https://ntrs.nasa.gov/search.jsp?R=20160011111 2020-05-10T04:34:22+00:00Z

An Automated Solution of the Low-Thrust Interplanetary Trajectory Problem

Jacob A. Englander*, Bruce A. Conway†

Preliminary design of low-thrust interplanetary missions is a highly complex process. The mission designer must choose discrete parameters such as the number of flybys, the bodies at which those flybys are performed, and in some cases the final destination. In addition, a time-history of control variables must be chosen that defines the trajectory. There are often many thousands, if not millions, of possible trajectories to be evaluated, which can be a very expensive process in terms of the number of human analyst hours required. An automated approach is therefore very desirable. This work presents such an approach by posing the mission design problem as a hybrid optimal control problem. The method is demonstrated on hypothetical missions to Mercury, the main asteroid belt, and Pluto.

Nomenclature

$\mathbf{F}$	constraint vector
$\mathbf{u}$	control vector
$\mathbf{v}_\infty$	hyperbolic excess velocity vector (km/s)
$\mathbf{x}$	decision vector
ΔF_{max}	largest violation in $\mathbf{F}$
δ	flyby turn angle (°)
δ_{power}	power margin
$\dot{m}$	mass flow rate (kg/s)
η	thruster efficiency
$[\gamma_i]$	solar array performance coefficients
$[a_F ... e_F]$	mass flow coefficients

*Aerospace Engineer, Navigation and Mission Design Branch, NASA Goddard Space Flight Center, Greenbelt, MD, 20771, USA, Member AIAA

†Professor Emeritus, Department of Aerospace Engineering, University of Illinois at Urbana Champaign, 104 S Wright Street, Mail Code-236, Associate Fellow AIAA

$[a_{LV}...e_{LV}]$ launch vehicle performance coefficients

$\left[a_{s/c}...c_{s/c}\right]$ spacecraft bus power coefficients

$[a_T...e_T]$ thrust coefficients

μ gravitational constant of a body (km^3/s^2)

μ_{GA} GA mutation frequency

Ω right ascension of ascending node ($^\circ$)

ω arguement of periapse ($^\circ$)

ρ metric of infeasibility

σ_{LV} launch vehicle margin

τ decay rate of the power supply

$\widetilde{P}$ pseudo-period (s)

a semi-major axis (AU)

C_3 hyperbolic excess energy (km^2/s^2)

CR_{GA} GA crossover frequency

D duty cycle

e eccentricity

g_0 gravity on Earth at sea level (m/s^2)

i inclination ($^\circ$)

I_{sp} specific impulse (s)

M mean anomaly ($^\circ$)

m spacecraft mass (kg)

N number of time-steps

$n_{available}$ number of available thrusters

N_{elite} number of elite unique individuals to hold between GA generations

N_{gen} number of generations to run the GA

$N_{parents}$ number of parents in the GA parent pool

$N_{stall-gen}$ number of generations after which the GA stops if no improvement occurs

P power available to the propulsion system (kW)

P_{0-BOL} power delivered by the power supply at 1 AU on the day of launch (kW)

$P_{generated}$ power generated by the power supply (kW)

P_{max} maximum power for a thruster (kW)

P_{min} minimum power for a thruster (kW)

$P_{s/c}$ power required by the spacecraft bus (kW)

r distance from a body (km)

r_a apoapse distance (km)

s_{mb} backward-propagated state vector at a match-point

s_{mf} forward-propagated state vector at a match-point

T thrust (N)

t_0 initial time (s)

t_f final time (s)

GA genetic algorithm

HOC hybrid optimal control

HOCP hybrid optimal control problem

MBH monotonic basin hopping

SNOPT Sparse Nonlinear Optimizer

I. Introduction

Preliminary design of low-thrust interplanetary missions is a complex, challenging problem which is often very expensive in terms of both human-hours and computer-hours. In any interplanetary mission design, the designer must select the appropriate launch date, flight time, maneuvers, and in some cases a sequence of planetary flybys. In some cases, such as small body missions, the trajectory designer is also tasked with choosing appropriate science targets that satisfy both scientific need and spacecraft capability. In the case of low-thrust missions, the designer must also choose a time history of control variables, such as thrust magnitude and direction. Both the discrete sequence-choosing problem and the continuous trajectory optimization problem are challenging and computationally expensive, and so an efficient, automated technique is highly desirable.

The traditional method of preliminary design is to approximate the low-thrust trajectory with a low-fidelity method such as a ballistic arc [1] or a shape-based approximation [2, 3, 4] and then to create a low-fidelity multiple-flyby trajectory by linking together many such arcs. A grid search tool [1] or a graphical tool combined with the designer's intuition [5] are then used to find the best low-fidelity solution. This solution is in turn used as an initial guess for a gradient-based low-thrust optimizer. Several such tools exist, including Mission Analysis Low-Thrust Optimization (MALTO) [6] and Gravity Assisted Low-thrust Local Optimization Program (GALLOP) [7]. This approach suffers from two significant drawbacks. First, the gradient-based optimizers converge to the locally optimal solution in the neighborhood of the initial guess and there is no guarantee that the best ballistic trajectory is in the neighborhood of the best low-thrust trajectory. In fact, the optimal flyby sequence or even the optimal target selection for a small body mission may not be the same between the best impulsive and low-thrust solutions. Second, it is computationally expensive to evaluate every possible ballistic solution in a grid and expensive in terms of human analyst time to try every promising candidate in a low-thrust solver and since there are frequently multiple locally optimal solutions it is not immediately obvious which possible solution the analyst should focus on. An automated method that intelligently explores the design space is highly desirable. Such a method should be capable of finding solutions (a) with less cost, in terms of human analyst time, (b) that may not be easily findable with a grid search or intuitive approach, and (c) without taking longer, in terms of total elapsed time, than other methods.

One such automated approach is to formulate the interplanetary design problem as a hybrid optimal control problem (HOCP). An HOCP is an optimization problem that is composed of two separable sub-problems, one with discrete variables and the other with continuous variables [8, 9]. For interplanetary design, the first problem is to choose the discrete parameters that define the mission, such as number of flybys, choice of flyby bodies, and, for some types of missions, the destination. The second problem is to find the parameters, such as launch date, flight times, flyby altitudes, encounter velocity vectors, and thrust program that characterize the optimal trajectory for each set of discrete parameters. An HOCP can be solved using two nested optimization loops. The "outer-

loop" solves the integer programming problem defining the discrete parameters. Each candidate solution to the "outer-loop" problem defines an "inner-loop" trajectory optimization problem. This approach was demonstrated first by Chilan, Wall, and Conway [10] for trajectories without flybys and then by Englander, Conway, and Williams for trajectories that include flybys and impulsive chemical propulsion [11].

Other researchers have addressed components of the outer-loop problem. Gad and Abdelkhalik [12, 13] solve the multiple-flyby problem with impulsive, chemical thrust by using a single genetic algorithm (GA) optimization loop. Another method, by Vasile and Campagnola [14], uses a set of successive deterministic algorithms to find candidate low-thrust, multiple flyby trajectories.

In this work we present a new framework for optimization of low-thrust interplanetary trajectories where the flyby sequence, and sometimes the destinations themselves, are not known *a priori*. The mission design problem is formulated as an HOCP where the outer-loop chooses the number of flybys, the identity of the flyby bodies, and, when appropriate, the destination. The outer-loop is based on the "null-gene" transcription presented by Englander, Conway, and Williams [11] and a discrete GA. The inner-loop is based on the Sims-Flanagan transcription [15] combined with the monotonic basin hopping (MBH) global search algorithm [16, 17, 18, 19]. The method is demonstrated on a hypothetical missions to Mercury, the main asteroid belt, and Pluto.

The algorithms described in this work are available as part of the open-source Evolutionary Mission Trajectory Generator (EMTG) project [20]. The authors encourage the reader to examine and use them and also welcome opportunities for collaboration.

II. Physical Modeling

A. Mission Architecture

Three trajectory components or trajectory divisions are defined in this work: *missions*, *journeys*, and *phases*. A *mission* is a top-level container that encompasses all of the events including departures, arrivals, thrust arcs, coast arcs, and flybys. A *journey* is a set of events within a mission that begin and end at a target of interest. The boundary conditions can be a powered or unpowered gravity assist, a planetary encounter, a small body intercept or rendezvous, or simply a state vector. For example, the interplanetary cruise portion of the Cassini mission was composed of a single journey that began at Earth and ended at Saturn. JAXA's Hayabusa mission, which rendezvoused and took samples from near Earth asteroid Itokawa, had two journeys - one from Earth to Itokawa, and one from Itokawa to Earth. NASA's Dawn mission is also composed of two journeys, one from Earth to Vesta and one from Vesta to Ceres. Each journey is composed of one or more *phases*. Like a journey, a phase begins at a planet and ends at a planet, but unlike the end points of a journey, the end points of a phase may represent a flyby of a body that is being used only to modify the trajectory of the spacecraft, i.e. a propulsive flyby. For example, the first journey of the Dawn mission may be considered to be a two-phase journey because it included a flyby of Mars. The number of

journeys in a mission is fixed *a priori* but the number of phases is not, and in the context of this work both the number of phases and the identity of the flyby planets in each phase may be chosen by the optimizer. Figure 1 is a block diagram of a mission using the journey/phase nomenclature.

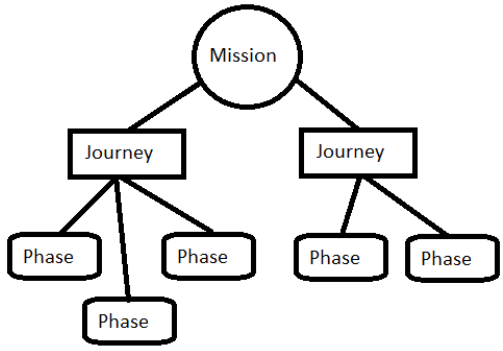

Figure 1. Anatomy of a Mission

B. Modeling of Dynamics

The dynamics of the spacecraft motion are modeled using the Sims-Flanagan transcription (SFT), a widely used method in which the continuous-thrust trajectory is discretized into many small time steps, and the thrust applied during each time step is approximated as a small impulse placed at the center of the time step. The trajectory is propagated between control points by solving Kepler's problem [15]. The SFT, when used with a nonlinear programming (NLP) problem solver such as the Sparse Nonlinear Optimizer (SNOPT)[21] and a suitable initial guess, is very fast and robust. It is considered to be a "medium-fidelity" transcription and is used in software packages such as MALTO [6], GALLOP [7], and Parallel Global Multiobjective Optimizer (PaGMO) [22].

In the classical SFT, the optimizer chooses the three components of an impulsive $\Delta\mathbf{v}$ vector at the center of each time-step. In order to improve the robustness of the solver, a modified transcription known as "up-to-unit vector control" [23] is used in this work, where instead of choosing the $\Delta\mathbf{v}$ vector directly the optimizer instead chooses a control 3-vector in $[-1.0, 1.0]$ that is multiplied by the maximum Δv that the spacecraft can produce in that time-step. The magnitude of the control vector is bounded in the range $[0.0, 1.0]$, i.e.,

$$\Delta\mathbf{v}_i = \mathbf{u}_i \Delta v_{max,i}, \|\mathbf{u}_i\| \leq 1.0 \tag{1}$$

where

$$\Delta v_{max,i} = \frac{D n_{available} T \left(t_f - t_0\right)}{mN} \tag{2}$$

where D is the thruster duty cycle, $n_{available}$ is the number of available thrusters, T is the maximum available thrust from one thruster, t_0 and t_f are the beginning and ending times of the time step, m

is the mass of the spacecraft at the center of the time step, and N is the number of time steps in the phase. This modified SFT is used in MALTO, PaGMO, and in this work.

The spacecraft state is propagated forward from the first endpoint (i.e. planet) in each phase and backward from the second endpoint. The trajectory is propagated by solving Kepler's equation and the spacecraft mass is propagated by assuming a constant mass flow rate across the each time-step. The specific Kepler propagator algorithm used here is a Laguerre-Conway method [24]. A set of nonlinear constraints are applied to ensure continuity in the center of the phase,

$$\mathbf{s}_{mf} - \mathbf{s}_{mb} = \begin{bmatrix} \Delta x & \Delta y & \Delta z & \Delta v_x & \Delta v_y & \Delta v_z & \Delta m \end{bmatrix} = \mathbf{0} \tag{3}$$

where Δx, Δy, and Δz are the position constraint defects, Δv_x, Δv_y, and Δv_z are the velocity constraint defects, and Δm is the mass constraint defect.

The optimizer also chooses the initial and final velocity vectors for each phase. If a phase begins with a launch, the magnitude of the initial velocity vector is used with a launch vehicle model to determine the initial mass of the spacecraft as described later in this work. If a phase begins with a planetary flyby, two nonlinear constraints are applied to ensure that the flyby is feasible. First, the magnitudes of the incoming and outgoing velocity vectors with respect to the planet must be equal,

$$v_\infty^+ - v_\infty^- = 0 \tag{4}$$

where v_∞^- and v_∞^+ are the magnitudes of the velocities before and after the flyby, respectively. Second, the spacecraft may not fly closer to the planet than some user-specified minimum flyby distance:

$$\frac{\mu_{planet}}{v_\infty^2}\left[\frac{1}{\sin(\frac{\delta}{2})} - 1\right] - (r_{planet} + h_{safe}) \geq 0 \tag{5}$$

where

$$\delta = \arccos\left[\frac{\mathbf{v}_\infty^- \cdot \mathbf{v}_\infty^+}{|v_\infty^-|\,|v_\infty^+|}\right] \tag{6}$$

Here μ_{planet} is the gravitational parameter of the planet, r_{planet} is the radius of the planet, δ is the flyby turn angle, and h_{safe} is the user-defined minimum altitude.

Figure 2 is a diagram of a simple low-thrust mission to Jupiter with one Earth flyby using the multiple gravity assist with low-thrust (MGALT) model. Here the Earth flyby occurs approximately one year after launch. The continuity constraints are deliberately left unsatisfied in the diagram to illustrate where they must be applied.

There are four significant advantages to using the SFT. First, the optimal objective function value for a Sims-Flanagan trajectory design is usually very close to the optimal cost value for a higher-fidelity version of the same trajectory. Second, a low-thrust trajectory generated using

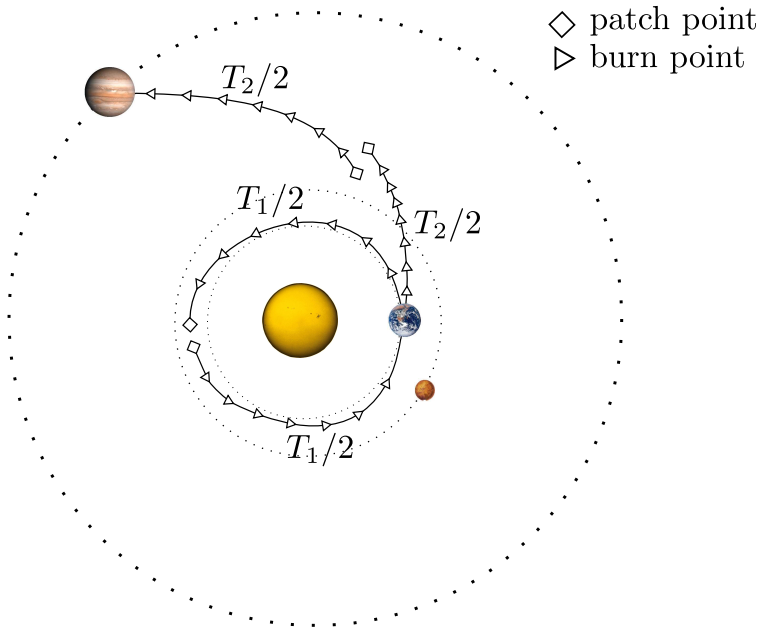

Figure 2. Schematic of a two-phase trajectory using the SFT

the SFT makes a very good initial guess for a higher-fidelity trajectory design. Third, the SFT is very fast because it does not require numerical integration of differential equations. Fourth, the convergence of an NLP solver solving a Sims-Flanagan problem is very robust to poor initial guesses, making it ideal for an automated design approach. It is notable that the more time-steps used, the more accurate the SFT is. Higher numbers of time-steps yield more realistic solutions but at a high computational cost as the size of the Jacobian increases as the square of the number of time-steps. It is desirable to have at least 10 time-steps per revolution about the central body. Commonly the authors start with 20 time-steps per phase and then increase from there until there are at least 10 time-steps per revolution in the final answer.

The formulation described here may be generalized to any choice of central body. While the examples in this work are all heliocentric missions, the model and the associated software can work about any body in the solar system or even a fictitious body. However, the Sims-Flanagan transcription, and all two-point direct shooting transcriptions, has a significant drawback. While it is very appropriate for trajectories in which the number of revolutions about the central body is small (less than ten) and the orbit changes significantly in each revolution, it is not appropriate for trajectories with many (tens to thousands) of revolutions about the central body and a very small change in orbit in each revolution. This property is a consequence of direct parametrization of the control - as the number of revolutions increases a two-point direct shooting transcription suffers from both the curse of sensitivity an the curse of dimensionality. However this drawback is not important for most interplanetary trajectories and does not affect the problems solved in this work.

C. Launch Vehicle, Propulsion, Power, and Ephemeris Modeling

Low-thrust trajectories are inextricably coupled to the specific hardware used by the spacecraft. The optimal trajectory for one combination of launch vehicle, propulsion system, and power system will not be the optimal trajectory for a different hardware combination. Realistic modeling of these three systems and of system margins is therefore applied in this work. In addition, accurate ephemeris modeling is provided by the SPICE toolkit [25]. **SPICE!** (**SPICE!**) is very accurate but does have some properties which add challenge to the trajectory optimization process. Most importantly the ephemerides provided by SPICE are not continuously differentiable. An alternative may be to use a spline approximation to **SPICE!** as described by Arora and Russell [26]. This would make the inner-loop solver more robust.

Launch vehicle performance, measured as delivered mass in kg as a function of C_3 in km^2/s^2, is modeled as a 5th degree polynomial:

$$m_{delivered} = (1 - \sigma_{LV}) \left(a_{LV} C_3^5 + b_{LV} C_3^4 + c_{LV} C_3^3 + d_{LV} C_3^2 + e_{LV} C_3 + f_{LV} \right) \tag{7}$$

where σ_{LV} is the user-defined launch vehicle margin in $[0, 1]$ and the other coefficients are chosen by a curve fit to published launch vehicle performance data available at the Kennedy Space Center (KSC) expendable launch vehicle (ELV) performance website [27]. Several types of electric thrusters are modeled in this work. The simplest method is to set constant values for thrust and specific impulse (I_{sp}). However, because both thrust and I_{sp} are functions of available power, which is itself a function of distance from the sun, constant values are not accurate in most cases. They are, however, appropriate when it is known *a priori* that the spacecraft's entire mission will take place in a regime where the propulsion system may be fully powered. This assumption is reasonable for a mission to the inner planets, Venus and Mercury. When the constant thrust and I_{sp} model is used, mass flow rate $\dot{m}$ is computed by:

$$\dot{m} = \frac{T}{I_{sp} g_0} \tag{8}$$

where g_0 is the acceleration due to gravity on Earth at sea level, 9.806650 m/s^2.

The second method of thruster modeling is to provide performance polynomials for T and $\dot{m}$ as a function of available power in kW of the form:

$$T = a_T P^4 + b_T P^3 + c_T P^2 + d_T P + e_T \tag{9}$$

$$\dot{m} = a_F P^4 + b_F P^3 + c_F P^2 + d_F P + e_F \tag{10}$$

The coefficients $\{a_T, b_T, c_T, d_T, e_T, a_F, b_F, c_F, d_F, e_F\}$ are drawn from the published literature and are curve fits to laboratory test data. Each thruster also has an associated minimum power P_{min} and maximum power P_{max}. If the available power P is less than P_{min}, then T and $\dot{m}$ are zero. If

P is greater than P_{max} then the performance polynomials are evaluated at P_{max}. When thruster performance polynomials are used, one must also determine the number of thrusters that can fire simultaneously, n_{active}, which is less than or equal to the user-defined number of available thrusters, $n_{available}$. In this work, n_{active} is calculated via one of two algorithms. In the first algorithm, "maximum number of thrusters," the spacecraft must fire as many thrusters as possible with the currently available power, even if doing so requires throttling the thrusters to a lower power level. In the second algorithm, "minimum number of thrusters," the spacecraft must fire as few thrusters as possible to fully use the available power and therefore at a higher power level per thruster than in "maximum number of thrusters." The most appropriate throttle logic for a given mission depends on whether the thruster being used is most efficient at its highest power level, in which case "minimum number of thrusters" is most appropriate, or if it is most efficient at intermediate power levels in which case "maximum number of thrusters" is more effective.

The third thruster model in this work computes available thrust T as a function of available power, fixed propulsion system efficiency η and I_{sp},

$$T = \frac{2\eta P}{I_{sp} g_0} \tag{11}$$

where I_{sp} may be user-defined, chosen by the optimizer and fixed over the length of the mission, or chosen by the optimizer at each time-step to simulate a variable I_{sp} (VSI) propulsion system.

The available power P is the difference between the power generated by the spacecraft $P_{generated}$ and the power required to operate the spacecraft bus $P_{s/c}$,

$$P = (1 - \delta_{power}) \left(P_{generated} - P_{s/c} \right) \tag{12}$$

where δ_{power} is a user-defined power margin.

In this work, the power delivered by a solar array is given by [28]:

$$P_{generated} = \frac{P_0}{r^2} \left(\frac{\gamma_0 + \gamma_1/r + \gamma_2/r^2}{1 + \gamma_3 r + \gamma_4 r^2} \right) \tag{13}$$

where the γ_i are user-defined solar panel coefficients, r is the distance between the sun and the spacecraft in Astronomical Unit (AU) and P_0 is the "base power" delivered by the array at 1 AU. P_0 is in turn a function of the time since launch,

$$P_0 = P_{0-BOL} \left(1 - \tau\right)^t \tag{14}$$

where P_{0-BOL} is the base power delivered by the array at 1 AU on the day of launch, τ is the decay rate of the solar arrays measured as a percentage per year, and t is the time since launch in years. Equation (14) may also be used to model the decay of an radioisotope thermal generator (RTG) or

advanced Stirling radiosotope generator (ASRG) power system.

The power required by the spacecraft bus $P_{s/c}$ is modeled as a polynomial,

$$P_{s/c} = a_{s/c} + \frac{b_{s/c}}{r} + \frac{c_{s/c}}{r^2} \tag{15}$$

where $a_{s/c}$, $b_{s/c}$, and $c_{s/c}$ are chosen by the user.

D. Lower and Upper Bounds

The stochastic optimizers used in this work require that the mission design problem be posed in a bounding box. If the bounds on the decision variables are too broad, the solvers will struggle to find a good solution. If the bounds are too constraining than there is a risk of pruning away the optimal solution. Furthermore, because the trajectory design problem is being solved as the inner-loop of a HOCP, there is no opportunity for a human analyst to set appropriate bounds for every possible problem. Some types of bounds, such as on launch C_3, stay time at intermediate science targets, and maximum flyby velocities may be chosen by an analyst and applied to the entire set of sub-problems generated by the outer-loop. However it is helpful to use a set of automated bounds-choosing laws to choose the flight time for each phase.

In this work, the flight time for each phase is driven by the "pseudo-period," or $\widetilde{P}$, of the boundary bodies,

$$\widetilde{P} = 2\pi\sqrt{\frac{r_a^3}{\mu}} \tag{16}$$

where r_a is the apoapse distance of the body. The apoapse distance is used to construct a pseudo-period instead of using the semi-major axis to construct the true period because low-thrust flight times to bodies on highly eccentric orbits often cannot be predicted just by looking at the orbital period of the body. Rather, the time necessary to travel to the outer solar system is relevant. The pseudo-period concept introduced here has proven useful in several mission design cases. $\widetilde{P}$ is linked to the definition of a length unit (LU) which is chosen *a priori* by the user. In this work 1 LU is equal to 1 AU. The flight times for various types of phases are derived from the $\widetilde{P}$ for the boundary bodies and are shown in Table 1.

Table 1. Automated choice of phase flight-time bounds

Case	Lower Bound	Upper Bound
repeated flyby of same planet	$\widetilde{P}/2$	$20\widetilde{P}$
outermost body has $a < 2LU$	$0.1min\left(\widetilde{P}_1, \widetilde{P}_2\right)$	$2.0max\left(\widetilde{P}_1, \widetilde{P}_2\right)$
outermost body has $a \geq 2LU$	$0.1min\left(\widetilde{P}_1, \widetilde{P}_2\right)$	$max\left(\widetilde{P}_1, \widetilde{P}_2\right)$

III. Outer-Loop Optimization of the Mission Sequence

A. Outer-Loop Transcription

The mission design problem in this work is posed as two nested optimization problems, an "outer-loop" discrete optimization problem and an "inner-loop" real-valued optimization problem. The outer-loop solves a multi-objective integer programming problem whose candidate solutions are themselves instances of the inner-loop trajectory optimization problem. Each candidate solution is then evaluated for its fitness by running the inner-loop solver, and then the resulting population of solutions may be compared and a new population created.

In this work, the outer-loop solver is used to choose two mission characteristics: the destination(s) for each journey and/or the number of flybys and identity of the flyby targets. If the outer-loop is being used to select journeys, the user specifies *a priori* a menu of candidate destinations for each journey. The outer-loop picks one choice from each menu and uses them to define a trajectory optimization problem that is then passed to the inner-loop.

Flyby sequence selection is similar to journey destination selection except that one does not always know how many flybys are to be performed. A "null-gene" technique is used to choose the number and identity of flyby bodies [11]. The analyst provides a list of acceptable flyby bodies and a maximum number of flybys for each journey. Then, for each potential flyby, the outer-loop may select from a list containing the specified acceptable bodies and also a number of "null" options equal to the number of acceptable bodies. The outer-loop therefore has an equal probability of selecting "no flyby" for each opportunity as it does to select a flyby. This technique has been shown to be very effective for designing multi-flyby interplanetary missions and has been used to reproduce the Cassini [11] trajectory.

B. Outer-Loop Optimization via GA

The outer-loop problem is an integer programming problem that lends itself well to solution by a GA. First proposed by Holland[29], the GA is a population-based search heuristic that mimics the evolution of a species.

The GA has been shown to be effective as an outer-loop solver for hybrid optimal control problem spacecraft trajectory optimization [10, 30], possibly because the choice of a sequence of discrete events in the trajectory is analogous to the choice of a sequence of genes in the biological systems that the GA is designed to mimic. The GA was previously applied by these authors to the HOCP of chemical propulsion trajectory design [11].

The GA first generates a random population P of decision vectors, or *chromosomes*, in an N_p-dimensional vector space. The algorithm is run for a number of generations. At each generation, the fitness of every individual in the population is evaluated. A *selection* operator is employed to choose the best $N_{parents}$ individuals who will become the parents of the next generation. A *crossover* operator is then used to create child vectors from the chosen parent vectors, replacing

a user-specified fraction of the population. This fraction is called the *crossover ratio*, or CR_{GA}. Then, with some small probability μ_{GA}, random *mutation* occurs in the resulting new population. Unfortunately this can sometimes result in the GA discarding the optimal solution after it is found. It is therefore desirable to preserve a small number N_{elite} individuals, termed *elite*, without modification in each generation. The process is then repeated either for a set number of generations N_{gen} or until some user-specified convergence criterion is met, such as the GA "stalling," that is, failing to improve the best known solution for $N_{stall-gen}$ generations. The basic GA is listed in the appendix.

An integer-encoded GA is used in this work. The selection operator is "tournament selection," where a "pool" of parents is selected from the population. The parent pool is then sorted and the best member of the pool is chosen to serve as a parent. This process is repeated every time a parent is required (twice per crossover operation). The well-known binary crossover and uniform mutation operators are used[29].

The elitism operator used in this work is not the classical elitism operator in which the best N_{elite} individuals are retained. Instead, the best N_{elite} *unique* individuals are retained. The elitism operator preserves the best members of the population and encourages diversity at the same time.

C. Parallel Outer-Loop Optimization

The evaluation of the objective functions for a candidate solution is very expensive - it requires solving the entire inner-loop trajectory optimization problem and can take at best several seconds, in most cases many minutes, and in some cases hours. Fortunately each inner-loop subproblem is independent of the others so it is natural to evaluate them in parallel. The pool of *N* inner-loop subproblems to be evaluated is distributed over *P* processor cores. If *N* exceeds *P*, the subproblems are agglomerated into *P* task pools with one assigned to each processor. The run-time is further decreased by saving each candidate solution to the outer-loop problem so that none need be evaluated more than once. Therefore the number of inner-loop instances to be run for each outer-loop generation tends to decrease and the algorithm speeds up with each generation of the GA.

There are two parallel processing paradigms: shared memory multiprocessing and distributed memory multicomputing. Shared memory multiprocessing, also known as threading, is easier to implement but is not appropriate for this work because it requires all processes to be "thread-safe," i.e. not interfere with each other even while operating in the same memory space. However the SPICE ephemeris reader used in this work is not thread-safe and therefore the authors never pursued shared-memory multiprocessing. Instead the algorithm described in this work is implemented using the distributed memory multicomputing library message passing interface (MPI) [31] in which each processor runs an independent process in distinct memory spaces. If in the future an alternative ephemeris reader such as Arora and Russell's FIRE [26] were used in place of SPICE, shared-memory multiprocessing may become feasible.

IV. Inner-Loop Trajectory Optimization

A. Stochastic Global Search via Monotonic Basin Hopping and SNOPT

Like most other optimization problems, low-thrust trajectory design problems require an initial guess. Such initial guesses are often generated using Lambert's method, by solving for the optimal impulsive trajectory, by using shape-based trajectory approximations [2, 3, 4], or by experience-driven intuition. Once an initial guess is found for one version of the problem, the solution to the first problem may be used as an initial guess for related problems. However there are drawbacks in this approach. First, the optimal low-thrust solution may not be in the neighborhood of the optimal solution to the low-fidelity Lambert version of the same problem. Second, sometimes the solution space bifurcates - that is, there may be multiple locally optimal solutions and it is difficult to know *a priori* which solution family is the best. This complexity can cause a human analyst to spend a great deal of time trying initial guesses that do not lead to the globally optimal trajectory. Finally, and most importantly, the inner-loop solver in an HOCP solver must be able to autonomously solve any trajectory optimization problem posed by the outer-loop GA. Since there is no room for human intervention, there is also no room for a human to provide an initial guess. Therefore an automated solution method is required.

Recent research in low-thrust trajectory optimization has led to the creation of stochastic search methods that do not require an initial guess [32, 33, 34, 28, 17, 11, 35, 18, 36, 37, 38, 19]. These methods are naturally suited for use in an HOCP framework. The stochastic search method used in this work is monotonic basin hopping (MBH) [39]. MBH is an algorithm for finding globally optimal solutions to problems with many local optima. MBH works on the principle that many real-world problems have a structure where individual local optima, or "basins" tend to cluster together into "funnels" where one local optimum is better than the rest. A problem may have several such funnels. MBH was originally developed to solve molecular conformation problems in computational chemistry, but has been demonstrated to be effective on various types of interplanetary trajectory problems [16, 17, 40, 35, 18, 36, 41]. MBH is a two-stage solver that alternates back and forth between a stochastic global search stage and a deterministic NLP stage.

First, an initial point $\mathbf{x}$ is randomly chosen. The NLP solver is run using $\mathbf{x}$ as the initial guess. If the NLP solver finds a feasible solution, then that new point $\mathbf{x}^*$ is adopted as the new current point. If the NLP solver does not find a feasible solution, then a new random point is chosen. Once a feasible solution is found, MBH will attempt to "hop" from the feasible and locally optimal $\mathbf{x}^*$ to a better point. The hop is a two-step process: first a small random perturbation vector is added to $\mathbf{x}^*$, producing a new $\mathbf{x}'$, and then the NLP solver is run. If the resulting solution is both feasible and superior to $\mathbf{x}^*$, then it is adopted as the new $\mathbf{x}^*$ and the hopping process begins again. Otherwise, MBH attempts a new hop from the current $\mathbf{x}^*$. Each feasible solution is stored in an archive.

MBH is run until either a specified number of iterations (trial points attempted) or a maximum CPU time is reached, at which point the best solution stored in the archive is returned as the

solution to the outer-loop. The version of MBH used in this work has two parameters - the stopping criterion and the type of random step used to generate the perturbed points $\mathbf{x}'$. Further, in this work the random step is drawn from a bi-directional Pareto distribution. The bi-directional Pareto distribution will usually generate small steps that allow MBH to *exploit* the local funnel around the current best solution. However some of the steps generated by the bi-directional Pareto distribution will be much larger, in some cases spanning the entire solution space. These larger steps allow MBH to *explore* the full problem. This approach is known to be robust on many complex low-thrust problems [19].

The algorithm is improved from that found in Englander and Englander[19] by the addition of a "time hop" operator [42] by which, in some small fraction of iterations with probability $\rho_{time-hop}$, the variable representing the time of flight between two bodies is incremented or decremented by the synodic period of those two bodies. Cassioli *et al.*[42] have found that the time hop operator can increase the efficiency of MBH in some circumstances. In addition, the variant of MBH used in this work employs a "feasible point finder" consisting of a penalty that is applied to any solutions for which the NLP stage does not converge. The penalty function is equal to the 2-norm of the constraint violation after the NLP solver halts, allowing MBH to distinguish between two infeasible solutions instead of having to be run repeatedly with a random starting location until a feasible solution is found as in classical MBH. The pseudocode for MBH is listed in the appendix.

If the inner-loop solver finds a feasible solution, the objective function value is returned to the outer-loop solver so that it may be used to rank candidate missions. If the inner-loop does not find a feasible solution, a metric of infeasibility is returned to the outer-loop so that it may compare feasible solutions against infeasible and even different infeasible solutions against each other. The infeasibility metric used here is given in Eq. (17), where ΔF_{max} is the largest-magnitude constraint violation and $\|\mathbf{x}\|$ is the 2-norm of the decision vector. This metric is the same as the constraint check used in SNOPT,

$$\rho = \frac{\Delta F_{max}}{\|\mathbf{x}\|} \tag{17}$$

The outer-loop then assigns the infeasible mission an objective function value of,

$$J = 1.0 \times 10^{100} + \rho \tag{18}$$

The constant term in Eq. 18 is very large to encourage the outer-loop to discard the infeasible solution in the next generation and the ρ term allows the outer-loop to compare two infeasible solutions in the selection operator and discard whichever is worse.

The MBH+NLP optimization algorithm in this work is efficient and does not require an initial guess. MBH is most useful when one does not have much *a priori* information about the solution

as is *always* the case when solving an inner-loop trajectory optimization problem inside an hybrid optimal control (HOC) mission design problem.

V. Examples

A. Low-Thrust Mission to Mercury

The first example case is a low-thrust mission to Mercury, based on a version of ESA's BepiColombo mission[43, 44, 17]. BepiColombo will use low-thrust electric propulsion and flybys of Venus and Mercury to reach its final orbit about Mercury. The specific version of BepiColombo studied here is the original concept study that assumed launch in 2009. This specific problem was chosen to enable comparison with the results of Yam *et al.* [17], who used monotonic basin hopping with nonlinear programming to optimize an Earth-Venus-Venus-Mercury-Mercury-Mercury (EVVYYY) trajectory with a 2009 launch. Because this mission travels to the inner solar system and will always have enough power for its propulsion system, and also for better comparison with Yam *et al.*, the thrust, specific impulse, and initial mass of the spacecraft are fixed. Table 2 describes the problem assumptions.

Table 2. Assumptions for the BepiColombo Mission

Option	Value
Maximum launch v_∞	1.925
Arrival type	intercept (match position) with bounded v_∞
Maximum arrival v_∞	0.5 km/s
Launch window open date	8/1/2009
Launch window close date	4/27/2012
Flight time upper bound	15 years
Propulsion type	fixed I_{sp} and thrust
Thrust	0.34 N
I_{sp}	3200 s
Initial mass	1300 kg
Number of time steps per phase	10
Maximum number of flybys	8
Outer-loop GA Population Size	60
Outer-loop GA number of generations	200
Outer-loop GA CR_{GA}	0.3
Outer-loop GA μ_{GA}	0.15
Inner-loop MBH run time	20 minutes
Inner-loop MBH Pareto α	1.5
Inner-loop MBH maximum iterations	100,000 (never reached)

The BepiColombo problem was run 10 times on a 60-core Intel Xeon E7-4890 running at 2.8 GHz with 500 GB of RAM. Each run took 67 hours. The 10 trials were conducted because both the outer-loop and the inner-loop are driven by stochastic processes and therefore the HOCP algorithm may not deliver the same results in every trial. Table 3 shows that the best sequence found was EEVVYY with a delivered mass of 1100 kg, compared to the EVVYYY solution delivering 1064 kg found by Yam *et al.*. This sequence was found in 3 of the 10 runs. Note from Table 3 that there are several solutions to the BepiColombo problem with very similar objective function values and in fact there are several very similar solutions to each inner-loop problem. In every case where the optimal EEVVYY sequence was not found to have the best objective function value, that sequence was considered by the HOCP automaton but the inner-loop failed to find the global best solution and therefore that sequence did not trade well against others. This result suggests that the most productive direction of future research on the HOCP automaton is to improve the performance of the inner-loop solver. It is also worth noting that sometimes the outer-loop found the best sequence very early - as early as generation 20, but in other cases the best solution was not identified until much later. A larger population size could mitigate this problem but this result does confirm that it

is valuable to run the outer-loop for at least the 200 generations used in this example.

Table 3. Performance of the HOCP automaton on the BepiColombo problem

Run	Best Sequence	Delivered mass (kg)	Generation of best sequence
1	EEVVYY	1100	175
2	EEVVYY	1091	129
4	EEVVYY	1078	84
3	EVVYY	1085	20
5	EEVVY	1069	151
6	EVVYY	1057	146
7	EVVYY	1050	194
8	EEVVY	1049	142
9	EEVVY	1043	145
10	EVVVY	1029	32
Yam *et al.*	EVVYYY	1064	N/A

Table 4 shows the best 20 (of 167) feasible solutions found by Run 3 of the HOCP automaton. Note that there are many solutions with very similar objective function values. Table 5 lists the itinerary of the optimal EEVVYY mission. Figure 3 is a plot of that mission.

Table 4. Top 20 solutions found by Run 3 of the HOCP automaton on the BepiColombo problem

Sequence	Delivered mass (kg)	Generation found
EEVVYY	1100	175
EVVYY	1048	148
EEVVY	1044	89
EEVYY	1039	118
EEVY	1023	5
EVVVYY	1022	175
EVVY	1015	186
EEEVVY	1014	118
EEEVY	1005	103
EVVVY	1001	143
EEVVVY	999	159
EVYY	986	49
EEEVVYY	975	166
EVEVVY	970	80
EEEVYY	966	176
EEVYYY	958	201
EVY	958	136
EVEVVYY	953	154
EVVYYY	950	0

Table 5. Itinerary for the optimal BepiColombo mission

Date	Event	Location	C_3 (km^2/s^2)	Flyby altitude (km)	Mass (kg)
7/19/2011	launch	Earth	3.7		1300.0
9/1/2016	flyby	Earth	13.5	4090	1280.3
12/3/2017	flyby	Venus	49.7	5194	1280.3
12/27/2020	flyby	Venus	62.0	300	1269.7
11/17/2024	flyby	Mercury	5.0	300	1126.8
3/8/2025	intercept	Mercury	0.3		1100

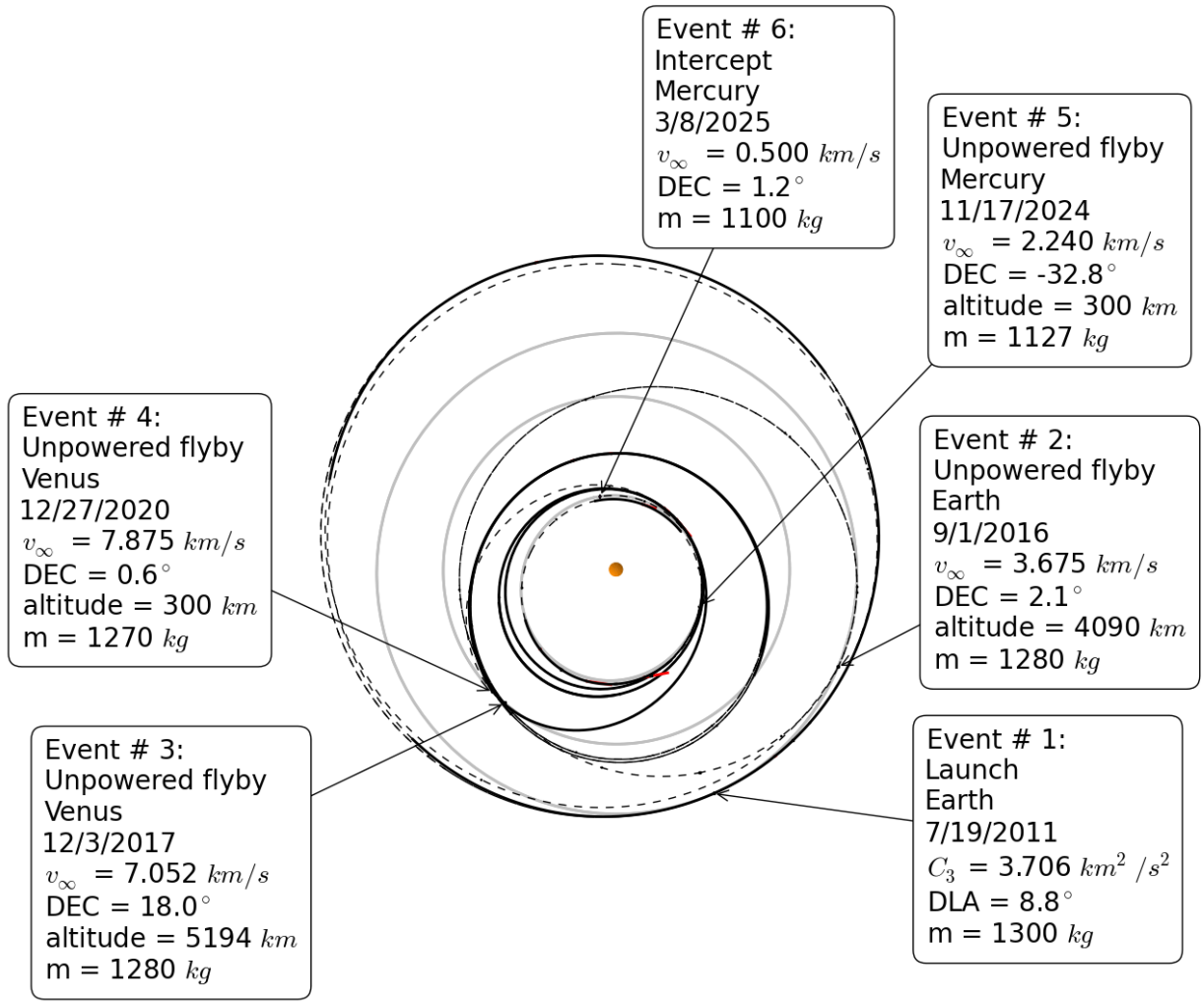

Figure 3. Optimal trajectory for the BepiColombo rendezvous mission

B. Main Belt Asteroid Rendezvous

The second example is of the type with an unspecified destination, a mission to visit two asteroids in the main belt. Any asteroid with a diameter of at least 200 km according to the JPL Small Bodies Database [45] is considered an acceptable target. The 23 bodies that meet this criterion are listed in Table 6. The outer-loop is allowed to choose any two of these bodies. Any candidate mission that visits the same body twice is assigned a poor fitness value so that the outer-loop evolves away from such missions. The outer-loop is also allowed to choose up two two flybys of Venus, Earth, or Mars before the first asteroid rendezvous and up to one flyby of Mars between asteroids. No pre-filtering is done on the space of possible missions - the purpose of this experiment is to see if the outer-loop can design the optimal mission with no *a priori* choices made. There are 38,088 solutions to the main belt asteroid rendezvous problem, including duplicate solutions that the outer-loop will discard. The number of possible solutions scales to the power of the number of rendezvous in the

mission, so if a third rendezvous were to be considered then filtering strategies would have to be introduced to keep the problem tractable.

Table 6. Main-belt asteroids with diameter greater than 200 km, reference epoch June 27th, 2015

Asteroid	a (AU)	e	i (°)	Ω (°)	ω (°)	M (°)
1 Ceres	2.77	0.08	10.6	80.3	72.7	138.7
2 Pallas	2.77	0.23	34.8	173.1	310.0	120.9
4 Vesta	2.36	0.09	7.1	103.8	151.2	75.2
10 Hygiea	3.14	0.11	3.8	283.4	312.1	264.5
511 Davida	3.16	0.19	15.9	107.6	337.8	88.4
704 Interamnia	3.06	0.15	17.3	280.3	95.3	202.2
52 Europa	3.09	0.11	7.5	128.7	344.4	343.0
31 Euphrosyne	3.16	0.22	26.3	31.1	61.5	201.1
15 Eunomia	2.64	0.19	11.7	293.2	97.5	323.4
16 Psyche	2.92	0.14	3.1	150.3	227.1	14.1
3 Juno	2.67	0.26	13.0	169.9	248.4	78.2
88 Thisbe	2.77	0.16	5.2	276.7	35.9	109.9
324 Bamberga	2.69	0.34	11.1	327.9	44.2	135.8
451 Patientia	3.06	0.08	15.2	89.3	337.2	205.6
532 Herculina	2.77	0.18	16.3	107.6	76.0	45.8
48 Doris	3.11	0.07	6.5	183.6	253.6	177.4
375 Ursula	3.13	0.11	15.9	336.6	342.5	45.9
45 Eugenia	2.72	0.08	6.6	147.7	88.7	90.3
29 Amphitrite	2.55	0.07	6.1	356.4	62.0	307.8
423 Diotima	3.07	0.04	11.2	69.5	200.8	163.4
13 Egeria	2.58	0.08	16.5	43.2	80.2	214.3
94 Aurora	3.16	0.09	8.0	2.6	61.0	62.5
19 Fortuna	2.44	0.16	1.6	211.2	182.2	195.5

The spacecraft in this second example is more complex than that in the BepiColombo example. Instead of supplying a fixed thrust, I_{sp}, and initial mass, this spacecraft is given two BPT-4000 Hall-effect thrusters [46] and a 15 kW solar array whose performance varies as per Eq. (13) with 800 W always reserved for the spacecraft bus. The spacecraft is constrained to use no more than 950 kg of propellant for interplanetary cruise plus an additional 25 kg for proximity operations at each body. The standard preliminary design margins are applied as per Oh *et al.* [47] - 15% power margin, 10% propellant margin, and a 90% thruster duty cycle. The "minimum number of thrusters" throttle logic is used. The spacecraft launches on a Falcon 9 v1.1 modeled as per Eq. (7). The inner-loop solver is permitted to underload the launch vehicle if it is optimal to do so, *i.e.* if a lower initial mass yields a higher final mass. A 60-day coast period is enforced after launch to allow for spacecraft checkout.

Table 7 lists the problem assumptions for the main belt tour mission. The main-belt tour problem was run 10 times on a 60-core Intel Xeon E7-4890 running at 2.8 GHz with 500 GB of RAM.

Table 7. Assumptions for the Main Belt Tour Mission

Option	Value
Launch window open date	1/1/2020
Launch window close date	9/27/2022
Flight time upper bound	10 years
Arrival condition	rendezvous (match position and velocity)
Launch vehicle	Falcon 9 v1.1
Launch asymptote declination bounds	$[-28.5, 28.5]$ (Kennedy Space Center)
Post-launch coast duration	60 days
Solar array $P0$	15 kW
Solar array coefficients γ_i	$[1, 0, 0, 0, 0]$
Spacecraft power coefficients $a_{s/c} - c_{s/c}$	$[0.8, 0, 0]$
Propulsion system	2 BPT-4000 in "high-I_{sp}" mode [46]
Throttle Logic	minimum number of thrusters
Propellant tank capacity	1000 kg (25 kg reserved for operations at each asteroid)
Duty cycle	90%
Power margin	15%
Propellant margin	10%
Number of time steps per phase	20
Target selection	outer-loop chooses any two from Table 6
Flyby sequence	
before first rendezvous	up to two flybys of Venus, Earth or Mars
between rendezvous	up to one flyby of Mars
Outer-loop GA Population Size	60
Outer-loop GA number of generations	200
Outer-loop GA CR_{GA}	0.3
Outer-loop GA μ_{GA}	0.15
Inner-loop MBH run time	5 minutes
Inner-loop MBH Pareto α	1.4
Inner-loop MBH maximum iterations	100,000 (never reached)

Each run took about 20 hours. Table 8 shows the best mission found by each run. Note that the best solution, E-M-4 Vesta-1 Ceres delivering 1506 kg after propellant margin was removed, was found only once. This result shows two flaws in the way that the problem was posed to the HOCP automaton. First, the outer-loop population size was too small and therefore there was insufficient diversity in the initial population. Second, the inner-loop solver was not run for a sufficient amount of time for MBH to locate the vicinity of the globally optimal trajectory for every candidate sequence. While it is not possible to guarantee that this will happen, the probability of MBH succeeding in finding the global optimum does scale with run time [19]. Note in Table 8 that the outer-loop considered less than 2000 possible sequences, only about 5% of the search space, in each trial. This sampling was not enough to find the global optimum every time.

Table 8. Performance of the HOCP automaton on the main belt tour problem

Run	Best Sequence	Generation of best sequence	Delivered mass (kg)	# sequences evaluated
1	E-4 Vesta-1 Ceres	20	1014	1613
2	E-4 Vesta-1 Ceres	29	1045	1552
3	E-M-4 Vesta-1 Ceres	21	1506	1628
4	E-4 Vesta-1 Ceres	77	1045	1584
5	E-M-19 Fortuna-48 Doris	77	1163	1549
6	E-4 Vesta-1 Ceres	19	1045	1454
7	E-4 Vesta-1 Ceres	81	1045	1566
8	E-M-4 Vesta-52 Europa	17	1213	1580
9	E-4 Vesta-1 Ceres	29	1045	1450
10	E-M-M-4 Vesta-423 Diotima	176	1061	1661

In an operational mission design environment, the HOCP automaton would be run with a larger population and with a longer inner-loop run time. The outer-loop would ideally explore a larger fraction of the total design space than the $\tilde{5}\%$ shown in this example. However a larger population would make the total run time for the solver longer, up to several days or for particularly difficult problems as long as two weeks on the computers available to these authors. For this paper, it was desired to run each example 10 times, and so example problems were constructed that were simple enough that they could be run quickly. Clearly the main-belt tour example problem is too challenging to be solved in such a short time via the methods presented in this work.

Table 9 shows the best 20 missions found by the HOCP automaton in Run 8, the run that found the global best solution. The best mission, E-M-4 Vesta-1 Ceres delivering 1506 kg, was found in generation 21 of 200 outer-loop generations. Table 10 shows the itinerary of the optimal mission, and Fig. 4 shows the trajectory. Figure 5 is a plot of the propulsion and power characteristics of the spacecraft as a function of time throughout the mission. It is interesting to note that while the spacecraft has two thrusters available, the number of active thrusters, represented by the "+" markers in Fig. 5, never rises above one thruster. This result is because in the optimal trajectory thrusting never occurs closer to the sun than 1.5 AU, at which point there is not sufficient power to require switching on a second thruster. Also note that the solution is highly dependent on the systems parameters, in this case power and propulsion. It would be of great value if the outer-loop could vary systems parameters as well as flybys and asteroids. This extension to the outer-loop is currently in development [48] and will be further explored in a future work.

Table 9. Convergence history for one run of the main-belt rendezvous problem

Sequence	Generation found	Dry mass (kg)
E-M-4 Vesta-1 Ceres	21	1506
E-19 Fortuna-88 Thisbe	16	1017
E-4 Vesta-1 Ceres	15	1014
E-4 Vesta-16 Psyche	25	936
E-19 Fortuna-48 Doris	22	902
E-4 Vesta-45 Eugenia	7	893
E-19 Fortuna-15 Eunomia	3	883
E-4 Vesta-423 Diotima	33	849
E-E-19 Fortuna-15 Eunomia	8	819
E-29 Amphitrite-324 Bamberga	9	765
E-1 Ceres-4 Vesta	9	744
E-19 Fortuna-324 Bamberga	6	736
E-88 Thisbe-15 Eunomia	8	734
E-19 Fortuna-16 Psyche	21	710
E-19 Fortuna-45 Eugenia	13	678
E-15 Eunomia-88 Thisbe	22	674
E-4 Vesta-94 Aurora	3	653
E-M-4 Vesta-94 Aurora	27	649
E-E-4 Vesta-423 Diotima	85	635
E-M-16 Psyche-45 Eugenia	52	612

Table 10. Itinerary for the optimal main-belt rendezvous mission

Date	Event	Location	C_3 (km^2/s^2)	Flyby altitude (km)	Mass (kg)
6/16/2022	launch	Earth	11.5		2481.4
2/5/2025	unpowered flyby	Mars	18.6	300	2358.7
8/9/2027	LT rendezvous	4 Vesta			1946.8
8/8/2028	departure	4 Vesta			1921.8
12/26/2031	LT rendezvous	1 Ceres			1592.8

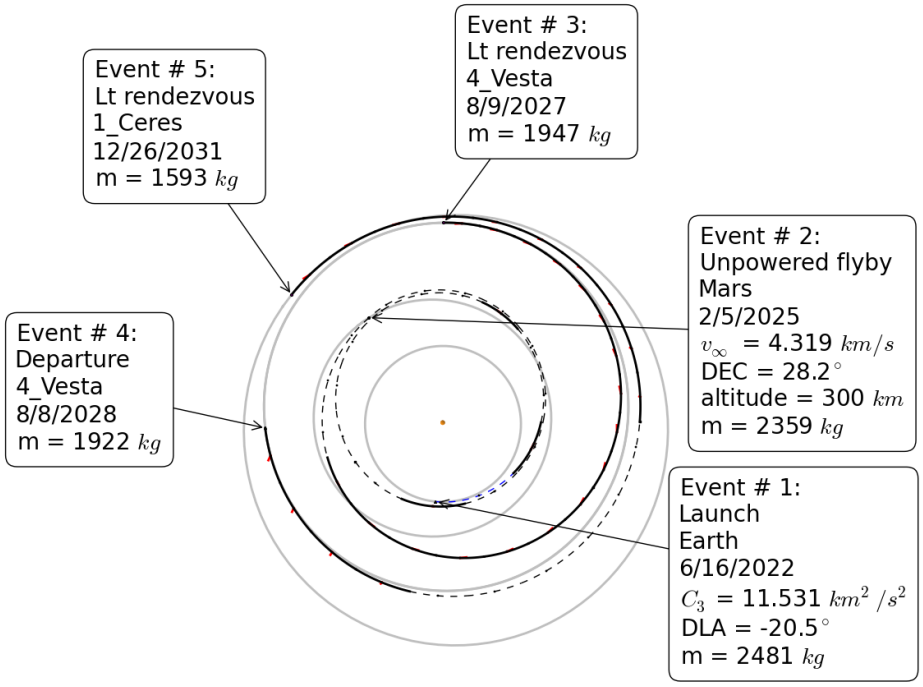

Figure 4. Optimal trajectory for the main-belt rendezvous mission

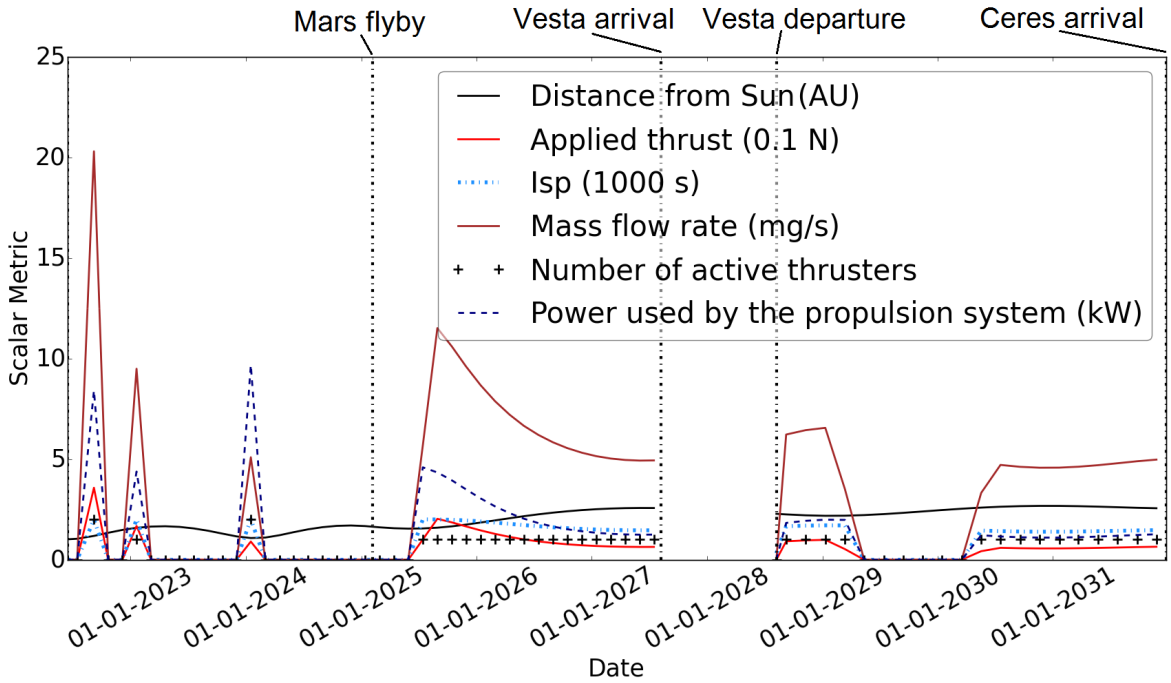

Figure 5. Time-history of power and propulsion for the main-belt rendezvous mission

C. Low-Thrust Mission to Pluto

The last example is a mission to Pluto, inspired by the recent success of New Horizons [49]. Unlike New Horizons, which was a fast flyby, the notional mission presented here would rendezvous with the Pluto system, enabling in-depth science. The spacecraft for this mission is given a 1 kW radioisotope power source that decays at a rate of 2% per year and a VSI propulsion system capable of operating at any I_{sp} between 1000 s and 3000 s with an efficiency of 60%. The spacecraft bus requires 200 W at all times, and the remaining power is available for the propulsion system. Like New Horizons, the spacecraft launches on an Atlas V 551 with a Star 48 upper stage. Once again the standard preliminary design margins are applied as per Oh *et al.* [47] - 15% power margin, 10% propellant margin, and a 90% thruster duty cycle. No propellant tank constraint was applied. The objective is to maximize delivered mass for a flight time of up to 25 years. The outer-loop is allowed to choose up to four flybys of Venus, Earth, Mars, Jupiter, Saturn, or Uranus. Table 11 lists the assumptions for the Pluto rendezvous mission.

The Pluto rendezvous problem was run 10 times on a 60-core Intel Xeon E7-4890 running at 2.8 GHz with 500 GB of RAM. Each run took 67 hours, although the optimal solution was often identified in less than half of that time. The algorithm continued to run because it had no way to know that it had found the global optimum. In every case the HOCP automaton converged to an Earth-Jupiter-Pluto flyby sequence with a delivered mass of 983 kg. The mean number of generations to find the optimal solution was 99 and the median was 111. The greater reliability of the HOCP automaton on the Pluto rendezvous problem relative to the previous examples is partly because the optimal sequence is quite short and therefore generates a smaller inner-loop problem and partly because MBH was run for a longer period of time. Table 12 shows all 19 solutions for one of the runs that were "feasible" in the sense that the inner-loop returned a solution. However a negative mass value is listed for several of the missions, meaning that while the continuity constraints were satisfied for the trajectory itself, it was not possible to fit the 10% propellant margin into the spacecraft. In these cases mathematical feasibility is not the same thing as physical feasibility.

Table 13 lists the itinerary of the optimal Pluto rendezvous mission. Figure 6 shows the trajectory followed by the optimal mission, and Fig. 7 shows the time history of power and propulsion parameters over the course of the mission. Note that in Fig. 7 the optimizer chooses the I_{sp}, denoted by the dash-dotted line, to be close to the lower bound of 1000 seconds at the beginning and end of the mission, but prefers a higher I_{sp} and therefore a lower thrust for the middle portion. An optimal coast arc is found, beginning one year after the Jupiter flyby and lasting for three years. Note also that in Table 13 and Fig. 6, the mass at arrival is 1064 kg. However when the 10% propellant margin is applied only 983 kg is available for dry mass, consistent with Table 12.

Table 11. Assumptions for the Pluto Rendezvous Mission

Option	Value
Launch window open date	1/1/2025
Launch window close date	1/1/2035
Flight time upper bound	25 years
Arrival condition	rendezvous (match position and velocity)
Launch vehicle	Atlas V 551 with Star 48 upper stage
Launch asymptote declination bounds	$[-28.5, 28.5]$ (Kennedy Space Center)
Power supply	1 kW radioisotope with 2% decay per year
Propulsion system	VSI with 60% efficiency and I_{sp} in $[1000, 3000]$ s
Duty cycle	90%
Power margin	15%
Number of time steps per phase	20
Flyby sequence	up to four flybys of Venus, Earth, Mars, Jupiter, Saturn, or Uranus
Outer-loop GA Population Size	59
Outer-loop GA number of generations	200
Outer-loop GA CR_{GA}	0.3
Outer-loop GA μ_{GA}	0.15
Inner-loop MBH run time	20 minutes
Inner-loop MBH Pareto α	1.2
Inner-loop MBH maximum iterations	100,000 (never reached)

Table 12. Convergence history for one run of the Pluto rendezvous problem

Sequence	Generation found	Delivered mass (kg)
EJP	82	984
EP	92	601
EMP	199	329
EJJP	146	322
ESP	187	169
EJSP	166	158
EEP	3	31
EVP	4	-20
EJUP	134	-75
EVEP	57	-76
EJSJP	28	-80
ESSP	2	-91
EJJUP	67	-103
EJSSP	13	-124
EUP	43	-144
EEMP	3	-203
EEEP	8	-244
EJMP	98	-282
EEJP	110	-588

Table 13. Itinerary for the optimal Pluto rendezvous mission

Date	Event	Location	C_3 (km^2/s^2)	Flyby altitude	Mass (kg)
12/15/2028	launch	Earth	76.9		1870
10/6/2030	flyby	Jupiter	88.7	23.9 R_j	1703
12/13/2053	rendezvous	Pluto			1064

VI. Conclusion

A. Summary

In this work we show that the low-thrust interplanetary mission design problem may be posed as a hybrid optimal control problem (HOCP) and effectively explored via the powerful combination of a discrete genetic algorithm (GA) outer-loop with a monotonic basin hopping (MBH)+nonlinear programming (NLP) inner-loop. The trade space for a given mission may be characterized even

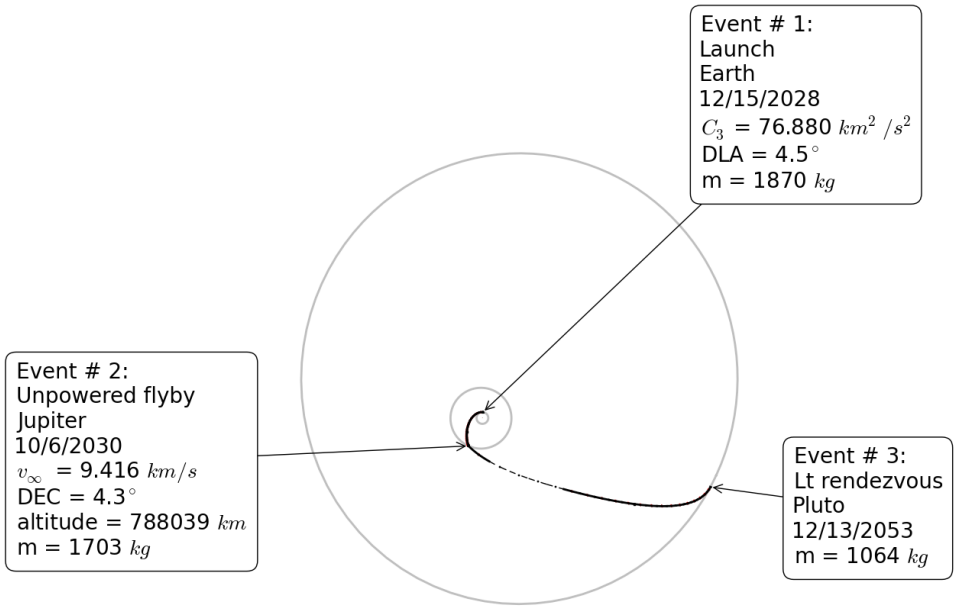

Figure 6. Optimal trajectory for the Pluto rendezvous mission

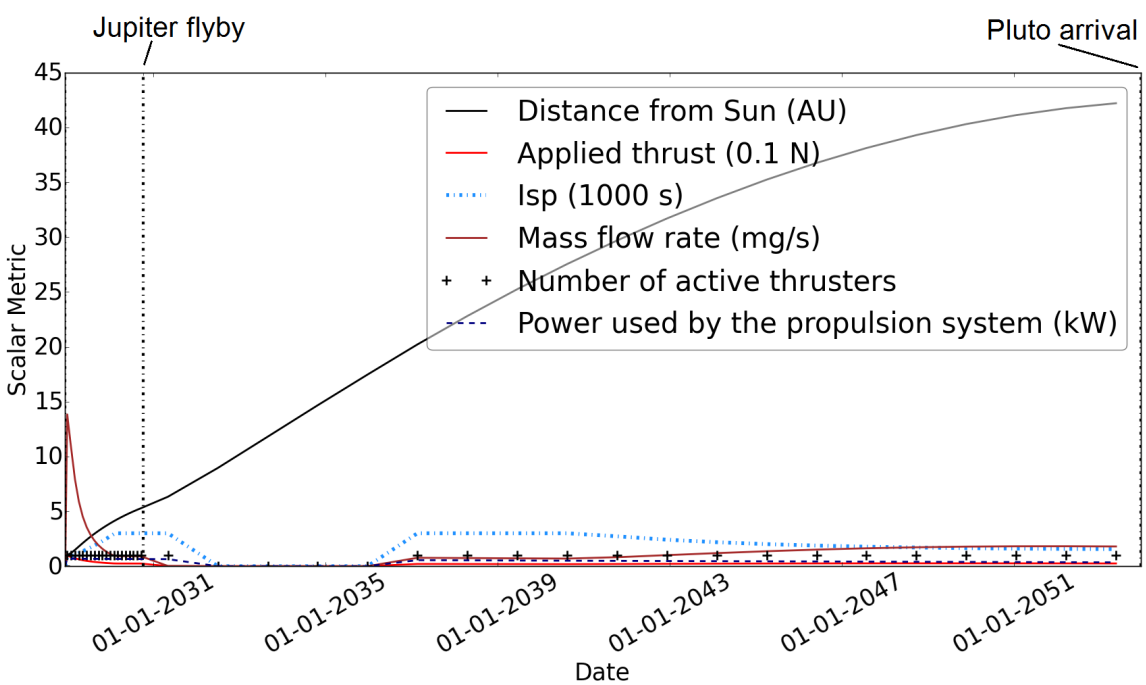

Figure 7. Time-history of power and propulsion for the Pluto rendezvous mission

when there is only enough time to evaluate a fraction of the full combinatorial design problem.

The algorithm described in this work is a fully automated technique that can design interplanetary trajectories with low-thrust propulsion and complex flyby sequences while not requiring any *a priori* information about the optimal sequence or trajectory. This process is far less expensive, in terms of human hours required, than traditional design processes in which a human analyst constructs the trajectory by hand. This can allow a single analyst to simultaneously work on several very different problems. In addition, the solutions found by the HOCP automaton are sometimes non-intuitive and might not be considered in an intuition-driven design process. For example, some low-thrust trajectory problems can be readily solved with an initial guess from a ballistic design and some cannot. The ability to find non-intuitive solutions is is a significant advantage in mission design, as the success or failure of a planetary mission proposal sometimes hinges on the flight dynamics team finding a non-intuitive trajectory design that enables a larger payload, a shorter flight time, or in some cases both.

However the quest for a robust, fully automated technique is ongoing. As demonstrated in the second example in this work, the HOCP automaton described here is not yet sufficiently reliable on some types of problems such as multi-rendezvous missions in which the outer-loop must select the targets. This will be the subject of future work.

generate P, a set of N_p random vectors $\mathbf{x}_{GA}$ in the decision space
evaluate all $\mathbf{x}_i \in P$
while not hit stop criterion **do**
  for i = 1 to $CR_{GA}N_p$ **do**
    randomly choose $N_{parents}$ individuals from the population, forming a parent pool
    choose the best member of the parent pool as $\mathbf{x_{mom}}$
    randomly choose $N_{parents}$ individuals from the population, forming a new parent pool
    choose the best member of the parent pool as $\mathbf{x_{dad}}$
    choose random integers r_1 and r_2 in $[0, len\,(\mathbf{x})]$
    form $\mathbf{x_{child}}$ by taking the first r_1 entries and last $(len\,(\mathbf{x}) - r_2)$ entries from $\mathbf{x_{mom}}$ and the middle entries from $\mathbf{x_{dad}}$
    insert $\mathbf{x_{child}}$ into the new population
  end for
  for i = 1 to $(1 - CR_{GA})\,N_p - N_{elite}$ **do**
    randomly choose $N_{parents}$ individuals from the population, forming a pool
    choose the best member of the pool as $\mathbf{x_{retained}}$
    form $\mathbf{x_{mutated}}$ by copying $\mathbf{x_{retained}}$ and then with probability μ_{GA} replacing each entry with a random value
    insert $\mathbf{x_{mutated}}$ into the new population
  end for
  for i = 1 to N_{elite} **do**
    insert the i-th best unique individual $\mathbf{x_{elite-i}}$, unmodified, into the new population
  end for
  evaluate all $\mathbf{x}_i \in P$
end while
return $\mathbf{x}_{best}$

Figure 8. Genetic Algorithm (GA)

```
generate random point x
run NLP solver to find point x* using initial guess x
x_c = x*
if x* is a feasible point then
    save x* to archive
end if
while not hit stop criterion do
    generate x' by randomly perturbing x_c
    for each time of flight variable t_i in x' do
        if rand(0,1) < ρ_{time-hop} then
            shift t_i forward or backward one synodic period
        end if
    end for
    run NLP solver to find point x* from x'
    if x* is feasible and f(x*) < f(x_c) then
        x_c = x*
        save x* to archive
    else if x* is infeasible and x_c is infeasible and ||c(x*)|| < ||c(x_c)||)
        x_c = x*
    end if
end while
return best x* in archive
```

Figure 9. Monotonic Basin Hopping (MBH)

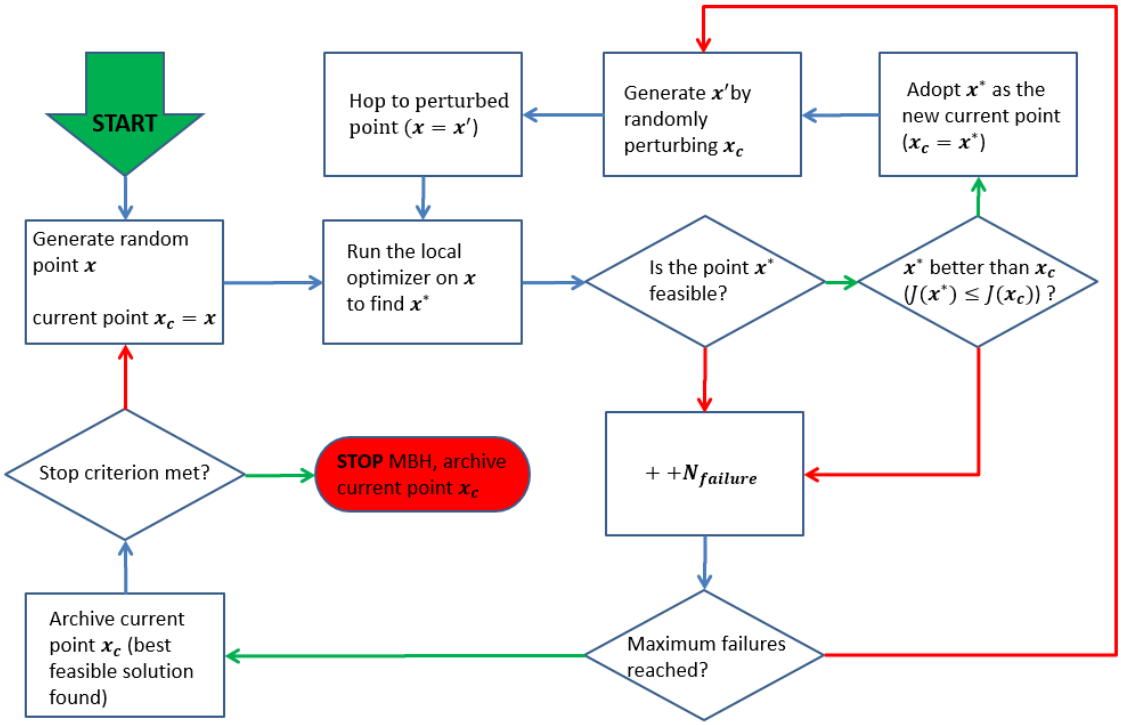

Figure 10. Monotonic Basin Hopping (MBH) flow chart

Acknowledgments

The authors would like to acknowledge Matthew Vavrina at a.i. solutions, Donald Ellison, Dr. Alexander Ghosh, and Ryne Beeson at the University of Illinois, and Jeremy Knittel at NASA Goddard Space Flight Center for their ongoing contributions to EMTG. Frank Vaughn, Wayne Yu, and Greg Marr of NASA Goddard Space Flight Center and Chelsea Welch of Lockheed-Martin assisted with testing. This research was funded by the NASA Graduate Student Researchers Program (GSRP), the NASA Goddard Space Flight Center Internal Research and Development (IRAD) program, and several customer-funded studies at NASA Goddard Space Flight Center.

References

[1] Longuski, J. M. and Williams, S. N., "Automated design of gravity-assist trajectories to Mars and the outer planets," *Celestial Mechanics and Dynamical Astronomy*, Vol. 52, No. 3, 1991, pp. 207–220, doi: 10.1007/BF00048484.

[2] Petropoulos, A. E. and Longuski, J. M., "Shape-Based Algorithm for the Automated Design of Low-Thrust, Gravity Assist Trajectories," *Journal of Spacecraft and Rockets*, Vol. 41, No. 5, 2004, pp. 787–796, doi: 10.2514/1.13095.

[3] Wall, B. J. and Conway, B. A., "Shape-Based Approach to Low-Thrust Rendezvous Trajectory Design," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 95–101, doi: 10.2514/1.36848.

[4] Wall, B. and Novak, D., "A 3D Shape-Based Approximation Method for Low-Thrust Trajectory Design," *Advances in the Astronautical Sciences*, Vol. 142, 2011, pp. 1163–1176.

[5] Downing, J., "Pyxis Tool," Tech. Rep. NASA/TM-2006-214139, Flight Dynamics Branch (FDAB), NASA Goddard Space Flight Center, 2006.

[6] Sims, J., Finlayson, P., Rinderle, E., Vavrina, M., and Kowalkowski, T., "Implementation of a Low-Thrust Trajectory Optimization Algorithm for Preliminary Design," in "AIAA/AAS Astrodynamics Specialist Conference and Exhibit," AIAA Paper 2006-6746, 2006, doi: 10.2514/6.2006-6746.

[7] McConaghy, T. T., *GALLOP Version 4.5 User's Guide*, School of Aeronautics and Astronautics, Purdue University, 2005.

[8] Buss, M., Glocker, M., Hardt, M., von Stryk, O., Bulirsch, R., and Schmidt, G., "Nonlinear Hybrid Dynamical Systems: Modeling, Optimal Control, and Applications," in "Modelling, Analysis, and Design of Hybrid Systems," Springer Science+ Business Media, pp. 311–335, 2002, doi: 10.1007/3-540-45426-8_18.

[9] Ross, I. M. and D'Souza, C. N., "Hybrid Optimal Control Framework for Mission Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp. 686–697, doi: 10.2514/1.8285.

[10] Conway, B. A., Chilan, C. M., and Wall, B. J., "Evolutionary principles applied to mission planning problems," *Celestial Mechanics and Dynamical Astronomy*, Vol. 97, No. 2, 2006, pp. 73–86, doi: 10.1007/s10569-006-9052-7.

[11] Englander, J. A., Conway, B. A., and Williams, T., "Automated Mission Planning via Evolutionary Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 6, 2012, pp. 1878–1887, doi: 10.2514/1.54101.

[12] Gad, A. and Abdelkhalik, O., "Hidden Genes Genetic Algorithm for Multi-Gravity-Assist Trajectories Optimization," *Journal of Spacecraft and Rockets*, Vol. 48, No. 4, 2011, pp. 629–641, doi: 10.2514/1.52642.

[13] Gad, O. A. A., "Dynamic-Size Multiple Populations Genetic Algorithm for Multigravity-Assist Trajectories Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 2, 2012, pp. 520–529, doi: 10.2514/1.54330.

[14] Vasile, M. and Campagnola, S., "Design of low-thrust multi-gravity assist trajectories to Europa," *Journal of the British Interplanetary Society*, Vol. 62, No. 1, 2009, pp. 15–31.

[15] Sims, J. A. and Flanagan, S. N., "Preliminary Design of Low-Thrust Interplanetary Missions," in "AAS/AIAA Astrodynamics Specialist Conference," Paper AAS 99-338, Girdwood, Alaska, 1999.

[16] Vasile, M., Minisci, E., and Locatelli, M., "Analysis of Some Global Optimization Algorithms for Space Trajectory Design," *Journal of Spacecraft and Rockets*, Vol. 47, No. 2, 2010, pp. 334–344, doi: 10.2514/1.45742.

[17] Yam, C. H., Lorenzo, D. D., and Izzo, D., "Low-thrust trajectory design as a constrained global optimization problem," SAGE Publications, Vol. 225, 2011, pp. 1243–1251, doi: 10.1177/0954410011401686.

[18] Englander, J. A., *Automated Trajectory Planning for Multiple-Flyby Interplanetary Missions*, Ph.D. thesis, University of Illinois at Urbana-Champaign, 2013.

[19] Englander, J. A. and Englander, A. C., "Tuning Monotonic Basin Hopping: Improving the Efficiency of Stochastic Search as Applied to Low-Thrust Trajectory Optimization," in "24th International Symposium on Space Flight Dynamics, Laurel, MD," ISSFD, 2014.

[20] "EMTG (Evolutionary Mission Trajectory Generator)," https://sourceforge.net/projects/emtg/, accessed 6/26/2016.

[21] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM J. Optim.*, Vol. 12, No. 4, 2002, pp. 979–1006, doi: 10.1137/s1052623499350013.

[22] "PaGMO (Parallel Global Multiobjective Optimizer)," https://github.com/esa/pagmo, accessed 6/26/2016.

[23] Vinkó, T. and Izzo, D., "Global Optimisation Heuristics and Test Problems for Preliminary Spacecraft Trajectory Design," Tech. Rep. GOHTPPSTD, European Space Agency, the Advanced Concepts Team, 2008.

[24] Conway, B. A., "An Improved Method due to Laguerre for the Solution of Kepler's Equation," *Celestial Mechanics*, Vol. 39, No. 2, 1986, pp. 199–211, doi: 10.1007/bf01230852.

[25] "SPICE Ephemeris," http://naif.jpl.nasa.gov/naif/, accessed 6/26/2016.

[26] Arora, N. and Russell, R. P., "A fast, accurate, and smooth planetary ephemeris retrieval system," *Celest Mech Dyn Astr*, Vol. 108, No. 2, 2010, pp. 107–124, doi: 10.1007/s10569-010-9296-0.

[27] "NASA Launch Services Program Launch Vehicle Performance Web Site," http://elvperf.ksc.nasa.gov/elvMap/, accessed 6/26/2016.

[28] Vavrina, M. and Howell, K., "Global Low-Thrust Trajectory Optimization Through Hybridization of a Genetic Algorithm and a Direct Method," in "AIAA/AAS Astrodynamics Specialist Conference and Exhibit," AIAA, 2008, doi: 10.2514/6.2008-6614.

[29] Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[30] Chilan, C. M. and Conway, B. A., "A space mission automaton using hybrid optimal control," *Advances in the Astronautical Sciences*, Vol. 127, 2007, pp. 259–276.

[31] "MPI: A Message-Passing Interface Standard," Tech. rep., Message Passing Forum, Knoxville, TN, USA, 1994.

[32] Rauwolf, G. A. and Coverstone-Carroll, V. L., "Near-optimal low-thrust orbit transfers generated by a genetic algorithm," *Journal of Spacecraft and Rockets*, Vol. 33, No. 6, 1996, pp. 859–862, doi: 10.2514/3.26850.

[33] Coverstone-Carroll, V., "Near-Optimal Low-Thrust Trajectories via Micro-Genetic Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 196–198, doi: 10.2514/2.4020.

[34] Coverstone-Carroll, V., Hartmann, J., and Mason, W., "Optimal multi-objective low-thrust spacecraft trajectories," *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, No. 2-4, 2000, pp. 387–402, doi: 10.1016/s0045-7825(99)00393-x.

[35] Englander, J. A., Conway, B. A., and Williams, T., "Automated Interplanetary Mission Planning," in "AAS/AIAA Astrodynamics Specialist Conference, Minneapolis, MN," AIAA paper 2012-4517, 2012, doi: 10.2514/6.2012-4517.

[36] Ellison, D. H., Englander, J. A., and Conway, B. A., "Robust Global Optimzation of Low-Thrust, Multiple-Flyby Trajectories," in "AAS/AIAA Astrodynamics Specialist Conference, Hilton Head, SC," AAS paper 13-924, 2013.

[37] Englander, J. A., Ellison, D. H., and Conway, B. A., "Global Optimization of Low-Thrust, Multiple-Flyby Trajectories at Medium and Medium-High Fidelity," in "AAS/AIAA Space-Flight Mechanics Meeting, Santa Fe, NM," AAS, 2014.

[38] Ellison, D. H., Englander, J. A., Ozimek, M. T., and Conway, B. A., "Analytical Partial Derivative Calculation of the Sims-Flanagan Transcription Match Point Constraints," in "AAS/AIAA Space-Flight Mechanics Meeting, Santa Fe, NM," AAS, 2014.

[39] Leary, R. H., "Global Optimization on Funneling Landscapes," *Journal of Global Optimization*, Vol. 18, No. 4, 2000, pp. 367–383, doi: 10.1023/A:1026500301312.

[40] Addis, B., Cassioli, A., Locatelli, M., and Schoen, F., "A global optimization method for the design of space trajectories," *Computational Optimization and Applications*, Vol. 48, No. 3, 2009, pp. 635–652, doi: 10.1007/s10589-009-9261-6.

[41] Englander, J., Vavrina, M. A., Naasz, B. J., Merrill, R. G., and Qu, M., "Mars, Phobos, and Deimos Sample Return Enabled by ARRM Alternative Trade Study Spacecraft," in "AIAA/AAS Astrodynamics Specialist Conference," AIAA paper 2014-4354, 2014, doi: 10.2514/6.2014-4354.

[42] Cassioli, A., Izzo, D., Lorenzo, D. D., Locatelli, M., and Schoen, F., "Global Optimization Approaches for Optimal Trajectory Planning," in "Springer Optimization and Its Applications," Springer Science+ Business Media, pp. 111–140, 2012, doi: 10.1007/978-1-4614-4469-5_5.

[43] Yarnoz, D. G., Jehn, R., and Croon, M., "Interplanetary navigation along the low-thrust trajectory of BepiColombo," *Acta Astronautica*, Vol. 59, No. 1-5, 2006, pp. 284–293, doi: 10.1016/j.actaastro.2006.02.028.

[44] Katzkowski, M., Corral, C., Jehn, R., Pellon, J.-L., Khan, M., Yanez, A., and Biesbroek, R., "BepiColombo Mercury Cornerstone Mission Analysis: Input to Definition Study," Tech. rep., 2002.

[45] "JPL Small-Body Database Search Engine," http://ssd.jpl.nasa.gov/sbdb_query.cgi, accessed 6/26/2016.

[46] Hofer, R., "High-Specific Impulse Operation of the BPT-4000 Hall Thruster for NASA Science Missions," in "46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit," AIAA paper 2010-6623, 2010, doi: 10.2514/6.2010-6623.

[47] Oh, D. Y., Snyder, J. S., Goebel, D. M., Hofer, R. R., and Randolph, T. M., "Solar Electric Propulsion for Discovery-Class Missions," American Institute of Aeronautics and Astronautics (AIAA), Vol. 51, 2014, pp. 1822–1835, doi: 10.2514/1.a32889.

[48] Vavrina, M., Englander, J., and Ghosh, A., "Coupled Low-Thrust Trajectory and Systems Optimization Via Multi-objective Hybrid Optimal Control," in "AAS/AIAA Space Flight Mechanics Meeting," AAS paper 15-397, 2015.

[49] Guo, Y. and Farquhar, R. W., "New Horizons Mission Design," in "New Horizons," Springer Science+ Business Media, pp. 49–74, doi: 10.1007/978-0-387-89518-5_4.