IAC-21-C1.5.7 (64234)

Efficiency of tree-search like heuristics to solve complex mixed-integer programming problems applied to the design of optimal space trajectories

A. Bellome[a*], M. Carrillo[a*], J.P. Sánchez[a], J. Del Ser[b,c], S. Kemble[a] and L. Felicetti[a]

[a] *Space Research Group, Centre of Autonomous and Cyber-Physical Systems, Cranfield University, UK,* *[*andrea.bellome@cranfield.ac.uk](mailto:andrea.bellome@cranfield.ac.uk); [*maria.carrillo-barrenechea@cranfield.ac.uk](mailto:maria.carrillo-barrenechea@cranfield.ac.uk)*
[b] *TECNALIA, Basque Research and Technology Alliance (BRTA), 48160 Derio, Spain*
[c] *Department of Communications Engineering, University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain*
* Corresponding Author

Abstract

In the past, space trajectory optimization was limited to optimal design of transfers to single destinations, where optimality refers to minimum propellant consumption or transfer time. New technologies, and a more daring approach to space, are today making the space community consider missions that target multiple destinations.

In the present paper, we focus on missions that aim to visit multiple asteroids within a single launch. The trajectory design of these missions is complicated by the fact that the asteroid sequences are not known a priori but are the objective of the optimization itself. Usually, these problems are formulated as global optimization (GO) problems, under the formulation of mixed-integer non-linear programming (MINLP), on which the decision variables assume both continuous and discrete values. However, beyond the aim of finding the global optimum, mission designers are usually interested in providing a wide range of mission design options reflecting the multi-modality of the problems at hand. In this sense, a Constraint Satisfaction Problem (CSP) formulation is also relevant.

In this manuscript, we focus on these two needs (i.e. tackling both the GO and the CSP) for the asteroid tour problem. First, a tree-search algorithm based upon the Bellman's principle of optimality is described using dynamic programming approach to address the feasibility of solving the GO problem. This results in an efficient and scalable procedure to obtain global optimum solutions within large datasets of asteroids. Secondly, tree-search strategies like Beam Search and Ant Colony Optimization with back-tracking are tested over the CSP formulations. Results reveal that BS handles better the multi-modality of the search space when compared to ACO, as this latter solver has a bias towards elite solutions, which eventually hinders the diversity needed to efficiently cope with CSP over graphs.

Keywords: asteroids; dynamic programming; global optimization; constraint satisfaction problem; space trajectory design

1. Introduction.

Instances of mission design problems that target multiple orbital waypoints, instead of only one final destination, represent a growing trend. Some examples of this include, but are not limited to, ESA's JUICE mission [1], which will perform more than twenty gravity assist manoeuvres with Jovian moons, commercial concepts for Active Debris Removal, which also consider to rendezvous with multiple spacecraft by means of one single platform to help restoring the Earth's orbital environment (e.g. see [2]), as well as CASTAway concept [3], [4] on which multiple asteroid rendezvous trajectories were presented in the context of ESA's Medium Class mission 2016 call.

In particular, this paper focuses on the mission design of a tour within the asteroid main belt. There is clear scientific interest in exploring this region of the Solar System, to better understand the composition and evolution from early stages of the Solar System. The aim is to pass-by as many asteroids as possible to obtain key information about the composition of objects by close encounter analysis [3]. A mission that can fly-by at least 10 asteroids will double the number of objects visited to date. The present work focuses on missions of 12 asteroids. It should be noted that the orbital waypoints (or targets) to be visited are not known a priori but are the objective of the optimization problem itself. Therefore, most of the approaches to design these missions make use of three steps [5]: the first one consists in defining a subset of potential asteroids to be visited, based on their orbits and scientific characteristics. In the second step, a sequence of objects is found by means of global optimization algorithms. In the last step, the trajectories between asteroids are optimized with either local or global optimization. This scheme is also adopted in the present work.

Designing missions that visit many asteroids is a notoriously challenging problem to solve. They are usually transcribed in some form of global optimization, referred to as Mixed-Integer Non-Linear Programming (MILP) problems [6], [7], also known in literature as Hybrid Optimal Control Problems (HOCP) [8]. A general MINLP/HOCP presents the following structure:

$$
\begin{array}{ll}
\text{Minimize} & f(x,y) \text{ where } x \in R^{n,cont} \text{ and} \\
& y \in Z^{n,int}, n_{cont}, n_{int} \in N \\
\text{Subject} & g_i(x,y) = 0, i = 1, \dots, m_e \in N \\
\text{to} & g_i(x,y) \geq 0, i = m_e + 1, \dots, m \in N \\
& x_{lb} \leq x \leq x_{ub} \text{ where} \\
& x_{lb}, x_{ub} \in R^{n,cont} \\
& y_{lb} \leq y \leq y_{ub} \text{ where} \\
& y_{lb}, y_{ub} \in R^{n,int}
\end{array}
\quad \textbf{(1)}
$$

Where: $f(x,y)$ is the objective function to be minimized (usually the propellant to be stored on the spacecraft); (x,y) are the decision variables, which belong to the continuous and integer domain, respectively; $g_i(x,y)$ represents the constraints of the problem at hand (usually the overall mission duration), where m_e and m are the cardinalities for equality and inequality constraints, respectively; (x_{lb}, x_{ub}) and (y_{lb}, y_{ub}) represent the box constraints (i.e. lower and upper bounds) for (x,y).

As such, these problems have been the target of many editions of the Global Trajectory Optimisation Competition (GTOC) [9]. There is an obvious need to manage the increasing complexity of the formulation of these problems, often with cleverly designed strategies to prune out the search space or manage the search strategy (see [10] for references about methods employed in past GTOCs).

Techniques to tackle MINLP can be divided into deterministic and stochastic ones. Deterministic algorithms are well established methods in literature [11], [12], among which Branch and Bound (BB) [13] is one of the most common ones. Among stochastic metaheuristics approaches applied to MINLP, Genetic Algorithms (GA) [14], [15], Particle Swarm Optimization (PSO) [16] and Ant Colony Optimization (ACO) [17] are popular choices. Stochastic algorithms present the advantage that convexity and linearity of $f(x,y)$ and/or $g(x,y)$ functions are not required, as well as the relaxation or separation of discrete variables. Another advantage is that they do not require gradients to know where to go along the search, but rather replace the gradient information with a self-learning procedure. However, they will generally require higher number of function evaluations due to their stochastic nature [6]. There is a substantial literature addressing the asteroid tour problem, involving both deterministic strategies (see, for example, Grigoriev and Zapletin [18], Izzo et al. [19], Di Carlo et al. [20] and Petropoulos et al. [21]) and stochastic metaheuristics or other bio-inspired approaches (see, for example, Sanchez et al. [4] Simões et al. [22], and Fan et al. [23]).

However, beyond the challenge to find the global optimum of a complex MINLP applied to space trajectory design (STD), thereafter referred as MINLP-STD, practical mission feasibility studies for asteroid tour missions (e.g. see Bowles et al. [3] and Sanchez et al. [4]) require an accurate description of the full topology of the feasible search space, rather than the identification of a global optimum. Hence, are more akin to a Constraint Satisfaction Problem (CSP) rather than to a global optimization problem, to account for the multi-modality of the problem at hand [5]. In the context of asteroids tours, a CSP is interesting because the range of different possible asteroid tours is relevant for scientists within an indicative spacecraft design limit (e.g. Δv related). Moreover, in the case of CASTAway-like missions [3], [4], obtaining the CSP solutions is based on a set of integer variables describing the asteroid tour all of which result in broadly similar spacecraft operational environment, thus Pareto optimality studies are less interesting compared to other mission types.

A general definition of the CSP can be stated as follows:

Given a set of variables:	$X = (x, y)$ where $x \in R^{n,cont}$ and $y \in Z^{n,int}$, $n_{cont}, n_{int} \in N$	
And their domains:	$x_{lb} \leq x \leq x_{ub}$ where $x_{lb}, x_{ub} \in R^{n,cont}$ $y_{lb} \leq y \leq y_{ub}$ where $y_{lb}, y_{ub} \in R^{n,int}$	**(2)**
And a set of constraints:	$C = C_1(x,y), \ldots, C_m(x,y)$, $m \in N$	

Find (one or more) assignments of a value for each variable such that the assignments satisfy all the constraints.

Compared to the global optimization problem defined in equation (1), a CSP as in (2) considers the functions $f(x,y)$ and $g(x,y)$ as part of the set of constraints C.

Therefore, there are clearly two needs that should be addressed when designing such missions. The first one is to find the actual global optimum solution of the problem at hand as described in (1). The second one is to solve a CSP as in (2), to provide adequate extent of mission design options, usually required in preliminary design.

The present paper thus approaches the asteroid tour problem from these two separated, though related, formulations. It first describes the strategy to transcribe the problem from a mixed formulation to a pure combinatorial one (see section 2). Section 3 also provides details about the structure of the search space associated to the problem at hand. In section 3, the paper presents a tree-search strategy based upon the Bellman's principle of optimality to obtain global optimum solution in an efficient manner, thus solving the problem as described in (1). Section 5 then approaches the problem as a CSP as in (2), introducing tree-search strategies to tackle it and a formulation of ACO with back-tracking for handling the constraints. Section 6 concludes the paper.

2. Main Asteroid Belt Tour Exploration.

The term *tour*, in the context of a space mission, simply indicates a mission that aims to visit not one but several celestial objects. There have already been multiple propositions of asteroid tour mission problems; such as those in 5 out of 11 editions of GTOC* (e.g., [24], [25]), or in more mission design oriented studies, such as in [4], [20], [26]. Any of these examples will imply many constraints and boundaries that are specific to the problem, however all will require dealing with comparable complexities on a similar dynamical framework.

The asteroid tour problem tackled here is relevant to the CASTAway mission concept submitted for consideration to ESA's 5th medium size mission call 6. The Comet and Asteroid Space Telescope Away in the asteroid belt (i.e. CASTAway) proposed a mission dedicated to gain understanding of the main asteroid belt (MAB), to inform solar system formation and evolution theories. CASTAway spacecraft was to carry a small telescope capable to observe asteroids both at a long-range (i.e. point source survey), as well as at a short-range, resolving them at ~10 m resolution during asteroid fly-bys. Full details on CASTAway mission design can be found in Sanchez et al. [4]. Here, however, so that the focus is on the methods to solve the problem itself rather than on the definition of the problem, suffices to say that the spacecraft is deployed into a direct escape trajectory (with limited range of possible v_∞); it needs to encounter as many main-belt asteroids as possible, while limiting its Δv and time of flight (*tof*) to values below a maximum threshold.

The problem is defined as fixed size tour with n_{max} asteroids (12 in this paper). The ambition would thus be to find as many feasible tours of n_{max} asteroids as possible, ideally including the global minimum (i.e. tackling both (1) and (2)). The function $f(x,y)$ takes the sequence of asteroids encoded in vector y (also accounting for Earth departure) and the visiting epochs

* https://sophia.estec.esa.int/gtoc_portal/

encoded in vector x and solves Lambert arc transfers between each pair of consecutive objects (e.g. Earth-*Asteroid*$_1$, *Asteroid*$_1$-*Asteroid*$_2$, etc). Two timestamps are necessary to solve the Lambert arc, which are defined by the vector of continuous real values design parameters y. Lambert arcs and velocity vector addictions are assumed for the dynamics of the spacecraft and for the computation of the Δv, respectively. Asteroid mass is neglected, however planetary swing-bys are possible by inserting a planet, rather than an asteroid in the sequence. Note each planetary swing-by would require three more real value parameters to define the hyperbolic passage. However, in this paper, the strategy described by the authors in [27] is employed. This is based upon approximated Δv at the swing-by, computed based upon the visiting epoch of the objects in the sequence. The function $f(x, y)$ finally returns the cost of the transfer in terms of the total Δv.

The preliminary design of the CASTAway trajectory is driven by the need to spend as much time as possible within the MAB, while limiting its *tof* and overall Δv. Maximising the time within MAB allows to ensure good survey conditions, as well as increased chances to come across fly-by opportunities. The original CASTAway searched trajectories that would complete the asteroid tour in 6 years in order to limit operational costs. As seen in Gallego [28] for *tof*<8 years, a simple Earth-Mars (EM) sequence is optimal to accumulate a high time fraction within the MBA. Hence the trajectories considered thereafter have a fixed EM sequence, where the Mars swing-by is used to increase the spacecraft reach within the MBA.

Figure 1 represents an example trajectory for CASTAway, which follows an EM sequence and encounter a total of 10 asteroid. As of September 2021, 1 million MAB objects are known. However, here a pre-pruned database of ~102,000 main belt asteroids is used instead. This database provides a prefiltered population with a diversity of asteroids in size and orbital distribution. Details on the pre-filtered process can be found in Sanchez et al. [4].

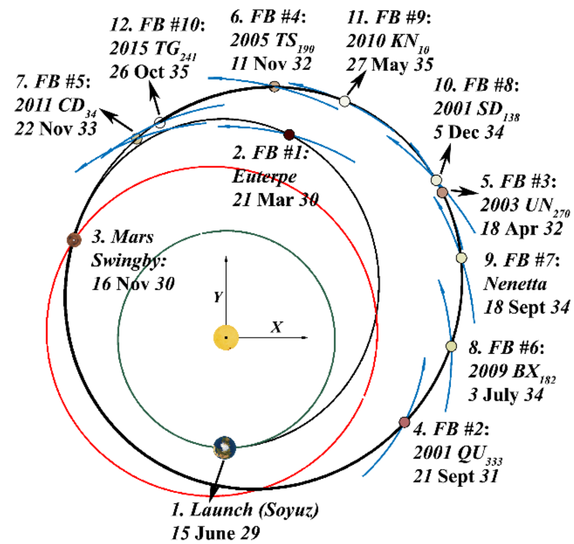

Figure 1. Example of main-belt asteroid tour trajectory for CASTAway design envelop.

Nevertheless, it should be noted that an asteroid set of ~102,000 objects is already a much larger set than in any asteroid tour related GTOC competition problem. Also, exploring, for example, all possible 12-asteroids tours from a set of 102,000 objects would represent 10^{60} tours to compute and a computing time likely exceeding the age of the universe. Hence, the key to solve this problem must lay on managing this complexity efficiently.

A potential way to manage this complexity is to find a process on which the mixed problem as in (1) and (2) can be transcribed in a pure combinatorial one. This process, as implemented in the present paper, employs a pre-optimization phase making specific mission-relevant assumptions. The aim is to generate a reference trajectory that maximises the number of asteroids visited with an approach distance of less than a specified upper limit. Limits used in this paper are 0.03 AU, 0.04 AU and 0.05 AU.

Minimum Orbital Interception Distances (MOIDs) between the spacecraft reference orbit (i.e. an Earth-Mars transfer) and each asteroid in the reduced set are computed. The mixed problem is then uncoupled by assuming asteroid fly-bys occurring at the asteroid MOID point. The tour of the MAB is assumed to start on the 24th of December 2030 as a result of an optimization process to maximise the time in the

MAB within 7 years of overall mission duration (considering the Earth-Mars reference trajectory).

A paramount feature of the proposed solution is the closeness of the Δv estimate from the combinatorial part of the problem to the actual optimal value after the trajectory refinement by means of global/local optimizers. Several test cases were performed for 1e6 randomly generated sequences with 12 asteroids on which the visiting epochs were also assigned randomly, and the best cost obtained were no lower than ~300 km/s. When introducing the MOID information, after 1e6 randomly generated 12-asteroids sequences the best cost was ~9.3 km/s. This is a strong indication of the merit of the sub-optimal solutions coming from the combinatorial part of the problem.

In addition, once the sequences have been obtained from combinatorial optimization, a standard genetic algorithm solver available within MATLAB global optimization toolbox was used to refine the asteroids encounter dates. Figure 2 confirms the efficiency of the proposed methodology. It shows the ~7500 solutions for which both the combinatorial and the continuous part were solved. In particular, it represents the optimized Δv cost of different sequences with respect to their guesses based only the combinatorial solutions of the problem. This clearly shows how the approximations made with MOID estimations are able to quickly provide sub-optimal solutions.

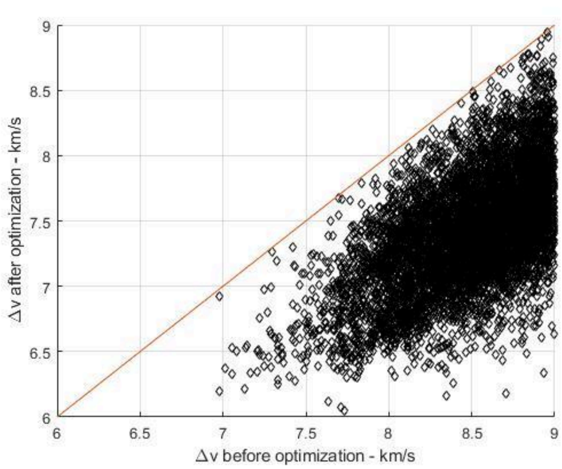

Figure 2. Summary of Δv solutions before and after the continuous optimization of sequences.

3. Graph structure of the search space.

The procedure described in section 2 has provided an ordered list of asteroids, with respect to the MOID epoch, which can potentially be reached. In this way, one has uncoupled the combinatorial part of the problem, associated to the selection of the optimal sequence of asteroids to be visited, from the continuous part, associated to the optimization of the visiting epochs. This section focuses on the combinatorial part of the asteroid tour problem. Combinatorial problems (CP) are often modelled with a search space that is a grid of connected nodes. A very common example of CP is the Travelling Salesman Problem (TSP) that is about a salesperson that has to visit a given number of cities, i.e. the nodes, which are connected by roads or paths of a given length. The salesperson has to start from his hometown, visit each city once and return back home. As an optimisation problem, the cheapest (in fuel consumption) or shortest (in time of travel) path or tour wants to be found.

Moving the TSP example to the problem in hand, the cities are the asteroids, and each combination of asteroids will have a cost which can be the Δv or the *tof*, or a combination thereof, which normally wants to be minimised. The cost can be calculated by a given function which will be the objective function or fitness function to be minimised. However, the main difference between the standard TSP and the asteroid tour problem lays in the cost connecting two objects. With the structure of the problem defined in section 2, each leg of the asteroid tour mission is a Lambert arc whose cost will be the required change in velocity Δv resulting from the difference between the velocity vector at the arrival and departure at a given asteroid.

Figure 3 illustrates the spacecraft trajectory between two asteroids of MOID indices j and k with the arrival velocity $\vec{v}_{j,arrival}$ and the departure $\vec{v}_{j,departure}$from asteroid j and the Δv of the leg given as $\Delta v_{leg(jk)} = |\vec{v}_{j,departure} - \vec{v}_{j,arrival}|$. In this way, the cost of a given asteroid-to-asteroid leg is not unique, but it depends upon the asteroid prior to that leg, by means of $\vec{v}_{j,arrival}$.

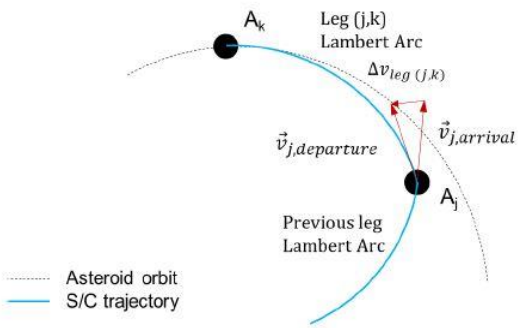

Figure 3. 2D sketch of spacecraft trajectory and Δv between asteroid j and k.

Thus, to uniquely define the cost of a given leg A_j and A_k, one needs to consider also the previously visited one, say A_i, so for the triplet (A_i, A_j, A_k), one has the unique cost of Δv^i_{jk}. Because of this tri-dependency, the search space is a graph that can be modelled as a multi-dimensional space of connected subspaces, being the search space a search grid $\mathcal{G}$. These subspaces $S_i \subseteq \mathcal{G}$ contain nodes which are pairs of asteroids. The initial subspace S_0 contains all the pairs of asteroids (A_0, A_i) $\forall i \in \mathbb{N} \mid 1 < i \leq n_{ast}$, where n_{ast} is the number of asteroids. S_0 is connected to a set of subspaces S_i $\forall i \in \mathbb{N} \mid 0 < i \leq n_{ast}$ each of which are again connected to a set of subspaces S_j $\forall j \in \mathbb{N} \mid i < j \leq n_{ast}$ and so on. The total number of subspaces in $\mathcal{G}$ is n_{ast} as the last subspace is $S_{nast\text{-}1}$ containing node ($A_{nast\text{-}1}$, A_{nast}). When choosing a node, the first asteroid in the node will be equal to the second asteroid in the previous node. This is how the subsets are connected among themselves. In this space, the cost of the paths between the nodes are unique, which is the main advantage of modelling the search space in this way.

Being the Δv^i_{jk} a tri-asteroids dependent cost, unique for each of the legs of the search space, a Score Matrix (SM) with the following characteristics was created for this research:

- Tri-structured: the matrix relates the pair of nodes of departures in rows with the asteroids of arrival in columns
- Unique: all the costs contained within the matrix are constant due to its structure

Strictly upper-triangular form pattern repetition: if the SM was asteroid-to-asteroid, the matrix would be a strictly upper-triangular one, as all the main diagonal and lower diagonal are non-feasible paths. As the rows of the matrix are pairs of nodes, this pattern is repeated downwards through the matrix.

Figure 4 shows an example of SM for a set of 158 asteroids. Each row represents a couple of asteroids, i.e. a trajectory between two asteroids, and each column is encoded with asteroids in the catalogue that complete the triplet. Anytime it is not possible to define a Δv between two asteroids, an 'x' is represented (i.e. the transfer is not possible). This happens when MOID epochs for the asteroids in the triplets are not consecutive.

From \ To	A_1	A_2	A_3	A_4	A_5	$\cdots$	A_{156}	A_{157}	A_{158}
(A_0, A_1)	x	$\Delta v^0_{1,2}$	$\Delta v^0_{1,3}$	$\Delta v^0_{1,4}$	$\Delta v^0_{1,5}$	$\cdots$	$\Delta v^0_{1,156}$	$\Delta v^0_{1,157}$	$\Delta v^0_{1,158}$
(A_0, A_2)	x	x	$\Delta v^0_{2,3}$	$\Delta v^0_{2,4}$	$\Delta v^0_{2,5}$	$\cdots$	$\Delta v^0_{2,156}$	$\Delta v^0_{2,157}$	$\Delta v^0_{2,158}$
(A_0, A_3)	x	x	x	$\Delta v^0_{3,4}$	$\Delta v^0_{3,5}$	$\cdots$	$\Delta v^0_{3,156}$	$\Delta v^0_{3,157}$	$\Delta v^0_{3,158}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
(A_0, A_{156})	x	x	x	x	x	$\cdots$	x	$\Delta v^0_{156,157}$	$\Delta v^0_{156,158}$
(A_0, A_{157})	x	x	x	x	x	$\cdots$	x	x	$\Delta v^0_{157,158}$
(A_0, A_{158})	x	x	x	x	x	$\cdots$	x	x	x
(A_1, A_2)	x	x	$\Delta v^1_{2,3}$	$\Delta v^1_{2,4}$	$\Delta v^1_{2,5}$	$\cdots$	$\Delta v^1_{2,156}$	$\Delta v^1_{2,157}$	$\Delta v^1_{2,158}$
(A_1, A_3)	x	x	x	$\Delta v^1_{3,4}$	$\Delta v^1_{3,5}$	$\cdots$	$\Delta v^1_{3,156}$	$\Delta v^1_{3,157}$	$\Delta v^1_{3,158}$
(A_1, A_4)	x	x	x	x	$\Delta v^1_{4,5}$	$\cdots$	$\Delta v^1_{4,156}$	$\Delta v^1_{4,157}$	$\Delta v^1_{4,158}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
(A_2, A_3)	x	x	x	$\Delta v^2_{3,4}$	$\Delta v^2_{3,5}$	$\cdots$	$\Delta v^2_{3,156}$	$\Delta v^2_{3,157}$	$\Delta v^2_{3,158}$
(A_2, A_4)	x	x	x	x	$\Delta v^2_{4,5}$	$\cdots$	$\Delta v^2_{4,156}$	$\Delta v^2_{4,157}$	$\Delta v^2_{4,158}$
(A_2, A_5)	x	x	x	x	x	$\cdots$	$\Delta v^2_{5,156}$	$\Delta v^2_{5,157}$	$\Delta v^2_{5,158}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
(A_{156}, A_{157})	x	x	x	x	x	$\cdots$	x	x	$\Delta v^{156}_{157,158}$
(A_{157}, A_{158})	x	x	x	x	x	$\cdots$	x	x	x

Figure 4. Example of SM for 158 asteroids. Rows are couples of asteroids, while columns are asteroids that conclude the triplet. An 'x' is included whenever it is not possible to define a Δv for the triplet.

Let *N* be the set of feasible solutions within the search space grid $\mathcal{G}$. With the aim of reducing the computational cost of the search, a reduction of the search space dimensions is done by removing all the elements that are not included in *N*. The cleaning of non-feasible solutions of the search space is done considering the following constraints:

- REQ-001: sequences shall contain $n_{max} = 12$ asteroids of increasing MOID
- REQ-002: the asteroid-to-asteroid Δv shall be of 1 km/s as maximum
- REQ-003: the overall cost of a sequence shall be of 9 km/s as maximum
- REQ-004: the cost of the first node, i.e. Earth-Asteroid1, shall be of 5 km/s as maximum

The first constraint REQ-001 implies some asteroids cannot be solutions of the sequence depending on their position *m* in the sequence. Thus, the set of feasible solutions considering both constraints is:

$$N = \left\{ \{A^0, A^1, \dots, A^m, \dots, A^{n_{max}}\} \middle| \begin{matrix} A^m \leq n_{ast} - (m - n_{max}) \\ \Delta v_{tot} \leq \Delta v_{max} \end{matrix} \right\} \quad (3)$$

Where Δv_{tot} is the sum of the Δv of all the legs that compose 12-asteroids sequence, adding the Earth-first asteroid Lambert arc cost $\Delta v_{A^0,A^1}$:

$$\Delta v_{tot} = \Delta v_{A^0,A^1} + \sum_{m=2}^{n_{max}} \Delta v_{A^{m-1},A^m}^{A^{m-2}} \quad (4)$$

However, as the search space nodes are modelled as pairs of asteroids, the only pre-cleaning that can be done for the solver is given by:

$$\mathcal{G}_c = \left\{ (A_i, A_j) \in \mathbb{N} \middle| \begin{matrix} j \leq n_{ast} - (i + l) \\ \Delta v_{jk}^{i} < \Delta v_{max} \forall k \in \mathbb{N} | (j+1, n_{ast}) \end{matrix} \right\} \quad (5)$$

Where $\mathcal{G}_c$ is the cleaned search space. For the second constraint, REQ-002, it is ensured that feasible solutions are not removed from $\mathcal{G}_c$ by marking as non-feasible the asteroid legs with $\Delta v \geq 1\ km/s$. If none of the possible triplets from a pair (A_i, A_j) comply this criterion, the pair is removed from $\mathcal{G}_c$.

4. Dynamic programming for asteroid tour problem.

To solve the first instance of mission designs towards asteroids as stated in equation (1), one is interested in finding the global optimum solution to the problem at hand. Provided the transcription of the problem from a mixed formulation as in section 2, one needs now to obtain the global optimum sequence with respect to the following cost function:

$$f(\bar{x}, y) = \Delta v_{tot} = \Delta v_{A^0,A^1} + \sum_{m=2}^{n_{max}} \Delta v_{A^{m-1},A^m}^{A^{m-2}} \quad (6)$$

Where the vector $\bar{x}$ encoding the visiting epochs is known from the procedure in section 2. Thus, the function f depends only upon the asteroids sequence.

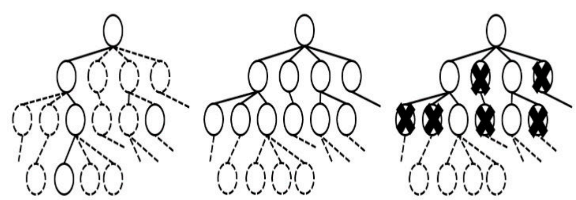

Figure 5. Comparison of tree search strategies: best-first (left), depth-first (middle) and beam search (right). Dotted nodes are not explored yet. Crossed nodes do not fall within the beam width and thus are pruned.

MINLP-STD can be often modelled as some form of tree search [19], where each node represents a transfer that can be incrementally constructed expanding one or more of its branches (i.e. adding a trajectory leg). Among tree-search algorithms that employ a deterministic branching procedure, the Depth-First (DF) and Breadth-First (BF) [29] are the most popular. These are known to be complete strategies, i.e. they allow to obtain the global optimum by evaluating all the possible branches of the tree. The difference between the DF and the BF lays on the way they perform the branching: DF explores as far as possible along each branch, while BF explores all nodes at the given tree depth prior to moving on to nodes at the next level (see also Figure 5). However, while DF/BF guarantee the global optimality of the solution, these strategies usually result impractical for MINLP-STD, as the design space might be too large that exhaustive searches require infeasible computational time. The total number of possible different permutations N_{eval} of a set n_{asts} asteroids would be:

$$N_{eval} = \binom{n_{asts}}{n_{max}} = \frac{n_{asts}!}{n_{max}!\,(n_{asts} - n_{max})!} \tag{7}$$

Where n_{asts} is again the number of asteroids in the catalogue and n_{max} is the length of the sequence. Hence, evaluating all the potential combinations of a set n_{asts} of 158 asteroids (tours of n_{max} with 12 asteroids) would already take 2 million years to standard laptop (i.e., i7-7500U CPU) (see also Table 1and Table 2).

Table 1. Number of permutations and computational times for different configurations of the asteroid tour problem.

Earth-Mars Baseline trajectory	Departing epoch: 24th of December 2030 Mars flyby: 3rd of March 2033	
Number of asteroids	N_{eval}	**Time full exploration [h]**
3	-	-
20	125,970	0.1
49	$9\text{x}10^{10}$	7000
98	$8\text{x}10^{14}$	$6\text{x}10^{8}$
158	$3\text{x}10^{17}$	$2\text{x}10^{11}$

Table 2. Time needed for BFS and DFS with respect to the number of asteroids in the catalogued. An exhaustive search is not feasible for catalogues greater than 49 asteroids, thus a '/' has been included.

Number of asteroids	**Time BFS [h]**	**Time DFS [h]**
3	-	-
20	0.005	$4.5\text{x}10^{-4}$
49	0.16	0.16
98	/	/
158	/	/

To mitigate the aforementioned curse of dimensionality associated to exhaustive DF/BF searches, a dynamic programming approach to the asteroid tour problem has been developed. A dynamic programming algorithm benefits from a description of optimization problems such that Bellman's principle of optimality [30] is true or holds. Such principle states that an optimal policy has the property to be independent from initial state and initial decisions. This means that the optimal policy is the same even if the optimal control is found starting at intermediate states. Here, by optimal policy is meant a sequence of decisions which is the most advantageous from a preassigned criterion.

Translated to the asteroid tour problem, Bellman's principle of optimality would state that, regardless of the asteroid at what the spacecraft is, the optimal set containing this specific asteroid would contain the optimal subset of asteroids before and after the visited one. Here, optimality is referred to minimum propellant consumption. This principle thus allows to apply tree traverse algorithms like DF/BF to at more local level.

Since the whole trajectory is modelled with consecutive Lambert arcs between pair of objects in the sequence, the Δv cost for a given asteroid-to-asteroid leg depends upon the previously visited object (see also section 3). Thus, the Bellman's principle is applicable only considering an intermediate couple of asteroids as initial nodes. It is thus possible to uniquely define the global optimum sub-sequences before and after the two objects by expanding the tree of possible reachable asteroids forward and backward in time.

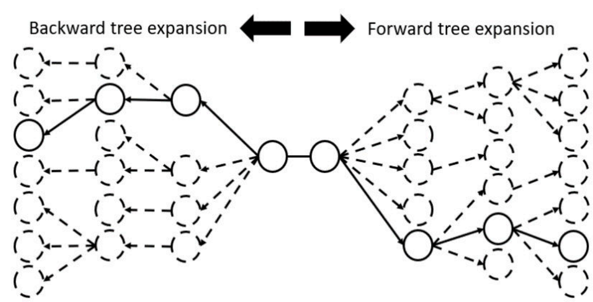

Figure 6. Backward and forward tree expansion for a given intermediate couple of asteroids. Bold nodes and branches belong to optimal sequence.

Figure 6 illustrates the application of Bellman's principle of optimality applied to the asteroid tour problem tree search. Once the intermediate couple is defined, the optimal sub-sequences can be obtained in backward and forward sides, and thus the global optimum sequence containing the given intermediate couple. In order to obtain the overall global optimum trajectory, one needs to scan every single couple of objects within a given catalogue of asteroids.

Nevertheless, this approach only allows to locate the global minimum, while the curse of dimensionality is still rendering the global optimization problem intractable for large number of asteroids. This is because the number of couples increases factorially with the number of asteroids in the catalogue. Moreover, the expansion of the sub-trees in the backward and forward directions can still grow with the number of asteroids in the catalogue. In this paper, the intermediate asteroids are assumed to be in 7th and 8th position of a sequence of 14 objects (i.e. 12 asteroids + Earth and Mars). Thus, Table 3 shows how the number of couples varies with the number of asteroids in the catalogue.

Table 3. Couples to be explored with respect to number of asteroids in a catalogue.

Number of asteroids	Couples to be explored
49	951
98	4283
158	11633

Some strategies can be adopted to mitigate the curse of dimensionality, like hybridizing with some tree-strategy like the Beam Search (BS) or ACO, thus sacrificing the global optimality assurance in the name of the computational effort. For larger catalogues, lower bound analysis by means of BS can be used to reduce the value of REQ-003 from section 3 from $9\ km/s$ to lower values. One can even split the sequence in multiple segments, and not only two as done in the present work, but this is out of the scope of the present paper and will be the focus of future research.

To further reduce the computational effort, once one of the two sides of the tree has been expanded (see again Figure 6), the available Δv is updated by excluding the one used for the constructed sub-sequence. In this way, the global optimum is obtained.

Table 4. Computing time and optimal costs with respect to number of asteroids in a catalogue.

Number of asteroids	Computing time [h]	Best cost [km/s]
49	0.16	7.247
98	1	6.977
158	4	6.977

Table 4 shows the computational time and the cost for the global optima solutions obtained with dynamic programming algorithm. Note that the global optimum of the catalogue with 0.04 AU and the 0.05 AU thresholds coincide, as they correspond to the same sequence. This is because all the asteroids present in small catalogues are also contained in bigger ones. One should notice how the curse of dimensionality is acting in terms if computational effort when large catalogues are tackled.

Details of the asteroids visited, times of flybys and Δv are provided in Table 5 and a graphical representation is given in Figure 7. These have been obtained for the largest catalogue explored, i.e. the one with 158 asteroids.

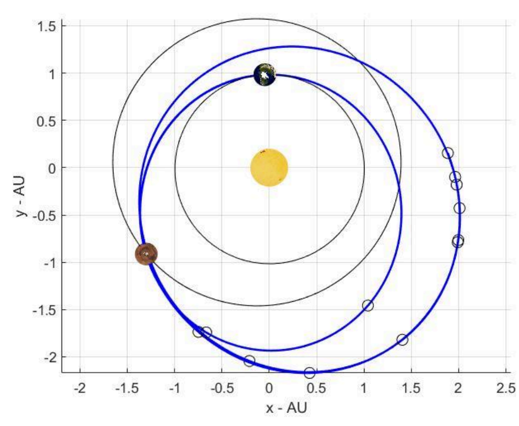

Figure 7. Global optimum tour of 12 asteroids within the catalogue of 158 objects.

Table 5. Global optimum solutions of the main belt tour for the catalogue of 158 asteroids.

Objects names and IDs	Δv [km/s]	*tof* [days]
Earth	-	-
2	4.309	247.79
14	0.0600	179.83
Mars	0.211	372.37
17	0.0366	77.63
33	0.425	118.76
62	0.158	236.69
75	0.119	61.47
100	0.165	484.80
118	0.150	173.13
129	0.101	125.73
135	0.406	34.57
141	0.268	22.78
152	0.567	29.95
TOTAL	6.9772	2165.5

5. Tree-search strategies for the Constraint Satisfaction Problem.

Apart from the global optimum solution, a mission designer is also interested in finding locally optimal solutions to provide an extent of trajectory design options, useful in preliminary stages of the design. This is similar to solve the CSP defined in equation (2), on which there is no cost function to be minimized, but just a series of constraints to be satisfied. In this case the function $f(\bar{x}, y)$ as defined in equation (6) becomes part of the set of constraints. The set of constraints are thus defined by REQ-001, REQ-002, REQ-003 and REQ-004 in section 3. Algorithms that are suitable for tackling a CSP are again tree-search strategies, also involving some back-tracking scheme [31]. In the present paper, we examine BS and ACO. In particular, we develop a version of ACO employing a back-tracking to improve the chances of finding solutions with respect to the CSP at hand.

The BS [32], [33] is a deterministic tree search algorithm firstly presented in the context of scheduling [34]. In BS algorithms, the computational effort is bounded by employing heuristics that prevent the exploration of non-promising branches. It can be executed as a variant DF or BF strategies (refer again to Figure 6). In the present paper, the BS is implemented as a variant of the BF. In this way, the exploration of possible trajectory options is performed one depth-level at a time. From all the branches generated in one level, only a limited set of it, i.e., the beam, is selected to be expanded in successive nodes. The size of the beam is called beam width (BW). A search where $BW = 1$ corresponds to a nearest-neighbour (NN) search [2], on which the expansion of the tree is done towards the most promising node, discarding all the others. A search where $BW = \infty$ corresponds to a comprehensive DF/BF search, on which all the possible solutions are evaluated.

Among tree-search strategies that employ a stochastic branching procedure, ACO is one of the most popular for solving complex combinatorial problems. This is a metaheuristic algorithm that mimics the behaviour of ant colonies in searching for food to solve complex combinatorial problems. Although in current literature there exist many variants, this algorithm was first proposed in the literature by Dorigo and Gambardella [35]. In ACO, some agents called artificial ants construct candidate solutions in a discrete graph, whose nodes are equivalent to the cities in the TSP. Compared to the BS, the branching procedure is stochastic and guided by the combination of the heuristic information (usually associated to the goodness of a specific city-to-city transfer) and what is called the pheromone, i.e. feedback laid by precedent ants when constructing candidate solutions upon the goodness of the explored tour. Thus, for the asteroid tour problem, the probability of an ant selecting a node (j, k) depends upon the pheromone $\tau_{(i,j),k}$ and the heuristic parameter $\eta_{(i,j),k}$ both associated to the path between two nodes (i, j) and (j, k), i.e. to the triplet (i, j, k):

$$P_{(i,j),k} = \frac{\tau_{(i,j),k}^{\alpha} \eta_{(i,j),k}^{\beta}}{\sum_{k \in allowed} \tau_{(i,j),k}^{\alpha} \eta_{(i,j),k}^{\beta}} \tag{8}$$

The ant stops searching more nodes when the termination criteria for the candidate solution have been met. This process is iterated for all the ants, whose number is n_{ants}, in the colony. After an entire colony has been released, i.e. has constructed candidate solutions, the pheromone is updated considering an evaporation rate $\rho \in [0,1]$ and how many ants pass through those paths:

$$\tau_{(i,j),k} \leftarrow (1-\rho)\tau_{(i,j),k} + \sum_{a=1}^{n_{ants}} \Delta\tau^{a}_{(i,j),k} \quad (9)$$

Where $\Delta\tau^{a}_{(i,j),k}$ is defined as:

$$\Delta\tau^{a}_{(i,j),k} = \begin{cases} \frac{1}{L_a} \; if\ the\ ant\ 'a'\ has\ gone\ through\ (i,j,k) \\ 0\ otherwise \end{cases} \quad (10)$$

corresponding to the new pheromone trail, depending upon the cost L of the tour.

The next population of ants repeats this process with more information than the previous one. The termination criteria for the graph exploration can be fixed by a maximum number of populations, by a maximum computational time, or a criterion related to the problem.

For the purpose of the present paper, the following variations to the standard ACO have been introduced:

- Backtracking: when an ant has selected a node that is not compatible with REQ-001, the last asteroid is removed from the tour. This is done to mitigate the risk for the ants of not building solutions compatible with REQ-001, due to the non-completeness of the score matrix (see section 3)
- by the ants that are non-feasible because of REQ-002 are kept in each iteration and considered in the pheromone update.
- Select next node: for the selection of the next node, only feasible nodes in the cleaned Search Space $\mathcal{G}c$ are considered at each step.
- Calculate Cost: the algorithm uses the Score Matrix SM to reduce computational cost.
- Pheromone parameter: The pheromone is initialised null in all the non-feasible entries. In each iteration the pheromone of the nodes contained in Avoid Tours is decreased.

The performances of the algorithms have been evaluated over several settings for three different catalogues (i.e. the ones with 49, 98 and 158 asteroids).

Performances of the BS algorithm are evaluated over a grid of settings for the BW for the catalogues considered. The BW varies from 0 to the one allowing the global optimum, as in section 4, to be included in the final set of solutions. Results obtained are shown in Figure 8 and Table 6. For all the three catalogues considered, it is possible to identify several levels of objective function values, ranging from about 7 km/s to approximately 8.4 km/s. A variation on the BW marks the transition from one level to another. This variation increases with the number of asteroids in the different catalogues, suggesting an underlying multi-multimodal structure of increasing complexity of the problem at hand. In other words, while the BS guarantees local optima solutions in reduced computational effort, the diversity of local minima of the asteroid tour problem is such that the efficiency of BS in escaping from one funnel reduces when larger catalogues are considered.

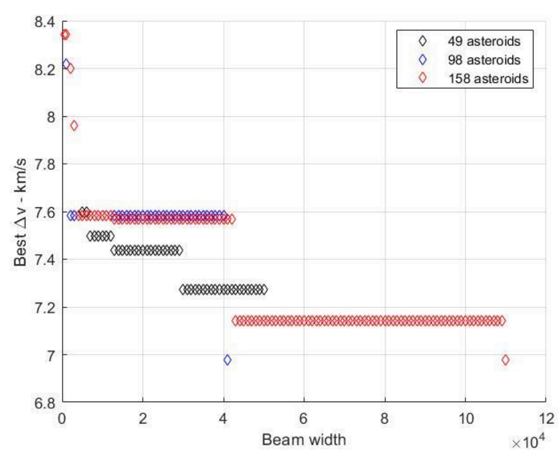

Figure 8. Best Δv solutions with respect to BW for different catalogues considered.

Table 6. BW for obtaining the global optimum sequence for different catalogues and number of solutions obtained.

Number of	BW for	Number of	Mean

asteroids	global optimum	solutions	cost [km/s]
49	30e3	3805	8.4458
98	41e3	2662	8.4111
158	110e3	7570	8.4355

While BS allows for very limited computational effort, it is known to be an uncomplete algorithm, i.e. it does not guarantee the globally optimal solution [33], [36], [37] since a too aggressive pruning (i.e., low BW) may discard good solutions at early stages of the search. The selection of the BW is thus crucial, as it represents the compromise between computational effort and quality of the obtained solution. However, for the problem at hand, it appears to be sufficient in obtaining wide range of solutions satisfying all the constraints of the CSP, without the need of more complex metaheuristic strategies (as ACO, see later).

Table 7. ACO variables values and definitions.

Variable	Definition	Value
α	Exponent weight to pheromone	1
β	Exponent weight to heuristic parameter	5
ρ	Pheromone evaporation rate	0.05
$n_{bt,max}$	Maximum number of backtrackings allowed	50

Regarding the ACO, the algorithm is run 30 independent times for each catalogue considered. All the cases are initialised with a Score Matrix pruned at 1 km/s per leg ($SM \leq 1 km/s$), and 5 km/s for the first node (Earth – Asteroid1), as for REQ-004. The parameter set-up is the same for all the runs and for all the catalogue explored and it is described in Table 7.

Table 8. Solutions from ACO for the catalogues explored.

Catalogue→	49 asteroids	98 asteroids	158 asteroids
Number of feasible solutions	79	22	1
Best tour cost	7.6438 km/s	6.9772 km/s	8.4149 km/s
Mean cost of solutions	8.2467 km/s	8.6403 km/s	11.2069 km/s

Table 8 shows results for the ACO with respect to the different catalogues explored. For the CSP problem at hand, BS provides many more solutions compliant with the problem constraints on similar computational effort, with respect to the ACO. In addition, because of the lack of smoothness of the search space, the ACO performance is highly dependent on the area where the ants start to search. If the neighbourhood is very bad (i.e. containing solutions with low cost or Δv), the ants tend to stay in the good area that can contain local minima, existing other areas where the local minima are lower. Mean values of solutions found by ACO are in general higher than the ones of BS, because ACO seems to explore less feasible regions.

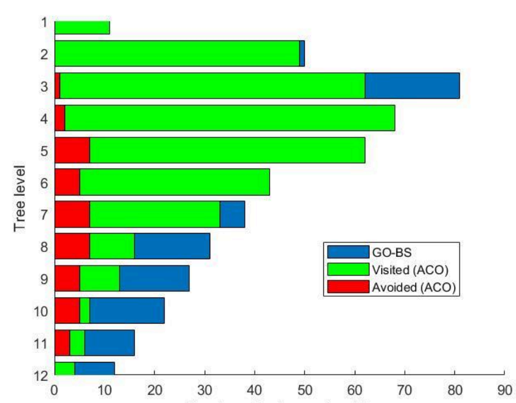

Figure 9. Number of unique asteroids visited and avoided by tree levels for ACO with SM ≤ 1 km/s for catalogue with 98 asteroids best case. GO-BS reports the BS exploration able to find the global optimum solution

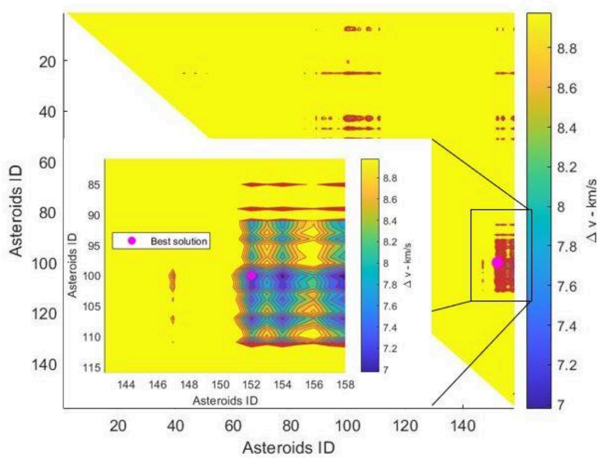

Figure 10. Δv for close-to-optimal solutions for catalogue with 98 asteroids.

Moreover, as a general consideration, it has found that ACO is an inconsistent algorithm for the problem at hand, meaning that it seems to explore very well the first layers of the search space, until the 5th and 6th level, while misses the global optimum solution when lower layers are explored. In fact, the GO is only found for the catalogue with 98 asteroids. This is shown in Figure 9 on which unique asteroids visited by the ants per tree level are plotted. On all the three cases considered, layers 9, 11 and 12 are very restricted on the diversity in the exploration of the ants. Most of the layers are vaguely explored, showing a lack of diversity in the construction of the solutions of the ants. This plot shows that even if layers 4, 5 and 6 seem to be more diverse, the rest of the layers are very restricted, because of the lack of smoothness of the search space. This is shown in Figure 10, showing the Δv distributions for sequences that differ from the optimal by two asteroids, reflecting the non-linearity and multi-modality of the problem at hand. In this case, the asteroid in 7th and 12th positions are removed. It can be appreciated that most of the search space is in fact dominated by non-feasible solutions, thus ACO struggles to build solution, as it cannot extract information from there areas. In the case of the 158-asteroids catalogue, early-stage layers are very diverse, while the diversity reduces as the layer increases from the 4th layer on. It is noticeable, however, that this reduction of diversity is more pronounced that for the lower MOID case of 0.04 AU. The search is even bigger in this case, with more sharp areas of good solutions that make it more difficult for the ants to find these areas, being able to only find one feasible solution in total.

As such, compared to the BS, ACO algorithm is a metaheuristic solver that is meant to find a feasible solution (local minimum) under some constraints. Being a heuristic solver, ACO does not guarantee the finding of the global minimum; it is in fact used when the analytical solver is too complex or takes too much time to solve. Niching strategies or hybridization with BS would increase the chance for ACO to deal with the multi-modality of the search space. In this sense, ACO is suited to deal with CSP, while providing less information about the multi-modality nature of the problem, as bias the elite solutions which results in the diversity loss.

Increasing the size of the Search Space seems to increase the difficulty for the ants to find feasible solutions. The number of asteroids to find are still 12 for each of the cases but the number of triplets increase rapidly with the expansion of the catalogue of asteroids. The Search Space sharpness is more noticeable when augmenting the catalogue as the ants find a total (summing all the execution times runs) of 84 feasible solutions for MOID 0.03 (49 asteroids), 24 for MOID 0.04 (98 asteroids) and 1 for MOID 0.05 (158 asteroids).

Analysing the bar plots by layers of the unique asteroids explored by the ants, it is clear that ACO does not explore the whole Search Space. It has to be considered that some nodes in each layer are explored by the ants but avoided with backtracking, as those nodes do not allow the construction of a 12-asteroid tour. Nevertheless, even considering this, the ants do not explore the whole space. In all of the three MOID cases, the most diverse layer is the 4th one, decreasing the diversity of the following layers from that on. The ants keep narrowing the solutions as they construct deeper in the tour, entering in a sharp area of the Search Space.

6. Conclusions

This work has transcribed the asteroid tour problem in two different, though related, formulations: a minimization formulation, aiming at finding the global optimum solution, and a formulation based on the Constraint Satisfaction Problem (CSP) paradigm. An algorithm has been presented for obtaining global optimum solutions even for large number of asteroids in the datasets. This is efficient and can be easily improved for handling even larger catalogues. However, the curse of dimensionality is only mitigated, and hybrid strategies are required for problems of increased complexity. This will be the focus of future research.

Moreover, Beam Search (BS) strategy and Ant Colony Optimization (ACO) with back-tracking have been tested over the defined CSP formulation for the asteroid tour problem. Deterministic tree-like strategies as the BS perform well in handling the multi-modality of the search space, as well as solving the CSP. On the other hand, ACO, while appears suited for the CSP, it needs to be improved by niching strategies or some form of hybridization with BS to handle the multi-modality of the search space. Moreover, diversity-preserving techniques for global optimization could be also in use to retain good yet diverse solutions, so that the search becomes biased not by optimality, but rather by diversity.

References

[1] O. Grasset *et al.*, "JUpiter ICy moons Explorer (JUICE): An ESA mission to orbit Ganymede and to characterise the Jupiter system," *Planet. Space Sci.*, vol. 78, pp. 1–21, 2013.

[2] D. Izzo, I. Getzner, D. Hennes, and L. F. Simões, "Evolving solutions to TSP variants for active space debris removal," in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 2015, pp. 1207–1214.

[3] N. E. Bowles *et al.*, "CASTAway: An asteroid main belt tour and survey," *Adv. Sp. Res.*, vol. 62, no. 8, pp. 1998–2025, 2018.

[4] J. P. S. Cuartielles, A. Gibbings, C. Snodgrass, S. Green, and N. Bowles, "Asteroid belt multiple fly-by options for m-class missions," in *Proceedings of the International Astronautical Congress, IAC*, 2016.

[5] K. Alemany and R. D. Braun, "Survey of global optimization methods for low-thrust, multiple asteroid tour missions," 2007.

[6] M. Schlueter, "Nonlinear mixed integer based optimization technique for space applications." University of Birmingham, 2012.

[7] M. Schlueter, S. O. Erb, M. Gerdts, S. Kemble, and J.-J. Rückmann, "MIDACO on MINLP space applications," *Adv. Sp. Res.*, vol. 51, no. 7, pp. 1116–1131, 2013.

[8] I. M. Ross and C. N. D'Souza, "Hybrid optimal control framework for mission planning," *J. Guid. Control. Dyn.*, vol. 28, no. 4, pp. 686–697, 2005.

[9] D. Izzo, "1st ACT global trajectory optimisation competition: Problem description and summary of the results," *Acta Astronaut.*, vol. 61, no. 9, pp. 731–734, 2007.

[10] S. Li, X. Huang, and B. Yang, "Review of optimization methodologies in global and China trajectory optimization competitions," 2018.

[11] C. A. Floudas, *Nonlinear and mixed-integer optimization: fundamentals and applications*. Oxford University Press, 1995.

[12] I. E. Grossmann, "Review of nonlinear mixed-integer and disjunctive programming techniques," *Optim. Eng.*, vol. 3, no. 3, pp. 227–252, 2002.

[13] A. H. Land and A. G. Doig, "An automatic method for solving discrete programming problems," in *50 Years of Integer Programming 1958-2008*, Springer, 2010, pp. 105–132.

[14] B.-S. Cheung, A. Langevin, and H. Delmaire, "Coupling genetic algorithm with a grid search method to solve mixed integer nonlinear programming problems," *Comput. Math. with Appl.*, vol. 34, no. 12, pp. 13–23, 1997.

[15] A. Munawar, M. Wahib, M. Munetomo, and K. Akama, "Advanced genetic algorithm to solve MINLP problems over GPU," in *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2011, pp. 318–325.

[16] L. Yiqing, Y. Xigang, and L. Yongjian, "An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints," *Comput. Chem. Eng.*, vol. 31, no. 3, pp. 153–162, 2007.

[17] M. Schlüter, J. A. Egea, and J. R. Banga, "Extended ant colony optimization for non-convex mixed integer nonlinear programming," *Comput. Oper. Res.*, vol. 36, no. 7, pp. 2217–2229, 2009.

[18] I. Grigoriev and M. Zapletin, "Choosing promising sequences of asteroids.," *Autom. Remote Control*, vol. 74, no. 8, 2013.

[19] D. Izzo, D. Hennes, L. F. Simões, and M. Märtens, "Designing complex interplanetary trajectories for the global trajectory optimization competitions," in *Space Engineering*, Springer, 2016, pp. 151–176.

[20] M. Di Carlo, M. Vasile, and J. Dunlop, "Low-thrust tour of the main belt asteroids," *Adv. Sp. Res.*, vol. 62, no. 8, pp. 2026–2045, 2018.

[21] A. E. Petropoulos *et al.*, "GTOC5: results from the Jet Propulsion Laboratory," *Acta Futur.*, vol. 8, pp. 21–27, 2014.

[22] L. F. Simões, D. Izzo, E. Haasdijk, and A. E. Eiben, "Multi-rendezvous spacecraft trajectory optimization with beam P-ACO," in *European Conference on Evolutionary Computation in Combinatorial Optimization*, 2017, pp. 141–156.

[23] Z. Fan, M. Huo, N. Qi, Y. Xu, and Z. Song, "Fast preliminary design of low-thrust trajectories for multi-asteroid exploration," *Aerosp. Sci. Technol.*, vol. 93, p. 105295, 2019.

[24] J. T. Olympio, "Optimal control problem for low-thrust multiple asteroid tour missions," *J. Guid. Control. Dyn.*, vol. 34, no. 6, pp. 1709–1720, 2011.

[25] K. Zhu, F. Jiang, J. Li, and H. Baoyin, "Trajectory optimization of multi-asteroids exploration with low thrust," *Trans. Jpn. Soc. Aeronaut. Space Sci.*, vol. 52, no. 175, pp. 47–54, 2009.

[26] A. S. Rivkin *et al.*, "The main-belt asteroid and NEO tour with imaging and spectroscopy (MANTIS)," in *2016 IEEE Aerospace Conference*, 2016, pp. 1–14.

[27] A. Bellome, J. P. Sanchez Cuartielles, S. Kemble, and L. Felicetti, "A multi-fidelity optimization process for complex multiple gravity assist trajectory design."

[28] R. Gallego, "Multi-gravity Assist Trajectories Preliminary Analsys with Tisserand Graph," Cranfield University, 2020.

[29] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[30] R. Bellman, "The theory of dynamic programming," *Bull. Am. Math. Soc.*, vol. 60, no. 6, pp. 503–515, 1954.

[31] V. Kumar, "Algorithms for constraint-satisfaction problems: A survey," *AI Mag.*, vol. 13, no. 1, p. 32, 1992.

[32] S. C. Shapiro, *Encyclopedia of artificial intelligence second edition*. John, 1992.

[33] C. M. Wilt, J. T. Thayer, and W. Ruml, "A comparison of greedy search algorithms," in *third annual symposium on combinatorial search*, 2010.

[34] P. S. Ow and T. E. Morton, "Filtered beam search in scheduling," *Int. J. Prod. Res.*, vol. 26, no. 1, pp. 35–62, 1988.

[35] B. W. Barbee, G. W. Davist, and S.-H. Diaz, "Spacecraft trajectory design for tours of multiple small bodies," *Adv. Astronaut. Sci.*, vol. 135, no. 3, pp. 2169–2188, 2009.

[36] B. W. Barbee, S. Alfano, E. Pinon, K. Gold, and D. Gaylor, "Design of spacecraft missions to remove multiple orbital debris objects," in *2011 Aerospace Conference*, 2011, pp. 1–14.

[37] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, 1997.