficient tools for low-thrust optimization, but also a correct choice for the solution strategy, i.e. one that does require only the strictly necessary amount of trajectory legs to be computed. In the case of the GTOC5 problem, this was probably a main factor to our algorithm failure in finding any $n = 18$ solution.

DOI: 10.2420/AF08.2014.45

phase	start asteroid	end asteroid	start mass	end mass	start epoch	end epoch	leg tof	tot tof	thrust duration
1	Earth	(2007 UN12)	4000.00	3874.24	59062.81	59241.23	178.43	0.49	140
2	(2007 UN12)	(2007 UN12)	3834.24	3736.18	59241.23	59366.70	125.47	0.83	125
3	(2007 UN12)	(2006 JY26)	3735.18	3487.07	59366.70	59718.72	352.02	1.80	280
4	(2006 JY26)	(2006 JY26)	3447.07	3360.45	59718.72	59827.12	108.41	2.09	108
5	(2006 JY26)	(2001 GP2)	3359.45	3122.40	59827.12	60120.57	293.45	2.90	293
6	(2001 GP2)	(2001 GP2)	3082.40	3005.40	60120.57	60222.60	102.03	3.18	102
7	(2001 GP2)	225312 (1996 XB27)	3004.40	2706.22	60222.60	60645.66	423.06	4.33	337
8	225312 (1996 XB27)	225312 (1996 XB27)	2666.22	2610.63	60645.66	60768.01	122.35	4.67	70
9	225312 (1996 XB27)	(2003 SM84)	2609.63	2491.05	60768.01	61044.11	276.10	5.42	162
10	(2003 SM84)	(2003 SM84)	2451.05	2391.03	61044.11	61135.71	91.60	5.68	71
11	(2003 SM84)	(2006 HE2)	2390.03	2227.29	61135.71	61410.80	275.09	6.43	218
12	(2006 HE2)	(2006 HE2)	2187.29	2132.20	61410.80	61490.44	79.64	6.65	62
13	(2006 HE2)	(2007 HC)	2131.20	2006.92	61490.44	61679.56	189.12	7.16	150
14	(2007 HC)	(2007 HC)	1966.92	1916.45	61679.56	61753.71	74.15	7.37	58
15	(2007 HC)	(1998 KY26)	1915.45	1734.75	61753.71	62127.99	374.28	8.39	374
16	(1998 KY26)	(1998 KY26)	1694.75	1649.67	62127.99	62192.70	64.71	8.57	50
17	(1998 KY26)	(2009 TD17)	1648.67	1552.93	62192.70	62383.79	191.10	9.09	111
18	(2009 TD17)	(2009 TD17)	1512.93	1473.12	62383.79	62440.28	56.49	9.25	43
19	(2009 TD17)	(2007 YF)	1472.12	1345.03	62440.28	62647.84	207.56	9.82	164
20	(2007 YF)	(2007 YF)	1305.03	1271.97	62647.84	62697.49	49.65	9.95	37
21	(2007 YF)	(1999 AO10)	1270.97	1190.78	62697.49	62949.99	252.50	10.64	144
22	(1999 AO10)	(1999 AO10)	1150.78	1127.17	62949.99	63002.23	52.23	10.79	24
23	(1999 AO10)	(2006 RH120)	1126.17	1027.11	63002.23	63303.21	300.99	11.61	140
24	(2006 RH120)	(2006 RH120)	987.11	964.00	63303.21	63344.09	40.88	11.72	22
25	(2006 RH120)	(2008 UA202)	963.00	926.63	63344.09	63481.71	137.62	12.10	32
26	(2008 UA202)	(2008 UA202)	886.63	867.50	63481.71	63522.77	41.06	12.21	17
27	(2008 UA202)	(2008 HU4)	866.50	770.33	63522.77	63874.80	352.03	13.17	167
28	(2008 HU4)	(2008 HU4)	730.33	713.20	63874.80	63905.35	30.55	13.26	15
29	(2008 HU4)	(2004 VJ1)	712.20	657.25	63905.35	64061.71	156.36	13.69	54
30	(2004 VJ1)	(2004 VJ1)	617.25	603.89	64061.71	64090.16	28.45	13.76	12
31	(2004 VJ1)	(2003 WT153)	602.89	554.57	64090.16	64225.14	134.99	14.13	59
32	(2003 WT153)	(2003 WT153)	514.57	501.00	64225.14	64244.62	19.47	14.19	13

Table 2. *Summary of the best trajectory found after the full low-thrust optimization*

DOI: 10.2420/AF08.2014.45

2. Traversing the search tree using a branch and prune algorithm, implemented as a depth-first search, is not necessarily a good idea. While depth first search has the advantage of delivering immediately some solutions one can work with (a very important property in the context of the GTOC competitions), its complexity grows exponentially and thus encourages the introduction of more and more arbitrary pruning criteria that eventually risk to make the search too greedy. Strategies such as breadth first search or A^* algorithm variants should also be considered and when possible adopted. It is likely that our strict pruning criteria, necessary for the depth first search to finish in a reasonable amount of time, did not allow us to find better solutions.

3. The use of evolutionary techniques, such as the order based genetic algorithm here used, to perform a trajectory search for a multiple asteroid rendezvous mission, while not competitive with other approaches for the considered problem, showed interesting results and it is considered as a very interesting research topic.

4. Monotonic Basin Hopping, in connection to the Sims-Flanagan trajectory model may help to avoid the need for an initial trajectory guess, resulting in an unbiased exploration of the low-thrust solution space.

References

[1] I. S. Grigoriev and M. P. Zapletin, "Problem description for the 5th global trajectories optimization competition," tech. rep., Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, October 2010.

[2] M. Belló Mora and J. Cano, "1st ACT global trajectory optimisation competition: Results found at DEIMOS Space," *Acta Astronautica*, vol. 61, no. 9, pp. 794–805, 2007.

[3] A. Petropoulos, T. Kowalkowski, M. Vavrina, D. Parcher, P. Finlayson, G. Whiffen, and J. Sims, "1st ACT global trajectory optimisation competition: Results found at the Jet Propulsion Laboratory," *Acta Astronautica*, vol. 61, no. 9, pp. 806–815, 2007.

[4] C. H. Yam, D. di Lorenzo, and D. Izzo, "Low-thrust trajectory design as a constrained global optimization problem," *Journal of Aerospace Engineering*, vol. 225, no. 11, pp. 1243–1251, 2011.

[5] J. A. Sims and S. N. Flanagan, "Preliminary Design of Low-Thrust Interplanetary Missions," in *AAS/AIAA Astrodynamics Specialist Conference*, Aug. 1999.

[6] D. J. Wales and J. P. K. Doye, "Global optimization by basin-hopping and the lowest energy structures of Lennard–Jones clusters containing up to 110 atoms," *Journal of Physical Chemistry A*, vol. 101, no. 28, pp. 5111–5116, 1997.

[7] B. Addis, A. Cassioli, M. Locatelli, and F. Schoen, "A global optimization method for the design of space trajectories," *Computational Optimization and Applications*, vol. online, june 2009.

[8] F. Biscani, D. Izzo, and C. H. Yam, "A Global Optimisation Toolbox for Massively Parallel Engineering Optimisation," in *4th International Conference on Astrodynamics Tools and Techniques (ICATT 2010)*, May 2010.

[9] T. Edelbaum, "Propulsion requirements for controllable satellites," *ARS (Am. Rocket Soc.) J.*, vol. 31, 1961.

[10] R. A. M. Pereira and R. A. Ribeiro, "Aggregation with generalized mixture operators using weighting functions," *Fuzzy Sets and Systems*, vol. 137, no. 1, pp. 43–58, 2003.

[11] A. E. Eiben, P.-E. Raué, and Z. Ruttkay, "Genetic algorithms with multi-parent recombination," in *Parallel Problem Solving from Nature – PPSN III* (Y. Davidor, H.-P. Schwefel, and R. Männer, eds.), vol. 866 of *Lecture Notes in Computer Science*, pp. 78–87, Springer Berlin / Heidelberg, 1994.

[12] C. H. Yam, D. Izzo, and F. Biscani, "Towards a High Fidelity Direct Transcription Method for Optimisation of Low-Thrust Trajectories," in *4th International Conference on Astrodynamics Tools and Techniques (ICATT 2010)*, May 2010.

[13] A. Jorba and M. Zou, "A Software Package for the Numerical Integration of ODEs by Means of High-Order Taylor Methods," *Experiment. Math.*, vol. 14, pp. 99–117, 2005.