J Glob Optim (2009) 44:493–508
DOI 10.1007/s10898-008-9352-4

Genetic algorithms applied to the solution of hybrid optimal control problems in astrodynamics

Bradley J. Wall · **Bruce A. Conway**

Received: 14 April 2008 / Accepted: 31 August 2008 / Published online: 19 September 2008

Abstract Many space mission planning problems may be formulated as hybrid optimal control problems, i.e. problems that include both continuous-valued variables and categorical (binary) variables. There may be thousands to millions of possible solutions; a current practice is to pre-prune the categorical state space to limit the number of possible missions to a number that may be evaluated via total enumeration. Of course this risks pruning away the optimal solution. The method developed here avoids the need for pre-pruning by incorporating a new solution approach using nested genetic algorithms; an outer-loop genetic algorithm that optimizes the categorical variable sequence and an inner-loop genetic algorithm that can use either a shape-based approximation or a Lambert problem solver to quickly locate near-optimal solutions and return the cost to the outer-loop genetic algorithm. This solution technique is tested on three asteroid tour missions of increasing complexity and is shown to yield near-optimal, and possibly optimal, missions in many fewer evaluations than total enumeration would require.

Keywords Hybrid optimal control · Genetic algorithm · Spacecraft trajectory optimization · Global trajectory optimization competition (GTOC) · Bilevel programming problem (BLPP)

Nomenclature

β	Thrust pointing angle (out-of-plane)
θ	Angular position
ΔV	Impulsive change in velocity
a, b, c, d, e, f, g	Coefficients of the 6th degree inverse polynomial

B. J. Wall (✉)
Aerospace Engineering, Embry-Riddle Aeronautical University, Prescott, AZ, USA
e-mail: wallb@erau.edu

B. A. Conway
Aerospace Engineering, University of Illinois at Urbana–Champaign, Urbana, IL, USA
e-mail: bconway@uiuc.edu

c	Spacecraft propellant exhaust velocity
J	Performance index
m	Mass
r	Radial distance from the spacecraft to the sun
T_a	Thrust acceleration
t_f	Time of flight
ν_r	Radial velocity
ν_t	Tangential velocity

1 Introduction

Hybrid optimal control problems (HOCP) are problems that include both continuous-valued variables and categorical variables in the problem formulation. In orbital mechanics problems the categorical variables will typically specify the sequence of events that qualitatively describe the trajectory or mission. For example, for an interplanetary spacecraft trajectory, a mission could be described very simply by the following sequence of categorical variables: Earth departure impulse, Mercury arrival impulse. An equally valid and perhaps lower cost sequence might be: Earth departure impulse, Venus gravity assist, Mercury arrival impulse. A current practice in solving HOCP's is to use two nested loops: an outer-loop problem solver which handles the finite dynamics and finds a solution sequence in terms of the categorical variables, and an inner-loop problem solver that performs the optimization of the corresponding continuous-time dynamical system and obtains the required control law time-history. The goal of the outer-loop problem solver is to find the optimal sequence of categorical variables in many fewer evaluations than total enumeration would require. Since the outer-loop problem optimization space is discontinuous, traditional numerical methods will not work well. Here two methods are investigated as possible outer-loop problem solvers; a simple evolutionary algorithm, the genetic algorithm (GA), and a systematic method known as Branch and Bound (B&B). When using a GA the inner-loop problem solver must *quickly* locate the optimal (or near-optimal) trajectory and performance index corresponding to each categorical sequence, because the outer loop GA typically requires the evaluation of the cost of hundreds of possible sequences for each generation, and there may in turn be hundreds of generations needed before the GA converges to a solution. The B&B method also requires many solutions of the inner-loop problem, including solutions of "relaxed" versions of the actual problem. In both cases, obtaining large numbers of low-thrust spacecraft trajectories, for various terminal boundary conditions, is problematic since there is no analytical solution in the general case and since, given a sufficiently large search space of departure and arrival times, there may be a significant number of local optima.

Thus methods that are normally successful for the solution of optimal low-thrust trajectories, e.g. direct methods using collocation with nonlinear programming [1–4], are not useful for this problem because of the execution time required to solve the NLP problem and the need to provide them with a suitable initial guess of the solution (which alone can be time-consuming), and since if the inner-loop solver fails to converge to a solution for some sequence it is called upon to evaluate by the outer-loop solver, the solution process will come to a halt. In this research these difficulties are overcome by using a GA, incorporating a shape-based approximation method, as the inner-loop problem solver. This solver is shown to be capable of rapidly locating near-optimal low-thrust rendezvous trajectories for a large search space of departure and arrival dates and, as common to all GA's, with no a priori information about the solution [5].

2 Outer-loop problem solvers

Two methods are used here, and compared, as outer-loop problem solvers: Branch-and-Bound (B&B) [6] and GA [7]. B&B is an optimization method widely used in industrial applications such as process optimization and task scheduling. It has been applied recently for discrete optimization in the context of HOCP's [8].

Optimal control theory states that a problem with relaxed or fewer constraints has a smaller cost than the nominal problem. Hence, evaluating partial categorical variable sequences provides lower bounds that can be compared with the cost of a complete "incumbent" solution sequence. The incumbent solution is a feasible suboptimal sequence of full length that can ordinarily be easily found by intuition or experience. If the cost of any partial sequence is higher than that of the incumbent, no sequence proceeding from that partial sequence will have a smaller cost and thus does not require evaluation. Note that modeling of the problem is a very important matter to ensure that relaxations of the problem exist in order to compute the cost of partial sequences. An example of direct relevance to the asteroid tour missions used as examples in this work is the famous traveling salesman problem. If the salesman is to leave the origin, visit N cities and then return to the origin, it is clear that the cost of a tour of $N-1$ cities (a partial sequence) is a lower bound for the cost of any N city sequence including the original $N-1$ cities (in original order). Thus no partial sequence with a trip cost higher than a known (incumbent) feasible N-city tour can be the foundation of an optimal full sequence.

2.1 The branch and bound method

The B&B solution process can be described as proceeding down the branches of a tree, as shown in Fig. 1 for a system with a sequence of length three and with categorical variables of type a, b, or c. If the search finds a partial sequence with a higher cost than the incumbent, the tree can be pruned at that point as no solutions found below that point can have a lower cost. If the search arrives at a final event node and the complete sequence is found to be better than the incumbent, the newly found sequence becomes the new incumbent. An important feature of the method is that it is deterministic, it finds the best solution while reducing the search space methodically. The performance of the method is affected primarily by the selection of the incumbent solution sequence. A poor incumbent will do little to eliminate full sequences from evaluation in a significant way causing the search to be similar to total enumeration.

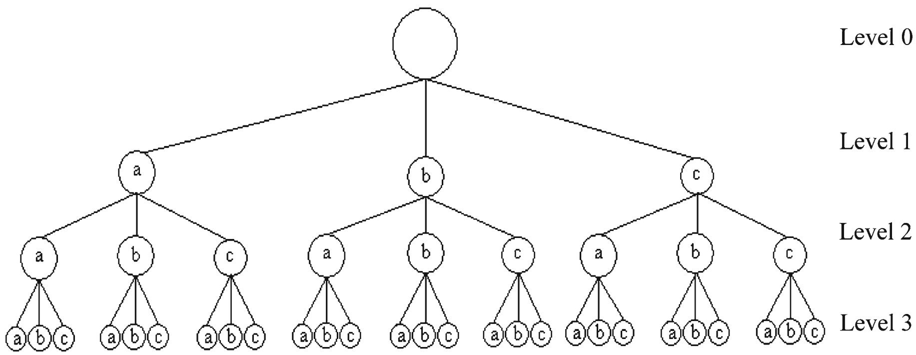


Fig. 1 Cartoon illustrating the Branch & Bound algorithm

Given that partial sequences must be evaluated, it is possible that the B&B method will evaluate more sequences altogether than a total enumeration of the full-length sequences would require. However, in fairness, evaluation of the cost of partial sequences is usually faster than that for total sequences.

2.2 The genetic algorithm

The second outer-loop problem solver that is used in this work is the GA. A genetic algorithm is a robust optimization driver based on the rules of evolution, i.e. selection, reproduction, and mutation. In the optimizer each parameter describing the solution is coded into a numeric string with lower and upper bounds. Each set of parameters, i.e. each *individual*, is then decoded (into a set of variables that define the trajectory or from which it may be determined) and then the fitness function or cost of the trajectory may be evaluated. After all the individuals in one generation, i.e. the total population, are evaluated, evolutionary rules of selection, reproduction, and mutation are applied to the corresponding individuals, based upon their fitness values, to create a new, and often improved, set of individuals. In short, a GA attempts to "evolve" to an optimal solution [7].

There are two different ways of encoding a GA parameter that are explored in this research; real-valued and binary. Of course, each method of encoding has advantages and disadvantages. A real-valued GA (RGA) has the advantage that it is easier to use. The user must only specify the lower and upper bounds on each parameter and the GA is able to pick any number, within machine precision, between those bounds. This gives the RGA the potential for high accuracy, however it is not suitable for an integer programming problem. A binary GA (BGA) has the advantage that each parameter is encoded into a binary string using a finite number of possibilities (usually a power of 2) between the user supplied lower and upper bounds making it suitable for integer programming problems. A disadvantage is that parameters that are real numbers must be encoded to a fixed precision resulting in lower accuracy. The encoding and decoding algorithms required for a BGA, but not a RGA, also add computational time. The BGA used in the first example (the "Asteroid Interception Problem") is based on the simple GA FORTRAN code by Carroll [9]. All of the other BGA's and RGA's used in this research are from the Genetic Algorithm and Direct Search Toolbox available in MATLAB. While the Genetic Algorithm and Direct Search Toolbox has many options to choose from for the evolutionary operators, in this research tournament selection, uniform (scattered) crossover, adaptive feasible mutation, and elitism are used.

When using a GA as the outer-loop problem solver one must devise a way of dealing with possible infeasible sequences. For example, in the famous traveling salesman problem, no city can be visited more than once therefore any city sequence that includes the same city twice is an infeasible sequence. There are three possible options for dealing with infeasible sequences:

1. The initial random population generator will run until the initial population sequences are all feasible and then an advanced crossover and mutation operator can be used such that no infeasible sequences arise during optimization.
2. The infeasible sequences can be removed from the population as they arise and then the selection, crossover, and mutation operations may be continued until the population is replenished with feasible sequences or the GA may be continued with less than a full population.
3. The infeasible sequences can be given an arbitrarily high fitness value to naturally remove them from the population.

According to Gage et al. [10] GA's are well suited for search spaces where no more than 90% of the population are infeasible. Since this is the case for the problems of interest here the third option is chosen because of its simplicity of implementation.

One advantage of using a GA as the outer-loop problem solver is that it does not require the partial sequence (relaxed problem) evaluations that the B&B method requires, making it more likely to need to evaluate the cost of many fewer sequences than total enumeration would require. However, the GA is a stochastic method, meaning that it may not yield the same solution every time whereas the B&B method is a deterministic approach and will always yield the global optimal categorical variable sequence (assuming that the inner-loop problem solver is locating the global optimal solution for each sequence). A second advantage of the GA is that the initial (zeroth generation) population is generated randomly, i.e. no a priori information about the optimal trajectory is required.

Note that there are no analytical conditions of optimality used in the GA method. In principle, to demonstrate optimality of a solution obtained using GA, the solution can be given to a calculus-of-variations based optimizer. Then one can see if this optimizer converges to essentially the same solution.

In each case it is required to have an inner-loop problem solver that provides the optimal cost for each sequence generated during the operation of the B&B or GA outer-loop problem solver and that returns that value to the outer-loop problem solver. This can be problematic as the outer-loop problem solver will halt operation if the optimization of the inner-loop problem results in an error for any reason, the most likely reason of course being that the inner-loop solver has failed to converge to a solution for some case.

3 Inner-loop problem solvers

There are many choices for the inner-loop problem solver. Direct methods are preferred over indirect methods because they are generally more robust [11]. Direct transcription methods do not require the classical Euler-Lagrange equations and optimality conditions thereby making this method easy to use for a wide variety of problems. As the problem structure changes, direct transcription methods are more easily modified than indirect methods that require a new analytical derivation of the Euler-Lagrange equations; something that is impractical for automatic use. When using direct transcription with nonlinear programming, the original continuous control problem must be discretized, i.e. the state and control time history must be represented by discrete parameters. There are several good direct transcription methods that have been successfully applied to orbit transfer problems [1–4,12,13]. The method used in this work is direct transcription with a fourth order Runge-Kutta (DTRK) integration rule and parallel-shooting scheme developed by Enright and Conway [2].

Another inner-loop problem solver used in this research is based on the GA. The authors and other researchers have had some experience applying the GA method to problems in astrodynamics [14–17] and this experience suggested that a GA might work well as a inner-loop problem solver. Of course the GA is not capable of the high accuracy that direct and indirect methods may achieve, so determining if this makes them unsuitable as an inner-loop problem solver was one of the goals of this work.

4 Asteroid interception problem

In this orbital mechanics analogue of the traveling salesman problem (TSP), a spacecraft departs from Earth orbit about the Sun and must visit three asteroids out of a population of eight. As in the TSP, the vehicle must visit each target only once. The maneuvers are impulsive; there must be one at the initial time to leave Earth orbit and then one at each of the first two asteroids to intercept the next asteroid. (The orbit of the spacecraft after the third interception is unconstrained; i.e. it is not important to the problem.) The spacecraft position and velocity are defined in a heliocentric polar coordinate system. For simplicity, the asteroid orbits are considered to be circular, and canonical units are used. The goal is to minimize the amount of fuel, that currently costs approximately \$10,000 per kg into low Earth orbit, required to perform the mission,

$$\min J = \Delta V_1 + \Delta V_2 + \Delta V_3 \tag{1}$$

The dynamics of the system are given by,

$$\begin{aligned} \dot{r} &= v_r \\ \dot{\theta} &= \frac{v_t}{r} \\ \dot{v}_r &= \frac{v_t^2}{r} - \frac{1}{r^2} \\ \dot{v}_t &= -\frac{v_r v_t}{r} \end{aligned} \tag{2}$$

where r is measured in astronomical units (AU), and θ is measured in radians with respect to a space-fixed reference line directed to the first point in Aries.

At the initial time, the polar coordinates of the spacecraft are $r = 1$ and $\theta = 0$; since it begins in a circular orbit $v_r = 0$, and $v_t = 1$. The initial locations of the eight asteroids are given in Table 1.

This problem is more complex than the traveling salesman problem for several reasons:

1. The equations of motion are nonlinear.
2. The asteroid targets are moving while of course the cities are stationary.
3. There is no symmetry, therefore total enumeration would require evaluating 336 feasible sequences.

A solution of the HOCP is found using a B&B outer-loop problem solver combined with a GA inner-loop problem solver. The only required GA parameters for this problem are the three transfer times. Given the transfer times, the time of arrival at each asteroid can be calculated and hence the position of any asteroid at arrival can be determined. With this information, a Lambert's solution yielding the impulse required to arrive at the asteroid in

Table 1 Initial positions of the asteroids

Asteroid	1	2	3	4	5	6	7	8
r	1.450	1.699	2.001	1.490	1.650	1.730	1.960	1.700
θ	0.377	0.564	0.761	0.430	0.512	0.617	0.708	0.813

the given transfer time can be found using the universal Lambert solver found in Battin [18]. Because the solution is exact, no equations of motion are required.

Two versions of relaxed problems are required by the B&B method. The first is defined as the minimum-fuel intercept of only the first asteroid since any two-asteroid sequence that includes the first asteroid must have a greater or equal cost. The second relaxed problem is defined as the minimum-fuel intercept of the first two asteroids. Each GA is run for 500 generations. After eight single-asteroid sequence evaluations, 56 two-asteroid sequence evaluations, and 132 three-asteroid sequence evaluations an optimal sequence of 8-4-5 and fitness value of 0.1688 AU/TU were found.

Conway et al. [17] solved the same problem using a GA as the outer-loop problem solver and a direct method, direct transcription with Runga-Kutta parallel shooting (DTRK), as the inner-loop problem solver. The equations of motion (Eq. 5) are implicitly integrated in this solutions, i.e. the Lambert analytical solution is not used. This solution should be more accurate than the solution obtained here because the GA inner-loop problem solver is not capable of the same accuracy as the DTRK method. However, this more accurate method finds the same optimal sequence, 8-4-5, and a cost of 0.1672 AU/TU. The discrepancy between the DTRK solution of 0.1672 AU/TU and the GA solution of 0.1688 AU/TU is in fact quite small and can be attributed to the two decimal place accuracy limitation in the GA transfer time parameters. Ostensibly then, the DTRK is better but each asteroid sequence evaluation using the GA inner-loop problem solver is much faster and does not require an initial guess, which can be problematic to generate. This needs to be considered when comparing the two inner-loop problem solvers. The solution using the DTRK inner-loop problem solver needed only six generations of the GA outer-loop problem solver and 72 total asteroid sequence evaluations while 196 total asteroid sequence evaluations were needed when the GA inner-loop problem solver was used with the B&B outer-loop problem solver. That is, the GA outer-loop problem solver is capable of locating the optimal asteroid sequence in fewer total asteroid sequence evaluations than the B&B outer-loop problem solver.

A comparison of the trajectories and transfer times for the GA solution and the more exact DTRK solution for the sequence 8-4-5 are given in Fig. 2 and Table 2. Note that the

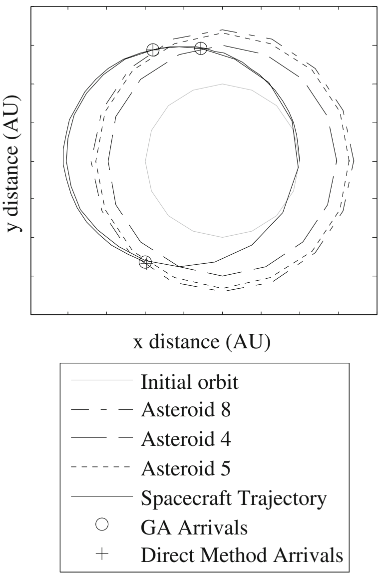


Fig. 2 Comparison of trajectories for optimal sequence 8-4-5

Table 2 Comparison of transfer times for optimal sequence 8-4-5

	DTRK	GA
$t1$ (TU)	2.91143499	2.92
$t2$ (TU)	13.8469257	13.85
$t3$ (TU)	20.8820705	20.85

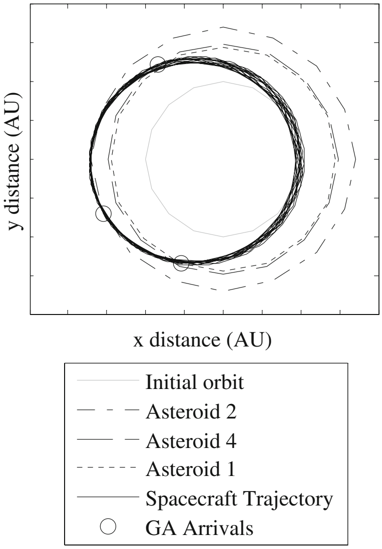


Fig. 3 Near-optimal multi-revolution trajectory of sequence 2-4-1

spacecraft trajectory generated using the GA and that generated by the direct method are so close that they would be indiscernible in this figure and thus only one trajectory is shown however, both sets of arrival points are shown.

This analysis has assumed that each leg of the trajectory requires less than one revolution. If this assumption is removed by including a multiple revolution Lambert solver a new optimal solution is obtained. Each GA is run for 300 generations. After eight single-asteroid, 42 two-asteroid, and 98 three-asteroid sequence evaluations an optimal sequence 2-4-1 and fitness value of 0.1416 AU/TU were found. As expected, the cost of the optimal multiple-revolution trajectory is less than that of the optimal less-than-one revolution trajectory. The multiple-revolution trajectory is shown in Fig. 3.

5 Low-thrust asteroid rendezvous problem

This problem is similar to the asteroid interception problem in that a spacecraft departs from Earth orbit about the Sun and must visit three asteroids out of a population of eight without visiting the same asteroid more than once. Here however, the orbital transfers are accomplished via a low-thrust engine and after rendezvous the spacecraft must remain with each asteroid for at least 90 days. This is qualitatively similar to the Global trajectory optimization competition 2 (GTOC2) problem (http://www.esa.int/gsp/ACT/mad/op/GTOC/index.htm), where the spacecraft must rendezvous for at least 90 days with four different asteroids, one from each of four families of asteroids. For simplicity, the asteroid orbits are considered to be the circular orbits with initial conditions used in the previous example (Table 1).

The goal is to minimize the amount of fuel required to perform the mission, i.e.

$$J = -m\left(t_f\right) \tag{3}$$

The dynamics of the system are given by,

$$\begin{aligned} \dot{r} &= v_r \\ \dot{\theta} &= \frac{v_t}{r} \\ \dot{v}_r &= \frac{v_t^2}{r} - \frac{1}{r^2} + T_a \cdot \sin(\beta) \\ \dot{v}_t &= -\frac{v_r v_t}{r} + T_a \cdot \cos(\beta) \\ \dot{m} &= -\frac{T_a \cdot m}{c} \end{aligned} \tag{4}$$

where T_a is the thrust acceleration of the low-thrust spacecraft, β is the thrust pointing angle, m is the mass, and c is the low-thrust exhaust velocity. The specific impulse chosen for the low-thrust engine is 3,875 s corresponding to an exhaust velocity c of 1.317 DU/TU. Specific impulse is a measure of the efficiency of a rocket motor. It is defined as the thrust produced divided by the weight flow rate of propellant. The initial mass of the spacecraft is 1,500 kg.

This problem is more complex than the asteroid interception problem for several reasons:

1. The equations of motion include the continuous controls, the thrust acceleration T_a and thrust pointing angle β.
2. When using low-thrust there is no longer an analytical solution (Lambert's solution) to the equations of motion.
3. Rendezvous, rather than interception, is required.

This problem is solved using two strategies in order to compare the efficiency of the two available outer-loop problem solvers, i.e. B&B and GA. The inner-loop problem solver must be able to calculate a low-thrust rendezvous trajectory for *any* combination of departure and arrival dates, i.e. to yield the low-thrust equivalent to the solution of Lambert's problem. For reasons described previously, i.e. robustness and speed, a shape-based approximation method using a parameterization of the trajectory based on a sixth degree inverse polynomial,

$$r = \frac{1}{a + b\theta + c\theta^2 + d\theta^3 + e\theta^4 + f\theta^5 + g\theta^6} \tag{5}$$

is used to model the trajectory [19]. The coefficients of the polynomial are chosen such that all the boundary conditions and equations of motion are satisfied for a specified transfer time. The thrust acceleration time history, allowing satisfaction of the system equations, is determined a posteriori, as described in [19].

A GA is used as the inner-loop problem solver. The shape-based method requires only the transfer time, where $1\,\mathrm{TU} \le t_f \le 40\,\mathrm{TU}$, and the number of revolutions N_{rev}, where $0 \le N_{\mathrm{rev}} \le 4$, for each leg of the three asteroid tour, resulting in a total of 6 GA parameters. Given a departure date and arrival date, the position and velocity of the departure body and target asteroid can be calculated. With this information the shape-based method is used to calculate low-thrust rendezvous trajectories, i.e. to find the thrust acceleration T_a and thrust pointing angle β time histories to satisfy the boundary conditions. Recall that the shape-based algorithm yields a feasible, not optimal, solution, therefore once a solution is found it must be optimized. There are several choices of when to perform this optimization:

1. Optimize every shape-based trajectory calculated.
2. Optimize the best shape-based trajectory found at the conclusion of each GA inner-loop evaluation.
3. Optimize the best shape-based trajectory or a selection of the best trajectories found at the conclusion of the outer-loop optimization.

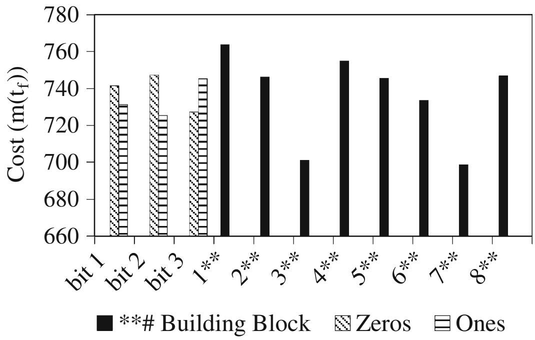
It is chosen here to optimize a collection of the best shape-based trajectories at the conclusion of the outer-loop optimization for two reasons. The first reason is that the computation time required for the optimization of this HOCP can be several hours. This obviously eliminates the ability to perform the optimization on every shape-based trajectory calculated. If only a fraction of the 336 total feasible sequences for this problem are evaluated, performing the optimization on each GA inner-loop shape-based solution could require additional days or weeks of computation time. The second reason is that it is likely that the ranking of outer-loop sequences will not change significantly after the optimization process, i.e. “good” trajectories will stay “good” and “poor” trajectories will stay “poor.” Recall that the shape-based solution satisfies the equations of motion so no numerical integration, which would be computationally expensive, is required.

The first solution of this HOCP is found using a B&B outer-loop problem solver combined with a GA inner-loop problem solver, an algorithm subsequently referred to as B&B + GA. Two versions of relaxed problems are required by the B&B method. The first is defined as the minimum-fuel rendezvous with only the first asteroid since any two-asteroid sequence that includes the first asteroid must have a greater or equal cost. The second relaxed problem is defined as the minimum-fuel rendezvous with the first and second asteroids. Each GA inner-loop is run for 30 generations. After eight single-asteroid sequence evaluations, 35 two-asteroid sequence evaluations, and 30 three-asteroid sequence evaluations an optimal sequence of 1-2-8 and fitness (final mass) of 1,253 kg were found. Recall that total enumeration would require 336 inner-loop evaluations however the B&B + GA method located the optimal asteroid sequence in only 73 inner-loop evaluations.

A GA outer-loop problem solver is also used to solve this HOCP with a GA inner-loop problem solver, an algorithm subsequently referred to as GA + GA. The GA outer-loop problem solver used a population of 50 and ran for 30 generations, requiring the GA inner-loop to evaluate 89 unique feasible sequences. The GA inner-loop problem solver uses a population size of 250 and is run for 30 generations for each sequence evaluation. An optimal sequence of 1-2-8, in fact the same sequence, and fitness of 1,257 kg were found. The difference of 4 kg, 1,253 kg vs. 1,257 kg, between the fitness values found using B&B + GA and GA + GA is attributed to the fact that the GA is a stochastic method and will likely never yield precisely the same solution twice. That is, the real-valued GA (RGA) being used here is capable of using parameters to machine precision. Because the GA is a stochastic method, i.e. it includes a random process (the mutation operator), it is highly probable that the GA will not converge to precisely the same (meaning to machine precision) string of numbers that represent the GA parameters.

To test the validity of the results obtained with the GA inner-loop problem solver utilizing the shape-based method, select solutions are used as initial guesses for a very accurate direct method, i.e. a DTRK method implemented with the MATLAB toolbox TOMLAB [20]. To determine what sequences should be evaluated, a total enumeration of all 336 sequences was completed and the solution-space analyzed.

Fig. 4 Intra-building block difficulty of the outer-loop problem for the low-thrust asteroid rendezvous problem



From the analysis of the categorical sequence versus cost solution-space the optimization difficulties of the outer-loop problem are made apparent. In terms of genetic algorithm difficulty, there exist three distinct types of difficult problems; they are intra-building block, inter-building block, and extra-building block [21]. Here a building block is another word for genetic algorithm optimization parameter. Intra-building block difficulty refers to problems where the costs of similar bit strings, known as schema, are seemingly random. For example, Fig. 4 shows the average cost of six different schemas; for bit strings where the first bit is a zero or a one, the second bit is a zero or a one, and the third bit is a zero or a one.

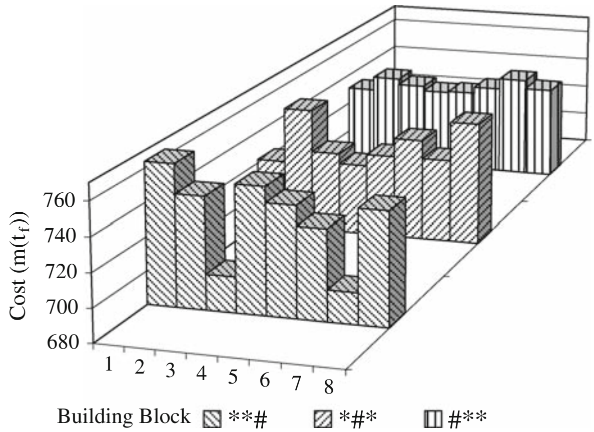
This illustrates that bit strings that have a zero as the first bit are on average better than bit strings that have a one as the first bit. Likewise, bit strings with a zero as the second bit are on average better than bit strings that have a one as the second bit and bit strings with a one as the third bit are on average better than bit strings that have a zero as the third bit. This favorable bit sting, i.e. 001, represents asteroid 2 thus, one would suspect that the building block schema associated with asteroid 2 being the first asteroid visited, i.e. 2**, would be on average better than any other first-asteroid schema however, this is not the case. In fact, building block schema 1** is on average better than any other first-asteroid schema with schema 2** only fourth best, being less than 4** and slightly less than 8**. This deception is known as intra-building block difficulty, i.e. the average cost of schema are seemingly random.

The second type of problem difficulty, inter-building block difficulty, refers to problems where some building blocks, i.e. GA parameters, are worth more to the solution of the problem than other building blocks. This can be seen in Fig. 5 which shows the average costs of each one-asteroid building block schema.

Since the variation of the average costs for the first-asteroid building block schemas, i.e. #**, is greater than the variation of the average costs for the second-asteroid building block schemas, i.e. *#*, which is greater than the variation of the average costs for the third-asteroid building block schema, i.e. **#, it can be deduced that the first building block, that associated with the first asteroid to be visited, is worth more to the solution than the second building block which is likewise worth more to the solution than the third building block. Thus, this outer-loop problem also has inter-building block difficulty.

The third type of problem difficulty, extra-building block difficulty, refers to problems where the cost of an individual is not fixed. For example, if a GA outer-loop problem solver with a GA inner-loop problem solver (GA + GA) HOCP solution method is used, unless the cost of each outer-loop sequence is saved and used subsequently for that sequence, the GA outer-loop problem also has extra-building block difficulty. This is because the GA is a stochastic method and will likely never result in the same cost twice for one given categorical sequence.

Fig. 5 Inter-building block difficulty of the outer-loop problem for the low-thrust asteroid rendezvous problem



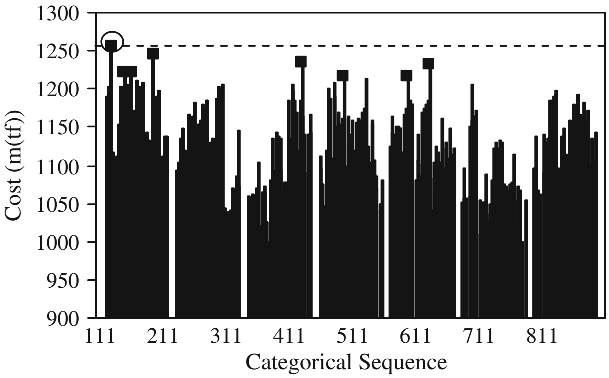


Fig. 6 The categorical sequence versus cost solution-space for the low-thrust asteroid rendezvous problem

The discontinuity and multi-modality of the categorical sequence versus cost solution-space, shown in Fig. 6, makes the outer-loop problem impossible for traditional optimizers. Although it has been shown that the outer-loop problem may have all three types of GA problem difficulty, the GA is still capable of solving such a problem [21]. Also shown in Fig. 6, and identified by the black boxes, are the eight best solutions obtained by the inner-loop GA using the shape-based approximation method.

These are used as initial guesses in a subsequent optimization using TOMLAB and the sparse nonlinear programming (NLP) problem solver SNOPT [22]. The DTRK method [2] is used as the direct transcription method. Twelve segments are used for each leg of the three asteroid tour, and two control variables, the thrust acceleration magnitude T_a and the thrust pointing angle β, are required. Experience has shown that an initial guess that satisfies the equations of motion and the boundary conditions is more likely to yield convergence of the NLP solver, therefore the shape-based method, which has these features, is an especially good initial guess. Since the thrust acceleration magnitude is not constant in the shape-based approximation method, it is defined as a NLP control variable. The thrust pointing angle, assumed to be along or against the velocity vector when using the shape-based method, is also defined as a NLP control variable. An upper-bound of $0.02\,\mathrm{DU/TU^2}$, corresponding to $1.2 \cdot 10^{-5}$ g, is enforced on the thrust acceleration magnitude.

Table 3 Low-thrust asteroid rendezvous comparison between the GA cost, using the shape-based method, and the optimal solution for the eight best shape-based solutions

Sequence	GA cost (m (t_f))	TOMLAB cost (m (t_f))	Optimization time (h)
1-2-8	1256.5	1252.9	6.53
1-8-2	1244.6	1252.0	2.10
4-2-8	1235.2	1252.3	4.74
6-2-8	1231.6	1232.1	3.84
1-5-4	1221.2	1230.4	9.09
1-4-5	1220.9	1252.8	5.54
4-8-2	1217.0	1249.5	11.14
5-8-2	1217.0	1237.1	6.50

The results of this optimization are shown in Table 3. There is a very good correspondence between the solutions using the shape-based method and the more accurate solutions obtained using the DTRK method. Note that the best GA solution remains the best solution after optimization.

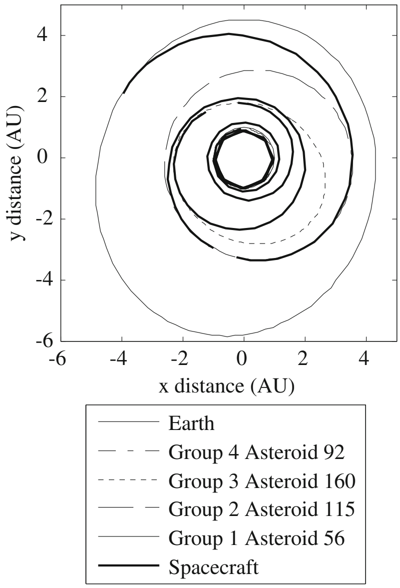
Thus, this example problem has shown that a GA outer-loop problem solver offers competitive computational results to that of the B&B solver presented for problems of this scale. Also, the GA inner-loop solver using the shape-based approximation [19] provides an acceptable approximation to the true optimal solution of each outer-loop sequence such that only a select few sequences require high-fidelity (costly) evaluation.

6 Solution of an ambitious asteroid rendezvous problem

To test the scalability of the GA + GA method, a more ambitious version of the low-thrust multiple asteroid rendezvous problem is solved. In this version of the problem a low-thrust spacecraft begins at Earth and must rendezvous with one asteroid from each of four groups of asteroids, each group containing many asteroids. The four groups of asteroids chosen are described in the GTOC2 problem statement (http://www.esa.int/gsp/ACT/mad/op/GTOC/index.htm). The number of possible sequences for visits to four groups (without repetition) is 24. For this example there are assumed to be 64 group 1 asteroids 128 group 2 asteroids, 256 group 3 asteroids, and 256 group 4 asteroids. These numbers are chosen because they are all powers of 2; they are more easily encoded into the binary format of the GA outer-loop problem solver and do not generate infeasible sequences. However, 25% of the outer-loop categorical state space is still infeasible due to there being 24 possible ways to visit the asteroid groups which must be encoded into a 5 bit (32 possibilities) binary string. The number of total feasible asteroid sequences is now 12.9 billion making total enumeration of the solution space impossible.

The problem is solved using a GA outer-loop problem solver with a GA inner-loop problem solver implementing the shape-based approximation method. The GA outer-loop problem solver is given a population of 50 and run for 50 generations. The GA inner-loop problem solver has a population of 250 and is run for 30 generations. On a 3.2 GHz computer, each inner-loop evaluation of a 4-asteroid tour takes approximately 12.5 min. For this problem, the GA inner-loop problem solver is allowed to evaluate any asteroid sequence it is given unless the asteroid sequence is infeasible or was the last asteroid sequence evaluated. This allows the same asteroid sequence to be evaluated more than once by the GA inner-loop problem solver making this outer-loop problem extra-building block difficult.

Fig. 7 Optimal trajectory for asteroid sequence 4.92-3.160-2.115-1.56



After 1,364 inner-loop problem evaluations the GA outer-loop problem solver identified two sequences with similar final masses, 967.8 and 967.7 kg. The total execution time required was 276 h on a 3.2 GHz processor. These sequences are (group number-asteroid number) 4.92-3.160-2.121-1.56 and 4.92-3.160-2.115-1.56 respectively. To determine which asteroid sequence is actually the best, these approximate optimal trajectories are used as initial guesses for a more-accurate direct method solver, as was done for the three asteroid rendezvous problem described in the previous section. The optimal final masses for the asteroid sequences are 991.3 and 992.8 kg respectively.

It is observed again that the mission final mass obtained using the shape-based method for low-thrust trajectories is a very good approximation of the "true" value obtained with the NLP-based direct solver. The execution time for the direct solver was 9 h on a 1.6 GHz processor, which is substantially longer than the 12.5 min required by the shape-based method. The optimal asteroid sequence trajectory is shown in Fig. 7.

Although it may be impossible to prove the global optimality of the asteroid sequence associated with the final mass of 992.8 kg, the value is consistent with "good" solutions obtained for the much smaller version of the asteroid rendezvous problem and has the same asteroid group sequence, 4-3-2-1, as the optimal solution found for the GTOC2 problem (http://www.esa.int/gsp/ACT/mad/op/GTOC/index.htm). Thus, the GA + GA method is capable of finding a near-optimal, if perhaps not optimal, categorical sequence for a large scale HOCP with many orders of magnitude fewer evaluations than total enumeration would require.

7 Conclusions

The hybrid optimal control problem solution algorithm developed here, in which GA's are employed for both the inner-loop and outer-loop problem solvers, has successfully and efficiently solved a number of mission planning problems in astrodynamics. The execution time for these very large problems is made feasible by using a shape-based method for the description of the low-thrust trajectory. The savings in computation time by using the shape-based method rather than a direct method, for example collocation with NLP (DCNLP) or direct transcription with RK parallel shooting (DTRK), for the inner-loop problem solver, is immense. For the final example problem, each four-asteroid sequence is evaluated in approximately 12 min using the shape-based method while the same sequence requires an average of over 7 h using the DTRK solver. It is suggested, from the results of this work, that for a problem such as this involving low-thrust arcs, the optimal solution strategy is the use of a GA outer-loop solver combined with a GA inner-loop problem solver that uses the shape-based approximation of the low-thrust trajectory. While the B&B outer-loop solver can be useful in some problems and has certain benefits compared to using a GA, it is virtually certain to require many more evaluations by the inner-loop solver than the GA. The advantage of the GA in this regard becomes more pronounced as the problem size increases because of the increase in the number of partial sequence evaluations required by the (classical) B&B outer-loop problem solver. For example, the B&B tree for the GTOC2 problem would include 910 one-asteroid sequences, 580,000 two-asteroid sequences, and 230 million three-asteroid sequences, in addition to the 41 billion four-asteroid, full-length, sequences. Recall that all of the one-asteroid sequences must be evaluated if B&B is used and that if the cost of the one-asteroid sequences does not exceed the incumbent four-asteroid sequence cost, which is likely for the GTOC2 problem, then a majority of the 580,000 two-asteroid sequences must also be evaluated.

More sophisticated B&B methods have been developed by Nemhauser and Wolsey [23] and Floudas [24] to significantly reduce the number of total evaluations however, these methods still do not scale as well to large scale problems as the GA + GA method presented. Also, some pre-pruning could be done without harm, eliminating some possibilities that are obviously bad, e.g. sequences that go out to an outer asteroid first and then return to an inner asteroid. This simple idea would eliminate half of the 41 billion sequences of the GTOC2 problem however; pre-pruning should always be done with great caution.

The GA + GA method for the solution of HOCP's is thus an improvement on current optimization technologies such as branch and bound. The GA outer-loop problem solver paired with the GA inner-loop problem solver implementing the shape-based method enables the solution of large HOCP's such as the GTOC problems without the need for pre-pruning the possible sequences, which allows the possibility that the true optimal sequence may be mistakenly removed from consideration.

References

1. Hargraves, C.R., Paris, S.W.: Direct trajectory optimization using nonlinear programming and collocation. J. Guid. Control Dyn. **10**, 338–342 (1987)
2. Enright, P.J., Conway, B.A.: Discrete approximations to optimal trajectories using direct transcription and nonlinear programming. J. Guid. Control Dyn. **15**(4), 994–1002 (1992). doi:10.2514/3.20934
3. Herman, A.L., Conway, B.A.: Direct optimization using collocation based on high-order Gauss-Lobatto Quadrature rules. J. Guid. Control Dyn. **19**(3), 592–599 (1996). doi:10.2514/3.21662
4. Ross, I., Fahroo, F.: Pseudospectral knotting methods for solving optimal control problems. J. Guid. Control Dyn. **27**(3), 397–405 (2004). doi:10.2514/1.3426
5. Wall, B.: Technology for the Solution of Hybrid Optimal Control Problems in Astronautics. Ph.D. Dissertation, University of Illinois, Urbana-Champaign (2007)
6. Little, J., Murty, K., Sweeney, D., Karel, C.: An algorithm for the traveling salesman problem. Oper. Res. **21**, 972–989 (1963)
7. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company, Reading (1989)

8. von Stryk, O., Glocker, M.: Decomposition of mixed-integer optimal control problems using branch and bound and sparse direct collocation. In: The 4th International Conference on Automation of Mixed Processes: Hybrid Dynamic Systems, Dortmund (2000)
9. Carroll, D.: Program gafortran version 1.7a. http://cuaerospace.com/carroll/ga.html (2004)
10. Gage, P., Braun, R., Kroo, I.: Interplanetary Trajectory Optimization Using a Genetic Algorithm. Paper AIAA 94-3773
11. Kraft, D.: On converting optimal control problems into nonlinear programming problems. In: Computational Mathematical Programming. NATO ASI Series, vol. F15. Springer-Verlag (1985)
12. Herman, A.L., Conway, B.A.: Direct solutions of optimal orbit transfers using collocation based on Jacobi polynomials. In: Advances in the Astronautical Sciences, vol. 87. Paper AAS 94-126 (1994)
13. Enright, P., Conway, B.: Optimal finite-thrust spacecraft trajectories using colocation and nonlinear programming. J. Guid. Control Dyn. **14**(5), 981–985 (1991). doi:10.2514/3.20739
14. Hartmann, J., Coverstone-Carroll, V., Williams, S.: Optimal interplanetary spacecraft trajectories via a pareto genetic algorithm. In: Advances in the Astronautical Sciences. Paper AAS 98-202 (1998)
15. Conway, B., Chilan, C., Wall, B.: Evolutionary principles applied to mission planning problems. Celestial Mech. **97**, 73–86 (2007). doi:10.1007/s10569-006-9052-7
16. Wall, B., Conway, B.: Developing a systematic approach to the use of genetic algorithms for the solution of optimal spacecraft trajectory problems. In: Advances in the Astronautical Sciences. Paper AAS 07-161 (2007)
17. Rauwolf, G., Coverstone-Carroll, V.: Near-optimal low-thrust orbit transfers generated by a genetic algorithm. J. Spacecr. Rockets **33**(6), 859–862 (1996). doi:10.2514/3.26850
18. Battin, R.H.: An Introduction to the Mathematics and Methods of Astrodynamics, 1st edn., 4th printing. AIAA, New York (1987)
19. Wall, B.J., Conway, B.A.: Shape-based approach to low-thrust rendezvous trajectory design. J. Guid. Control Dyn. (Feb 2008) (submitted)
20. Holmstrom, K., Goran, A., Edvall, M.: User's Guide for Tomlab 4.0.6. Tomlab Optimization, Sweden (2003)
21. Goldberg, D.: The Design of Innovation: Lessons from and for Competent Genetic Algorithms. Kluwer Academic Publishers, Boston (2002)
22. Gill, P., Murray, W., Saunders, M.: User's Guide for SNOPT 5.3: A Fortran Package for Large-scale Nonlinear Programming. Stanford University, Stanford (1998)
23. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley, New York (1988)
24. Floudas, C.A.: Deterministic Global Optimization: Theory, Methods and Applications. Kluwer Academic Publishers, Boston (2000)