AAS 16-272

GLOBAL OPTIMIZATION OF *N*-MANEUVER, HIGH-THRUST TRAJECTORIES USING DIRECT MULTIPLE SHOOTING

Matthew A. Vavrina*, Jacob A. Englander†, and Donald H. Ellison‡

The performance of impulsive, gravity-assist trajectories often improves with the inclusion of one or more maneuvers between flybys. However, grid-based scans over the entire design space can become computationally intractable for even one deep-space maneuver, and few global search routines are capable of an arbitrary number of maneuvers. To address this difficulty a trajectory transcription allowing for any number of maneuvers is developed within a multi-objective, global optimization framework for constrained, multiple gravity-assist trajectories. The formulation exploits a robust shooting scheme and analytic derivatives for computational efficiency. The approach is applied to several complex, interplanetary problems, achieving notable performance without a user-supplied initial guess.

INTRODUCTION

Deep-space maneuvers (DSM), or interior impulsive maneuvers, frequently improve performance of chemical propulsion based missions, reducing propellant cost or time of flight (TOF).[1,2,3,4] Moreover, the performance can sometimes be improved further with more than one DSM.[5,6] Mid-course maneuvers are particularly useful in reducing the ΔV cost, or even enabling a mission to close, for tightly constrained transfers such as those with a strict planetary entry requirement or a restrictive TOF. Additionally, non-traditional objective functions may dictate multiple DSMs on a single transfer leg. Identifying the optimal maneuvers is not trivial, however, as the design space is frequently sensitive to small changes in optimization parameters. While the benefits of multiple gravity assists are well documented,[7] designing a globally-optimal solution with gravity assists is difficult as the design space can grow significantly and sensitivities become more acute with each additional flyby and maneuver. Incorporating trajectory constraints to accommodate mission requirements on thermal, communication, and terminal conditions into the problem further complicates the optimization process, but ensures that solutions that are identified are indeed viable for a mission. In turn, generating an initial guess that is sufficiently close to the optimal solution frequently requires hard-earned expertise with the problem at hand.

A variety of approaches have been devised to address particular aspects of the outlined challenges. Lawden's foundational development of the primer vector in 1963 allows for the optimization of multiple mid-course maneuvers.[1] In the late 1960's, Lion and Handelsman[8] as well as Jezewski and Rozendaal[9] built upon Lawden's work, using primer vector theory to determine optimal transfers with interior impulses from a given sub-optimal trajectory. These ideas have been effectively translated to multiple gravity-assist trajectories with tools such as MIDAS by Sauer[10] and a multi-DSM, gravity-assist technique by Olympio.[11] Direct

* Senior Systems Engineer, a.i. solutions, Inc., 500 Forbes Boulevard Suite 300. Lanham, MD 20706, Lanham, MD, 20706, USA., Member AAS
† Aerospace Engineer, Navigation and Mission Design Branch, NASA Goddard Space Flight Center, Greenbelt, MD, 20771, USA.
‡ Ph. D. Candidate, Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, 104 South Wright Street, Mail Code-236, Member AAS, AIAA

optimization schemes using multiple-shooting formulations have also been successfully developed and widely applied in software such as CATO and MALTO.[12,13,14]

To identify trajectory parameters for an initial guess within an optimization framework a broad scan for ballistic trajectories using a grid-based search tool is often the first step in the design of multiple gravity-assist trajectories. While a traditional pork-chop plot or grid-based scan (e.g., STOUR[15]) indicates ideal ballistic transfer opportunities for use as an initial guess, a DSM can strongly alter mission performance. Accordingly, Patel and Longuski developed a grid-based approach with the ability to automatically include a single impulsive maneuver at a strategic location of the trajectory.[16] Lantukh and Russell also developed an approach for the automated design of transfers with leveraging maneuvers.[17] These strategies are highly advantageous for certain maneuver types (e.g., v_∞-leveraging or a broken-plane maneuver), but an arbitrary number of DSMs of any variety does not appear to be computationally tractable for large date bounds or many-phase missions. A solution from the grid-search is then typically used as an initial guess within a medium-fidelity patched conics optimization tool or ported directly to a high-fidelity optimizer. Following this design procedure, the refinement step is constrained to the local neighborhood of the initial guess.

Alternatively, global optimization strategies have received much recent attention, but identifying an arbitrary number of globally optimal maneuvers is challenging given the sensitivity to the number of maneuvers, the maneuver times, magnitude, and direction. Metaheuristic-based routines (e.g., evolutionary algorithms, monotonic basin hopping, and simulated annealing) have been of particular interest for global trajectory optimization because a user-defined initial guess is frequently not required to start the optimization process. Gage et al. developed one of the first stochastic global search techniques for trajectory optimization without a DSM using a genetic algorithm.[18] Researchers followed with an array of different strategies to globally optimize the multiple gravity assist one deep-space maneuver (MGADSM) problem.[19,20,21,22] A direct formulation that is frequently incorporated in several global optimization techniques is using an optimizer to vary the initial maneuver vector and epoch as well as the final rendezvous epoch, and then solving the resulting Lambert problem from the DSM date to rendezvous epoch.[19] This approach guarantees a feasible solution for an unconstrained problem and is the original MGADSM transcription in the Evolutionary Mission Trajectory Generator (EMTG) software frequently employed by NASA for preliminary design of chemical and electrical propulsion based missions.[23,24]

In this work a multiple-shooting trajectory transcription is developed, allowing for any number of impulsive maneuvers on a single leg or phase for multiple gravity assist trajectories (the MGA*n*DSM problem). The trajectory formulation is a direct method in which the optimal control problem is parameterized in a shooting framework with similarities to the Sims-Flanagan transcription[25] frequently used for low-thrust trajectory optimization and the Byrnes-Bright transcription[12] in CATO. The nonlinear programming (NLP) problem resulting from the formulation is solved using a global-local hybrid algorithm in which a gradient-based local optimizer, sequential quadratic programming (SQP) in this work, is combined with a global optimization routine, namely monotonic basin hopping (MBH). The hybrid formulation does not require an initial guess as the MBH routine stochastically explores the design space, selecting trajectory parameters for refinement in the inner loop. Additionally, the approach is amenable to wide-ranging mission constraints, and analytic derivatives are provided for a substantial portion of the Jacobian, vastly improving efficiency. The approach is applied to example problems, the Cassini[26] and OSIRIS-REx[27] interplanetary trajectories, searching globally for single-objective optimal solutions. Performance is then compared to a Lambert-based transcription.

While single-objective, globally-optimal solutions are valuable in and of themselves, rarely is there only one objective that is the sole interest of a mission. Ideally, a mission designer would be able to consider multiple objectives, and then select from among the set of globally optimal solutions with a full understanding of the tradeoffs between the objectives. Two common objectives for most interplanetary missions are the minimization of flight time and the maximization of delivered mass to a target. Recent work on solving multi-objective hybrid optimal control problems (MOHOCP) with discrete and continuous variables for low-thrust trajectories[28,29,30,31] is extended to the MGA*n*DSM transcription developed in this work. The non-dominated sorting genetic algorithm II (NSGA-II)[32] is used as a wrapper around the MBH+NLP loop to simultaneously solve for multiple objectives in a single run. The NSGA-II outer loop is capable of globally searching over different gravity assist bodies using the null gene technique developed by Englander.[33] The

resulting MOHOCP technique is applied to a large MGA*n*DSM problem for an Earth to Jupiter mission with up to five variable gravity assist bodies and up to two DSMs between bodies, generating a representation of the Pareto front of optimal solutions.

MISSION MODELING

The mission model for solving the MGA*n*DSM problem follows from the base EMTG architecture described in Reference 24. The structure enables an optimizer to search over a variable number of flyby bodies and different destinations in one run. The distinguishing difference is the incorporation of the MGA*n*DSM two-point shooting transcription formulated in this section.

EMTG Mission Architecture

Missions in EMTG are structured using three levels of event types. As depicted in Figure 1, at the top of the structure is the mission level composed of all event types throughout the mission. These event types include departures, arrivals, DSMs, and flybys. One or more journeys comprise a mission, and define the trajectory in terms of the set of events at the required target bodies of the mission, namely the starting, ending, and any required intermediate bodies. A journey's boundaries are locations in which the spacecraft will execute user-defined specific events, and can be constrained in a number of ways. Within a journey are one or more phases, which are similar to journeys in that they start and end at bodies. However, phases may start and end at bodies other than the required targets defined at the journey level. In particular, phases are utilized to incorporate flyby bodies that are used strictly to modify the trajectory with the goal of improving mission performance versus an encounter that is required to meet mission objectives. As an example, the OSIRIS-Rex mission to return a sample of the asteroid Bennu would be composed of two journeys: an outbound Earth to Bennu journey and a return Bennu to Earth journey. However, the outbound journey incorporates an Earth flyby to decrease total ΔV, and two phases comprise that journey. This mission structuring enables an optimizer to vary the number of phases as well as the flyby bodies within a journey. This three-level architecture is used in all of the transcriptions within EMTG.

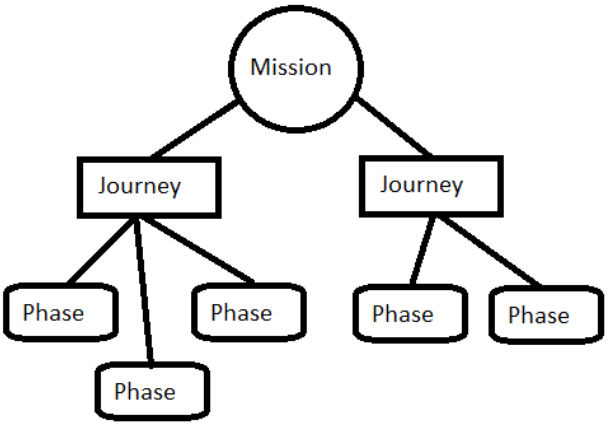

Figure 1. EMTG mission Structure

MGA*n*DSM*s* Trajectory Transcription

The direct method developed in this work employs a multiple shooting transcription within a patched-conic modeling framework. The current implementation is focused on rapid, medium fidelity trajectory generation, but patched conics are not required should a higher-fidelity application be desired. A match point is incorporated in each trajectory phase (between each body) with forward/backward shooting to enable reduced sensitivity to initial guess errors. This strategy avoids unmitigated error growth that can occur from small initial errors in a forward-only shooting scheme. Additionally, in this way, trajectories are preconditioned with the intermediate (flybys) and terminal body encounters imposed on the underlying trajectory structure

regardless of the quality of the initial guess. Encounter states can be parameterized to ensure flybys meet specific conditions provided user-defined design variable bounds.

This trajectory phase option is referred to as MGA*n*DSM*s* in EMTG parlance for multiple gravity assists (MGA) with *n* DSMs per phase, using a shooting technique, *s*. A diagram of the trajectory transcription is illustrated in Figure 2, where MP designates the match point. The diagram outlines the trajectory structure for a single phase in a mission. The endpoints represent the beginning and end of the phase in mission time (left to right in figure). End points are typically planetary bodies or another spacecraft, but can also be an arbitrary state specified by the user. Impulsive maneuvers, or DSMs, are represented by arrows, and instantaneously change the spacecraft's velocity. The maneuvers are separated in time by a Δt optimization variable. Note that maneuvers at the phase end points are also possible. Alternatively, the phase endpoint may be a gravity-assist body. The number of maneuvers per phase may be specified by the user or an outer-loop optimizer. If fewer than *n* maneuvers are optimal for the transfer the formulation is structured so that one or more of the potential maneuvers will have a magnitude of zero.

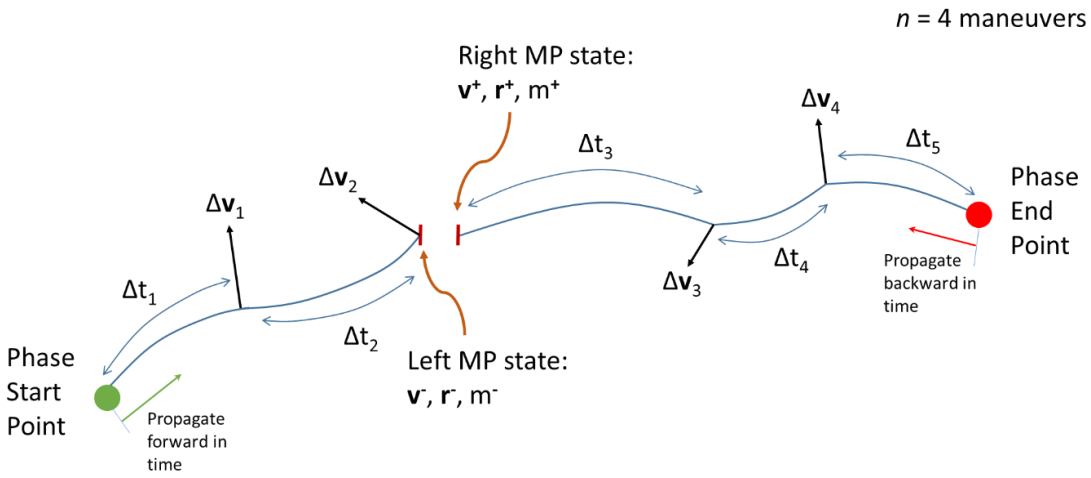

Figure 2. MGA*n*DSM*s* transcription for *n* = 4, depicting the two-point shooting scheme to match point for a trajectory phase.

The trajectory is propagated forward in time, nominally using a Kepler propagation between each maneuver point, from the phase start point until the match-point DSM is reached. At each maneuver point the three components comprising the impulsive maneuver instantaneously change the spacecraft velocity. Similarly, the trajectory is propagated backwards in time from the phase end point to the match point time, stopping at each maneuver along the way to modify the spacecraft velocity. The match point maneuver is defined by the number of maneuvers in the phase, where the match point maneuver will be at the $n/2$ maneuver if the number of maneuvers is even or at the middle maneuver if the number of maneuvers is odd. The match point maneuver always resides on the forward propagation side of the match point, and varies in time according to the Δt variables of the phase. In general, a discontinuity in position, velocity, mass, and time will exist at the match point from the forward and backward shooting.

Ensuring that the position, velocity, mass are continuous (within a tolerance) at the match point are the critical nonlinear constraints of the formulation. That is, the optimizer must eliminate the match point errors to achieve a feasible trajectory. Time continuity is enforced via a linear constraint that the sum of the Δt variables equals the phase time of flight. Analytic derivatives are provided for all continuity constraints and are critical to the transcription's robustness.

The core optimization variables comprising this phase transcription are outline in Table 1. These optimization variables comprise the phase decision variable vector **p** that is later used to describe the partial derivatives for the Jacobian. Other important match point state parameters that are not optimization variables, but

are instead a function of the optimization parameters are detailed in Table 2. Mass is included in the propagation to allow for mass constraints outside of the match points or when mass is involved in the objective function. It is particularly useful to consider mass when missions begin with a launch vehicle and the objective is to maximize the delivered mass at a destination versus minimizing total ΔV. In these circumstances total ΔV is not an accurate representation of mass as the launch vehicle I_{sp} is typically different from the spacecraft I_{sp}.

Table 1. MGA*n*DSM*s* Core Phase Optimization Variables

Variable	Description
Δt_{phase}	phase time of flight
Δt_1	time between phase start point and first maneuver
$\Delta t_2, \ldots \Delta t_n$	relative time between maneuvers, where n is the total number of maneuvers in the phase
Δt_{n+1}	time between last maneuver of phase and phase end point
$\mathbf{\Delta v}_1, \mathbf{\Delta v}_2, \ldots, \mathbf{\Delta v}_n$	velocity change vectors for maneuvers one through n
$\mathbf{v}_{\text{initial}}$	initial velocity vector relative to the central body at the phase start node
$\mathbf{v}_{\text{final}}$	final velocity vector relative to the central body at the phase end node.

Table 2: MGA*n*DSM*s* Match Point State Parameters

State Parameter	Description
r_x^-, r_y^-, r_z^-	Cartesian position state at forward propagation side of the match point
r_x^+, r_y^+, r_z^+	Cartesian position state at backward propagation side of the match point
v_x^-, v_y^-, v_z^-	Cartesian velocity state at forward propagation side of the match point
v_x^+, v_y^+, v_z^+	Cartesian velocity state at backward propagation side of the match point
m^-	mass state at forward propagation side of the match point
m^+	mass state at backward propagation side the of match point

The match point continuity constraints are summarized in vector form as :

$$\mathbf{c}_{mp} = \left[r_x^+ - r_x^-,\ r_y^+ - r_y^-,\ r_z^+ - r_z^-,\ v_x^+ - v_x^-,\ v_y^+ - v_y^-,\ v_z^+ - v_z^-,\ m^+ - m^-\right]^T = \boldsymbol{\varepsilon}^T, \tag{1}$$

where $\boldsymbol{\varepsilon}$ is a vector of near-zero tolerances. These constraints ensure a feasible trajectory with no state discontinuities across the match point. Another key phase constraint for the transcription maintains that the summation of the time between the maneuvers in addition to the time from phase start to the first maneuver and the time from the last maneuver to the phase end point does not exceed the phase time of flight:

$$\sum_{n=1}^{n+1} \Delta t_n = t_{phase}, \tag{2}$$

In implementation the Δt variables are normalized by t_{phase} such that the sum is equal to one. User-defined distance constraints to any body are also available, and are enforced at the maneuver or solely at the match point if there are not any maneuvers. Alternative time transcriptions are possible within the same multiple shooting trajectory structure. As an example, absolute DSM epochs could be used as design parameters with additional constraints ensuring that events occur in logical order.

The phase start and end points are located at control nodes, and dictate the phase starting and ending position, velocity, mass states, and epoch. These start and end states are dependent on the departure and arrival type. Departure types include launches within a specified launch vehicle performance envelope and

direct insertions that allow for an initial departure maneuver as well as post-flyby states following a gravity assist at a body. The forward propagation state of the phase begins after all departure activities. Arrival types include analogs such as an impulsive arrival maneuver and the start of planetary flyby. Similarly, the backward propagation state and the phase end point "begins" prior to the arrival activities.

If a phase begins with a launch, a polynomial curve fit is employed to determine the initial spacecraft mass, $m_{0,}$ as a function of the C_3:

$$m_0 = (1-\sigma_{LV})(a_{LV}C_3^5 + b_{LV}C_3^4 + c_{LV}C_3^3 + d_{LV}C_3^2 + e_{LV}C_3 + f_{LV}) \tag{3}$$

where C_3 is equal to the square of the magnitude of the outgoing excess velocity vector (a function of the initial central body velocity), σ_{LV} is a user-specified launch vehicle margin, and a_{LV} through f_{LV} are the polynomial coefficients derived from launch vehicle performance curve published in the literature or supplied to the mission from the launch vehicle provider. The optimizer selects the initial control node velocity along this curve within any user-defined bounds on the right ascension, declination of the launch asymptote and C_3.

The full mission formulation is depicted in Figure 3, illustrating the multiple shooting strategy from body-based control nodes to the phase match points. Multiple trajectory phases between flybys are constructed for MGA problems with n maneuvers allowed for each phase. Flybys are modeled using the zero sphere-of-influence approach[34] in which the spacecraft position is matched to the flyby body at a control node, and the body imparts an instantaneous change in the spacecraft's central-body velocity (e.g., heliocentric velocity). In the construct of the transcription, the optimizer selects the initial and final velocity vectors of a phase ($\mathbf{v}_{\text{initial}}$ and $\mathbf{v}_{\text{final}}$ as noted previously), which, given the flyby body velocity, define the incoming and outgoing hyperbolic excess velocities relative to flyby body at a node connecting two phases: $\mathbf{v}_{\infty\text{-in}}$ and $\mathbf{v}_{\infty\text{-out}}$, respectively. Assuming the flyby is unpowered (i.e., no Oberth maneuver), the magnitude of these velocities must be equal for a physically realizable hyperbolic transfer. The first flyby constraint is then

$$\left|\mathbf{v}_{\infty-in}\right| = \left|\mathbf{v}_{\infty-out}\right|. \tag{4}$$

A second constraint at the flyby, ensures the spacecraft stays above a user-specified minimum flyby altitude, h_{min}:

$$\frac{\mu_{flyby\_body}}{v_\infty^2}\left[\frac{1}{\sin(\delta/2)} - 1\right] - r_{flyby\_body} + h_{\min} \le 0, \tag{5}$$

where δ is the flyby turn angle as illustrated in Figure 4, and μ_{flyby_body} and r_{flyby_body} are the gravitational parameter and radius of the flyby body, respectively. The flyby angle is defined as

$$\delta = \arccos\left(\frac{\mathbf{v}_{\infty-in} \cdot \mathbf{v}_{\infty-out}}{|\mathbf{v}_{\infty-in}||\mathbf{v}_{\infty-out}|}\right). \tag{6}$$

.

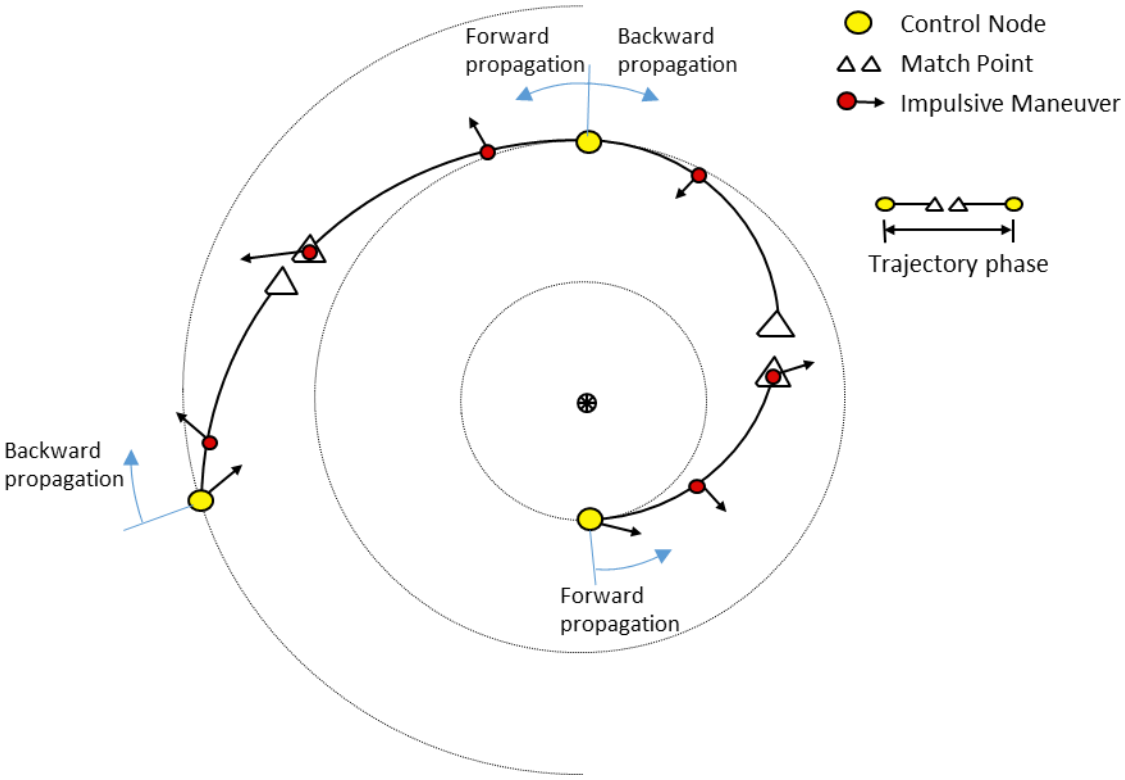

Figure 3: MGA*n*DSM*s* full-mission transcription

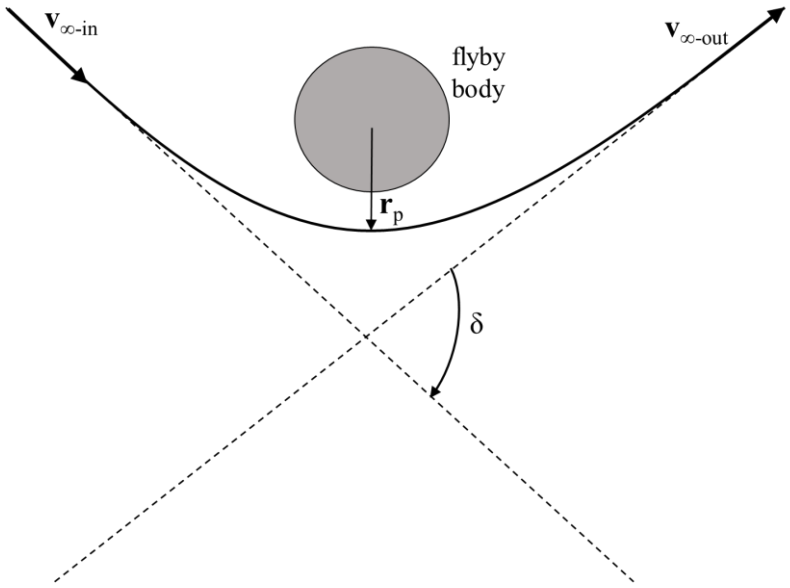

Figure 4: Flyby geometry

The MGA*n*DSM*s* transcription is integrated into the EMTG code base and enables the design of a wide variety of mission scenarios with any number of DSMs, multiple gravity assists (with or without powered flybys), and a range of mission-specific constraints directly prescribed in the optimization formulation. Control nodes are formulated in the same way as the Sims-Flanagan low-thrust phase transcription in EMTG.

This common configuration allows for mixed mission types in EMTG in which a mission can be composed of both chemical propulsion and electric propulsion journeys. Mission constraints such as entry interface azimuth, latitude, and longitude can be constrained for a planetary entry mission (e.g., sample return). Additionally, the solar phase angle can be constrained at arrival, which is frequently important for small body rendezvouses. A number of other constraints are outlined in Reference 35.

Derivatives

Accurate partial derivatives comprising the Jacobian are important to MGA*n*DSM*s* robustness and efficiency. As the match-point derivatives can be highly nonlinear, analytic partial derivatives enable the optimizer to converge with a less accurate initial guess and fewer iterations than with numerical derivatives via finite differencing. Furthermore, analytic derivatives save many computations required to compute the derivatives as compared with finite differencing, which is particularly beneficial for larger problems (i.e., many flyby bodies and many maneuvers). This efficiency gain is important not only for transcription efficacy in terms of solution of the NLP problem, but is critical to the automated global optimization outer loop.

It is the experience of the authors that, somewhat surprisingly, multiple-DSM, gravity-assist trajectories can be noticeably more sensitive to the quality of the initial guess than the low-thrust analog in a Sims-Flanagan transcription. It is postulated that the reason for this heightened sensitivity is the large change that a single maneuver can impart on the transfer topography. Contrastingly, a low-thrust transfer incurs relatively small perturbations from typical design variable changes after an iteration of a gradient-based optimizer because of the limitations of the low-thrust engine. This sensitivity can make identifying global optimums more difficult for high-thrust MGA*n*DSM trajectories than low-thrust MGA trajectories despite fewer design variables (in direct formulations) because a better initial guess is often required. This observation further supports the need for accurate derivatives.

A key benefit over transcriptions that apply Lambert-based targeting is that derivatives for MGA*n*DSM*s* are more readily derivable. Moreover, the derivatives are similar in form to those that exist for the low-thrust, gravity-assist transcription already existing in EMTG. The match point derivatives make up the majority of the non-zero Jacobian entries, and the process of calculating the critical match point derivatives for the Sims-Flanagan low-thrust transcription was detailed in Reference 36. Similar procedures for calculating MGA*n*DSM*s* derivatives can be followed, where for the match point constraint vector, $\mathbf{c}_{mp}$, the requisite partial derivatives for the Jacobian with respect to the decision variables are

$$\frac{\partial \mathbf{c}_{mp}}{\partial \boldsymbol{p}} = \frac{\partial \mathbf{S}_{mp}^{+}}{\partial \boldsymbol{p}} - \frac{\partial \mathbf{S}_{mp}^{-}}{\partial \boldsymbol{p}} \tag{7}$$

.

The state vector $\mathbf{S}_{mp}^{-}$ is simply,

$$\mathbf{S}_{mp}^{-} = \begin{bmatrix} \mathbf{r}^{-} \\ \mathbf{v}^{-} \\ m^{-} \end{bmatrix}, \tag{8}$$

where $\mathbf{r}^{-}$, $\mathbf{v}^{-}$, and m^{-} represent the position state, velocity state, and mass, respectively, on the forward propagation side of the match point. The match point state at the backwards propagation side follows in an analog form. The two terms on the right-hand side of Equation 7 can be expanded using the chain rule to expose the needed partial derivatives, and is written in a general form as:

$$\begin{aligned} \frac{\partial \mathbf{S}_k}{\partial p_i} = {} & \frac{\partial \mathbf{S}_k}{\partial r_{x,k-1}} \frac{\partial r_{x,k-1}}{\partial p_i} + \frac{\partial \mathbf{S}_k}{\partial r_{y,k-1}} \frac{\partial r_{y,k-1}}{\partial p_i} + \frac{\partial \mathbf{S}_k}{\partial r_{z,k-1}} \frac{\partial r_{z,k-1}}{\partial p_i} + \\ & \frac{\partial \mathbf{S}_k}{\partial v_{x,k-1}} \frac{\partial v_{x,k-1}}{\partial p_i} + \frac{\partial \mathbf{S}_k}{\partial v_{y,k-1}} \frac{\partial v_{y,k-1}}{\partial p_i} + \frac{\partial \mathbf{S}_k}{\partial v_{z,k-1}} \frac{\partial v_{z,k-1}}{\partial p_i} + \frac{\partial \mathbf{S}_k}{\partial t} \frac{\partial t}{\partial p_i}, \end{aligned} \tag{9}$$

where the subscript i indicates the i^{th} entry in the phase decision vector and k refers to the k^{th} segment in the phase. To calculate the terms of Equation 8 that will then form a substantial portion of the Jacobian, an augmented two-body state transition matrix (STM) can be chained from the control node to the match point across maneuver transitions, assuming Kepler propagation. The augmented STM for takes the form:

$$\mathbf{\Phi}(t,t_0)=\begin{bmatrix}\frac{\partial \mathbf{r}}{\partial \mathbf{r}_0} & \frac{\partial \mathbf{r}}{\partial \mathbf{v}_0} & \frac{\partial \mathbf{r}}{\partial m_0}\\ \frac{\partial \mathbf{v}}{\partial \mathbf{r}_0} & \frac{\partial \mathbf{v}}{\partial \mathbf{v}_0} & \frac{\partial \mathbf{v}}{\partial m_0}\\ \frac{\partial m_k}{\partial \mathbf{r}_0} & \frac{\partial m_k}{\partial \mathbf{v}_0} & \frac{\partial m_k}{\partial m_0}\end{bmatrix}=\begin{bmatrix}\frac{\partial \mathbf{r}}{\partial \mathbf{r}_0} & \frac{\partial \mathbf{r}}{\partial \mathbf{v}_0} & \mathbf{0}_{3\times 1}\\ \frac{\partial \mathbf{v}}{\partial \mathbf{r}_0} & \frac{\partial \mathbf{v}}{\partial \mathbf{v}_0} & \mathbf{0}_{3\times 1}\\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & 1\end{bmatrix} \tag{10}$$

representing the state sensitivities at time t with respect to variations in the state at an earlier time t_0. The upper-left quadrant entries comprise the standard two-body STM.[37] Provided a two-body propagation, the F and G Lagrange coefficients required for the STM are calculated along with the Kepler propagation. However, the augmented STM does not consider the state transition across the impulsive maneuvers. To address this deficiency, a maneuver transition matrix (MTM), is constructed to represent the sensitivities across maneuver k:

$$\mathbf{M}_k=\begin{bmatrix}\frac{\partial \mathbf{r}_{k+}}{\partial \mathbf{r}_{k-}} & \frac{\partial \mathbf{r}_{k+}}{\partial \mathbf{v}_{k-}} & \frac{\partial \mathbf{r}_{k+}}{\partial m_{k-}}\\ \frac{\partial \mathbf{v}_{k+}}{\partial \mathbf{r}_{k-}} & \frac{\partial \mathbf{v}_{k+}}{\partial \mathbf{v}_{k-}} & \frac{\partial \mathbf{v}_{k+}}{\partial m_{k-}}\\ \frac{\partial m_{k+}}{\partial \mathbf{r}_{k-}} & \frac{\partial m_{k+}}{\partial \mathbf{v}_{k-}} & \frac{\partial m_{k+}}{\partial m_{k-}}\end{bmatrix}=\begin{bmatrix}\mathbb{I}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 1}\\ \mathbf{0}_{3\times 3} & \mathbb{I}_{3\times 3} & \mathbf{0}_{3\times 1}\\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \frac{\partial m_{k+}}{\partial m_{k-}}\end{bmatrix}. \tag{11}$$

The augmented STM and MTM can then be chained together from the phase segment of interest, k, to the match point to generate the match point derivatives in which either a control node or maneuver define the end points of a segment. The resulting partial derivatives from the appropriate STM chain are used to aid the population of the right-hand side of Equation 8 for the segment of interest. Other partial derivatives of constraints (e.g., flyby altitude and v_∞ constraints) are outlined in Reference 38.

OPTIMIZATION FORMULATIONS

Both single-objective and multi-objective optimization strategies are applied to the MGA*n*DSM*s* transcription. The approach enables the generation of single global optimum in the case of a single objective, or a set of equally optimal solutions should multiple objectives be considered without the need for an initial guess. The automated routines are described in the following section.

Global Single-Objective Optimization

The NLP problem resulting from the MGA*n*DSM*s* transcription for a single objective can be stated as

$$\begin{aligned}\text{minimize:}\quad & f(\mathbf{x})\\ \text{subject to:}\quad & \mathbf{c}(\mathbf{x})\le 0\\ & A\mathbf{x}\le 0\\ & \mathbf{x}_{lb}\le \mathbf{x}\le \mathbf{x}_{ub}\end{aligned} \tag{12}$$

where $f(\mathbf{x})$is the objective function, $\mathbf{c}(\mathbf{x})$ is a vector of the nonlinear inequality constraints, A is a matrix of linear constraints, and $\mathbf{x}_{lb}$ and $\mathbf{x}_{ub}$ are vectors defining the lower and upper bounds on the vector of problem decision variables $\mathbf{x}$.

To exploit the availability of analytic derivatives, a gradient-based sequential quadratic programming (SQP) solver, namely SNOPT[39], is applied to optimize the stated NLP problem. While the local trajectory optimization problem solved with SNOPT is notably robust, it does require an initial guess and is only capable of local optimization. One approach to automating the trajectory design process is to apply a stochastic-based wrapper to guide a global search of the NLP design space, a global-local hybrid scheme.[40,23]

In this work a monotonic basin hopping algorithm is combined with SNOPT to provide a global search of the design space following from Reference 24. MBH is designed to solve global search problems in a multi-modal design space. It operates as a Monte-Carlo-like optimization scheme, taking stochastic "hops" in the design space from a base design. The algorithm is adept at identifying the global optimum in design spaces in which the local optima are often clustered together, exploiting the search area near any local optimum that is identified. The exploration component of the algorithm is enhanced by randomly resetting the base design from which to hop. Importantly, MBH does not require an initial guess and can be started from a random initial point design. The utility of the algorithm for interplanetary trajectory design has been demonstrated in several recent studies.[23,24,35] The MBH+NLP algorithm is summarized in Algorithm 1.[23]

Algorithm 1 Monotonic Basin Hopping with NLP

generate random point $\mathbf{x}$
run NLP solver to find point $\mathbf{x}^*$ using initial guess
$\mathbf{x}_{current} = \mathbf{x}^*$
if $\mathbf{x}^*$ is feasible **then**
  save $\mathbf{x}^*$ to archive
while not hit stop criterion **do**
  generate $\mathbf{x}'$ randomly perturbing $\mathbf{x}_{current}$
  for each TOF variable t_i in x' **do**
    if rand(0,1) < $\rho_{time\text{-}hop}$ **then**
      shift t_i forward or backward 1 synodic period
  run NLP solver to find point $\mathbf{x}^*$ from $\mathbf{x}'$
  if $\mathbf{x}^*$ is a feasible **and** $f(\mathbf{x}^*) < f(\mathbf{x}_{current})$ **then**
    $\mathbf{x}_{current} = \mathbf{x}^*$
    save $\mathbf{x}^*$ to archive
  else if $\mathbf{x}^*$ is infeasible and $\|c(\mathbf{x}^*)\| < \|c(\mathbf{x}_{current})\|$ **then**
    $\mathbf{x}_{current} = \mathbf{x}^*$
return best $\mathbf{x}^*$ in archive

Several objective functions are available for the user to select. The traditional minimization of ΔV can be used as well as the maximization of final mass. Maximizing the final mass while applying a launch vehicle model has the advantage of trading launch vehicle performance (i.e., mass versus C_3) from the departure planet versus DSMs using the spacecraft's propulsion system. Ultimately, the delivered mass to the destination is of critical importance to a mission. A straight maximization of delivered mass can be sensitive numerically because the derivatives of the rocket equation are not well behaved to the problem decision variables. Reference 31 notes that the derivatives of an objective function that takes the $\log_{10}$ of the final mass are much better behaved. Additionally, minimization of time of flight, launch date, and arrival date are possible objectives in EMTG, as well as maximization of launch date and arrival date.

Global Multi-objective Optimization

The multi-objective optimization problem can be stated as follows:

$$\begin{aligned} \text{minimize:} \quad & \mathbf{f}(\mathbf{x}) \\ \text{subject to:} \quad & \mathbf{c}(\mathbf{x}) \le 0 \\ & A\mathbf{x} \le 0 \\ & \mathbf{x}_{lb} \le \mathbf{x} \le \mathbf{x}_{ub} \end{aligned} \tag{13}$$

where **f(x)** is a vector of objectives

$$\mathbf{f}(\mathbf{x}) = [\, f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \quad \cdots \quad f_{n_{obj}}(\mathbf{x})]^{\mathrm{T}}, \tag{14}$$

$\mathbf{x}$ is a vector of design variables (with $\mathbf{x}_{lb}$ and $\mathbf{x}_{ub}$ lower and upper bounds), and n_{obj} is the scalar number of objectives. The objective space is n_{obj}-dimensional, and the objective functions are often coupled (i.e., containing the same design variables) and competing (i.e., the optimal solution in one objective is not the same optimal solution in the other objectives).

Given multiple objectives the aim of the optimization process is the generation of the Pareto front of solutions. The Pareto front[41] represents the set of equally-optimal tradeoff solutions, in which no improvement can be attained in one objective without degrading another objective. As such, the goal of multi-objective optimization is to identify as many Pareto-optimal solutions such that a representation of the Pareto front is generated. This hyper-surface of dimension n_{obj} then enables a tradeoff decision between the optimal designs with an understanding of objective function sensitivities.

The multi-objective concept of domination permits the comparison of a set of designs with multiple objectives, yielding a measure of the relative quality of the design. When comparing two multi-objective designs, the design $\mathbf{x}_1$ dominates design $\mathbf{x}_2$ if:

$$\begin{aligned} &\forall p : f_p(\mathbf{x}_1) \le f_p(\mathbf{x}_2) \qquad p = 1,2,\ldots,n_{obj} \\ &\qquad\qquad\qquad \textit{and} \\ &\exists p : f_p(\mathbf{x}_1) < f_p(\mathbf{x}_2) \qquad p = 1,2,\ldots,n_{obj} \end{aligned} \tag{15}$$

The designs are non-dominant to each other if neither design dominates the other. In a comparison of designs in a set, the best designs are those that are not dominated by any other designs, forming the non-dominated subset. In this way, any Pareto-optimal design is a member of the non-dominated subset associated with the entire feasible objective space and is located along the Pareto front.

The approach to multi-objective optimization of MGA*n*DSM trajectories in this development follows the recent work in multi-objective optimization of MGADSM trajectories using a Lambert-based transcription in Reference 31. However, that previous work limited transfers to a single DSM per phase, whereas multiple DSMs are enabled in the current implementation. As in the previous work, a multi-objective hybrid optimal control problem algorithm is structured with the modified NSGA-II as an outer-loop optimizer and the MBH+NLP algorithm as an inner-loop trajectory optimizer. With application of the NSGA-II, the outer loop provides global search capability with the capacity to optimize discrete parameters. It generates a representation of the Pareto front for any number of mission objectives including minimization of flight time, ΔV, and arrival v_∞ as well as the maximization of arrival mass, or any available inner-loop objective function. In turn, the complementary, inner-loop component provides robust, automated trajectory optimization capability, solving the resulting nonlinear programming problem using MBH+NLP.

Within this nested loop structure, the outer-loop employs a "cap and optimize" strategy. The outer loop chooses the design variables such as destination bodies, flyby bodies, launch vehicle, and bounds on the launch epoch, flight time, and final body arrival C3 to define a tractable inner-loop subproblem. Launch epoch is transcribed as a menu of candidate launch dates plus a launch window size. The method for time of flight is similar, in which the user specifies a list of flight times and the optimizer chooses one and sets it as the upper-bound for the inner-loop problem.

Flyby sequence selection is unique in that the outer-loop optimizer may vary the number of flybys. This variation creates non-uniform "chromosome" lengths in the NSGA-II. To accommodate this non-uniformity, the null-gene technique is applied to select the number and identity of flyby bodies. In this strategy, a list of possible flyby bodies and a maximum number of flybys for each journey is provided by the user. For each potential flyby, the outer-loop chooses from a list of the possible bodies and also a number of null options equal to the number of acceptable bodies. In this way, the outer loop is as likely to select "no flyby" for each opportunity as it is a flyby in the creation of the first generation of the NSGA-II population. The null gene approach has shown to be efficacious in the selection of the flyby sequence in a number of interplanetary problems.[33]

With the objectives selected and a menu of discrete design variables designated for the mission design problem at hand, the outer loop defines a trajectory optimization problem to be solved by the inner loop. The NSGA-II selects the optimal discrete parameters, bounding the trajectory problem, while the inner loop executes a global search according to the specified trajectory parameters for an optimal solution. Trajectory constraints are then efficiently handled by the MBH+NLP-based inner loop and the multi-objective problem of the outer loop is effectively unconstrained. If no feasible trajectory is identified within the inner loop, the design is penalized in the NSGA-II such that it will be dominated by feasible solutions. If two infeasible solutions are compared to each other, the less feasible design is considered to be dominated. Once the population of trajectories is solved in the inner loop, the individuals are sorted according to their objective function values using the NSGA-II non-dominated sorting routine, and a single relative fitness metric is assigned based on the individual's non-dominated front value and crowding distance (i.e., proximity to neighboring designs in solution space). A new parent population of top-performing solutions is then created with the NSGA-II tournament selection genetic operator. Next, the parent population is combined to generate an offspring population via the NSGA-II uniform crossover genetic operator. The offspring population of trajectories undergoes mutation with relatively small alterations of an individual's design variables, before the trajectories are optimized in the MBH+NLP loop. The process is then repeated until a maximum number of generations or a time limit is reached as illustrated in the algorithm flowchart of Figure 5. For further details on the multi-objective hybrid optimal control algorithm the reader is referred to Reference 29.

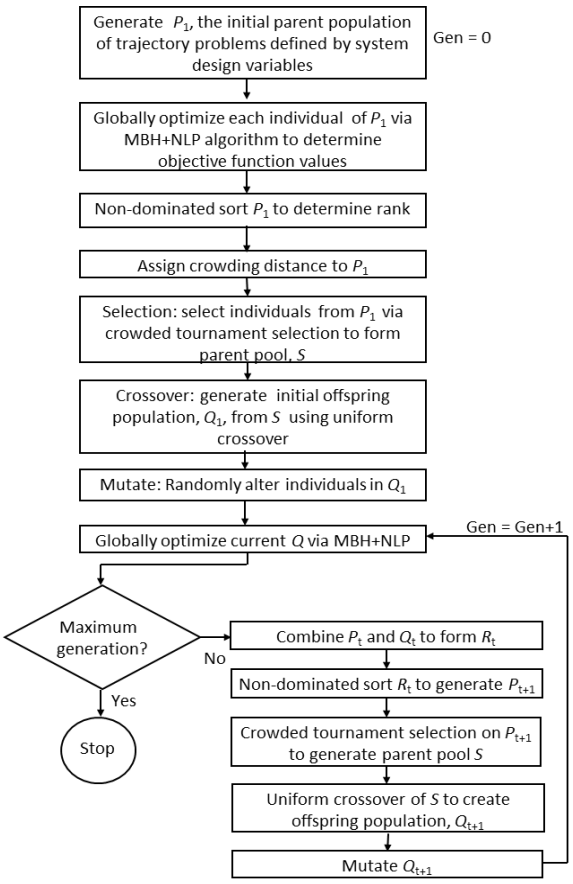

Figure 5: Multi-objective optimization algorithm flowchart

EXAMPLE APPLICATIONS

The MGA*n*DSM*s* phase transcription is applied to three interplanetary mission design problems to evaluate different performance aspects. In the first two examples, single-objective performance of the new EMTG transcription is compared to a Lambert-based transcription that allows a single DSM. The final example illustrates application to a multi-objective MGA*n*DSM problem.

Application to OSIRIS-REx

In this first example, multiple DSM options for early return options for the OSIRIS-REx mission are evaluated. These results are compared to scenarios with a single DSM using a Lambert-based targeting transcription, MGADSM*k* (see Reference 31 for a detailed description of MGADSM*k*). The OSIRIS-Rex trajectory problem is a compelling example because it involves an Earth gravity assist on the outbound journey, a rendezvous and subsequent stay at the asteroid Bennu, and then return to Earth with atmospheric entry interface requirements.

The project is interested in returning as early as possible while understanding the sensitivity of mission performance to the stay time at Bennu and the allowable bounds of the entry interface velocity azimuth. With the Earth arrival dates constrained to be at least 10 months earlier than the nominal 2023 arrival, a variety of scenarios are evaluated. The minimum stay time at Bennu was varied between 300 and 500 days in 100 day increments and the entry-interface azimuth is constrained between 46 and 66 degrees in one scenario, and 0 and 180 degrees, enforcing a prograde return, in a second scenario. In the latter scenario, the Bennu arrival date is constrained to be no later than October 15, 2018 to accommodate operational considerations. Pertinent parameters are outlined in Table 3.

Table 3. Trajectory assumptions, bounds, and constraints for OSIRIS-REx

Description	Value
Launch date open	September 11, 2016
Launch window	30 days
Launch declination	[-28.5, 28.5] deg
Max. launch C_3	29.27 km^2/s^2
Max. launch mass	1995 kg
Wait time at Bennu	<300, <400, or <500 days
Entry interface (EI) altitude	6503.14 km
Max. speed relative to atmosphere at EI	12.4 km/s
Entry interface altitude	6503.14 km
Entry interface latitude	37.0
Entry interface azimuth bounds	[46, 66] or [0, 180] deg.
Earth arrival date	< Dec. 1, 2022

A comparison of total mission ΔV using the two different transcriptions is highlighted in Table 4. In every case MGA*n*DSM*s* is able to identify a lower ΔV cost than MGADSM*k* as long as at least one maneuver is allowed. In the case of an Earth entry interface azimuth between 46 and 66 degrees and a 500 day minimum stay, the new transcription is able to reduce the ΔV cost by over 200 m/s. MGADSM*k* was allowed to run for 8 hours, but was not able to identify the more optimal DSM locations and event dates. This difference is attributable to the improved efficiency and robustness of the MGA*n*DSM*s* formulation within the global optimization framework. A single DSM provides a substantial reduction in ΔV versus no DSMs, but more than one DSM is not beneficial as indicated in Table 4. MGA*n*DSM*s* run time to achieve the assumed global minimum ranges from 8 seconds to 9 minutes on a 2.6 GHz Intel Core i7-496 processor. No initial guesses are provided to the optimizer.

Example trajectories for a 500 day minimum stay at Bennu with an Earth return azimuth between 46 and 66 degrees are shown in Figure 6. The body encounter dates are similar, but the placement and magnitude of the DSMs are significantly different, driving the large disparity in optimal ΔV. Each of the four phases includes a DSM, contributing to the 300 m/s savings versus a mission without DSMs. The DSM prior to the Earth flyby is particularly important as it facilitates a more optimal flyby geometry.

Table 4, OSIRIS-REx MGADSM*k* and MGA*n*DSM*s* comparison

Scenario	Minimum Stay Time (days)	ΔV (km/s)				
		MGAnDSMs max. DSMs: 0	MGAnDSMs max. DSMs: 1	MGAnDSMs max. DSMs: 2	MGAnDSMs max. DSMs: 3	MGADSMk (1 DSM)
46° < Azimuth < 66°, Bennu Arrival Free	300	1.492	1.330	1.331	1.331	1.352
	400	1.537	1.338	1.338	1.338	1.352
	500	2.026	1.711	1.712	1.711	1.923
0° < Azimuth < 180°, Bennu Arrival < 10/15/18	300	1.779	1.620	1.620	1.620	1.638
	400	1.779	1.622	1.622	1.622	1.638
	500	2.019	1.746	1.746	1.746	1.773

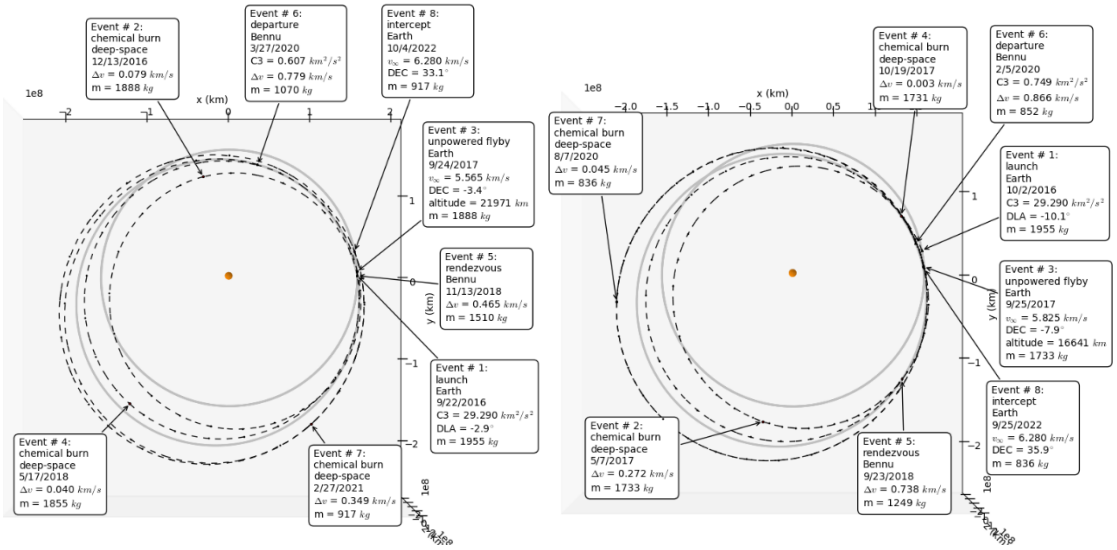

Figure 6: Example OSIRIS-REx trajectories for a min. 500 day stay at Bennu with an Earth return azimuth between 46 and 66 degrees: MGA*n*DSM*s* solution (left), MGADSM*k* solution (right)

Application to Cassini

The MGA*n*DSMs transcription is also evaluated for a Cassini mission test problem. This problem is challenging as the nominal interplanetary trajectory includes four gravity assists on the way to Saturn (Venus, Venus, Earth, Jupiter), and a single deep space maneuver is allowed in each phase. Problem parameters are outlined in Table 5, and an insertion maneuver into a highly eccentric orbit, assuming a burn at periapsis, is included in the ΔV total and final mass computation.

Table 5: Cassini example problem parameters and bounds

Description	Value
Launch date open	January 1, 1997
Launch window	365 days
Launch declination	[-28.5, 28.5] deg
Max. launch C_3	18.06 km^2/s^2
Launch vehicle curve	Atlas V, 551
Saturn arrival date	unbounded
Saturn insertion orbit semi-major axis	5,447,500 km
Saturn insertion orbit eccentricity	0.98
Objective function	Max: $\log_{10}$(final mass) for MGAnDSMs Min: ΔV for MGADSM*k*

The time to identify the globally optimal solution using one DSM across ten independent runs is listed in Table 5. The comparison runs are conducted with a 2.6 GHz Intel Core i7-496 processor. The mean time to global optimum is 16 minutes and the median time to optimum is 8 minutes. The known global optimum is identified in all ten cases using MGA*n*DSMs. Independent runs using MGADSM*k* are also executed, and while all of the runs identify a solution within 50 m/s of the globally optimal solution ΔV of 1.004 km/s after launch (launch C_3 of 18.07 km^2/s^2), none of the runs identify the solution within an eight hour run time. The mean computation time for MGADSM*k* to identify the best solution is an order of magnitude higher than MGA*n*DSMs. The globally optimal solution is depicted in Figure 7, illustrating a DSM at the apoapsis of the transfer trajectory preceding the second Venus flyby. Notably, the minimum ΔV and the maximum final mass solution are not the same. The ΔV associated with maximum final mass objective is 1.049 km/s with a final mass of 3221 kg, while the minimum ΔV solution has a final mass of 3129 kg. Note that the minimum ΔV objective is applied for the MGADSM*k* runs for improved efficiency in that transcription.

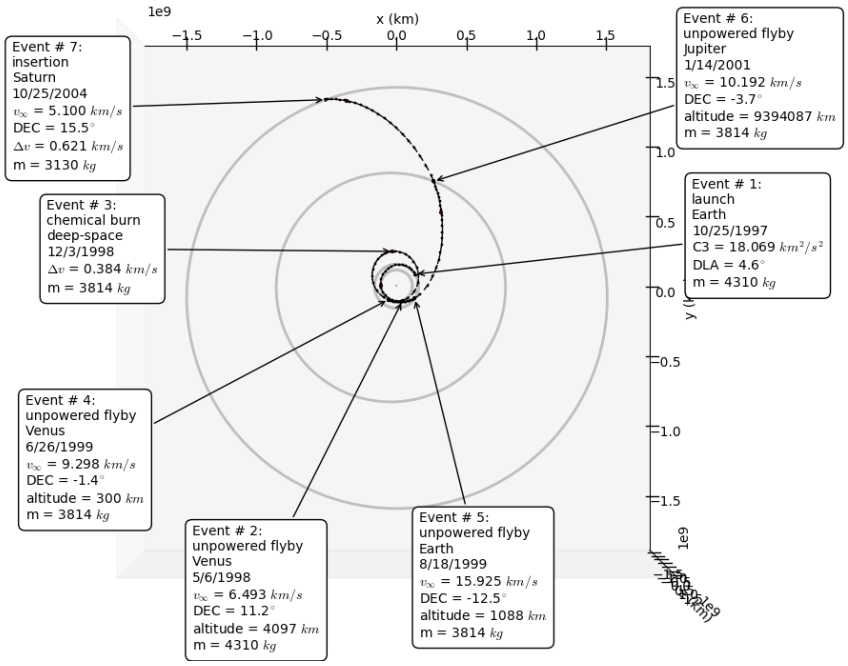

Figure 7: Optimal Cassini mission interplanetary trajectory

Table 6. Cassini Mission Δ*V* Comparison, MGADSM*k* versus MGA*n*DSM*s*

Run Number	MGA*n* DSM*s* Time to Identify Global Optimum (minutes)	MGADSM*k* Time to Identify Best Solution (minutes)	MGADSM*k* Time Identify Solution within 50 m/s of Global Optimum (minutes)	MGADSM*k* Best ΔV (km/s)
1	5.3	223.8	220.3	1.025
2	7.9	140.9	32.9	1.043
3	8.8	260.8	149.3	1.026
4	12.8	74.3	74.3	1.014
5	63.1	178.1	122.0	1.020
6	4.3	295.4	112.0	1.034
7	5.0	471.3	465.7	1.039
8	6.3	382.9	328.5	1.049
9	9.4	80.6	58.4	1.033
10	38.4	63.6	63.6	1.038
Mean	**16.1**	**217.2**	**162.7**	**1.032**
Median	**8.4**	**200.9**	**117.0**	**1.033**

Multi-objective Optimization of a Mission to Jupiter

The final example demonstrates the MGA*n*DSM*s* multi-objective optimization capability to search over different flyby bodies and generate a representation of the Pareto front of globally optimal solutions. A mission to Jupiter is examined with up to two DSMs per phase and three different objectives: maximize $\log_{10}$(final mass), minimize time of flight, and minimize Jupiter arrival v_∞. Early 2020 launch dates and flight times ranging from two to nine years are considered. Additionally, up to five gravity assists are available in the outer loop menu. The null gene approach is utilized to allow for a variable number of flybys using any

combination of Venus, Earth, and Mars. These outer-loop menu choices are listed in Table 7. Many parameters and assumptions are common to all inner-loop problems, and are outlined in Table 9. Also note that all trajectories start from an Atlas V-551 launch, and the inner-loop objective is to maximize final mass. Upon Jupiter arrival the spacecraft is inserted into a highly eccentric orbit using a periapsis burn, and the deduction of the propellant for the insertion burn is reflected in the final mass. In the outer loop, an NSGA-II population of 256 is used with a 15% mutation rate to encourage population diversity as listed in Table 8.

Table 7. Outer-Loop Design Variable Menu for Jupiter Example

Design Variable	Value	Resolution
Launch window open epoch	{1/1/2021, 1/1/2022, 1/1/2023, 1/1/2024}	1 year
Flyby body	{Venus, Earth, Mars, null, null, null}	n/a
Flight time	[730, 3467.5] days	182.5 days

Table 9. Common Trajectory Assumptions for Jupiter Example

Description	**Value**
Launch window	365.24 days
Launch declination	[-28.5, 28.5] deg
Launch vehicle curve	Atlas V, 551
Chemical I_{sp}	320 s
Jupiter arrival date	Determined by optimizer
Jupiter insertion orbit semi-major axis	10,054,900 km (140.6 R_J
Jupiter insertion orbit eccentricity	0.911
Maximum number of DSMs	2
Inner-loop objective function	Max: $\log_{10}$(final mass)
Inner-loop run time	40 minutes

Table 8. Outer loop optimization parameters for Jupiter Example

Parameter	Value
Population size	256
Mutation probability	15%
Objective functions	$\log_{10}$(final mass), TOF, arrival v_∞

The outer loop is stopped after 100 generations, which takes 3.5 days on a 2.6 GHz 64-core AMD Opteron. The optimization is completely automated after initial problem definition with no human oversight, and no user-supplied initial guesses. A representation of the globally-optimal Pareto front is shown in Figure 8. The Pareto front is three dimensional as there are three objectives. Labeled on the delivered mass versus time of flight projection plot are mission sequence tags representing a sampling of the wide variety of sequences in the non-dominated set. Mission sequences along the best non-dominated front include a mixture of classic and non-classic Jupiter flyby sequences: EJ, EEJ, EEEJ, EVEJ, EVEEJ, EVEEEJ, EVVEJ, EVVVEJ, EVMVEJ, EMJ, EMMJ, EMMMJ, EMMMMJ, EVMJ, EVMMJ, EVVMJ, EVEMJ, EMEJ, EEMJ, EMEEJ, EEMMJ, and EEEMMJ. The best performing trajectory in terms of maximum arrival mass is depicted in Figure 9. The trajectory is a classic VEEGA transfer (EVEEJ), delivering 3192 kg into the eccentric Jupiter orbit after a 7.9 year TOF. At the other end of the Pareto front is the shortest flight time solution, a direct Earth-to-Jupiter (EJ) transfer with a flight time of two years and a delivered mass of 642 kg. This solution is plotted in Figure 10, and has two DSMs on the out-bound journey. The first maneuver, 7 days after launch, offsets the launch vehicle, allowing for a lower launch C_3 and higher launch mass while leveraging the spacecraft I_{sp} (no consideration of spacecraft propellant tank limits are included in the example). The second maneuver is a broken plane maneuver to enable an efficient 180 degree transfer. Notably, scenarios with multiple Mars flybys preceding the final Jupiter phase achieve low arrival v_∞.

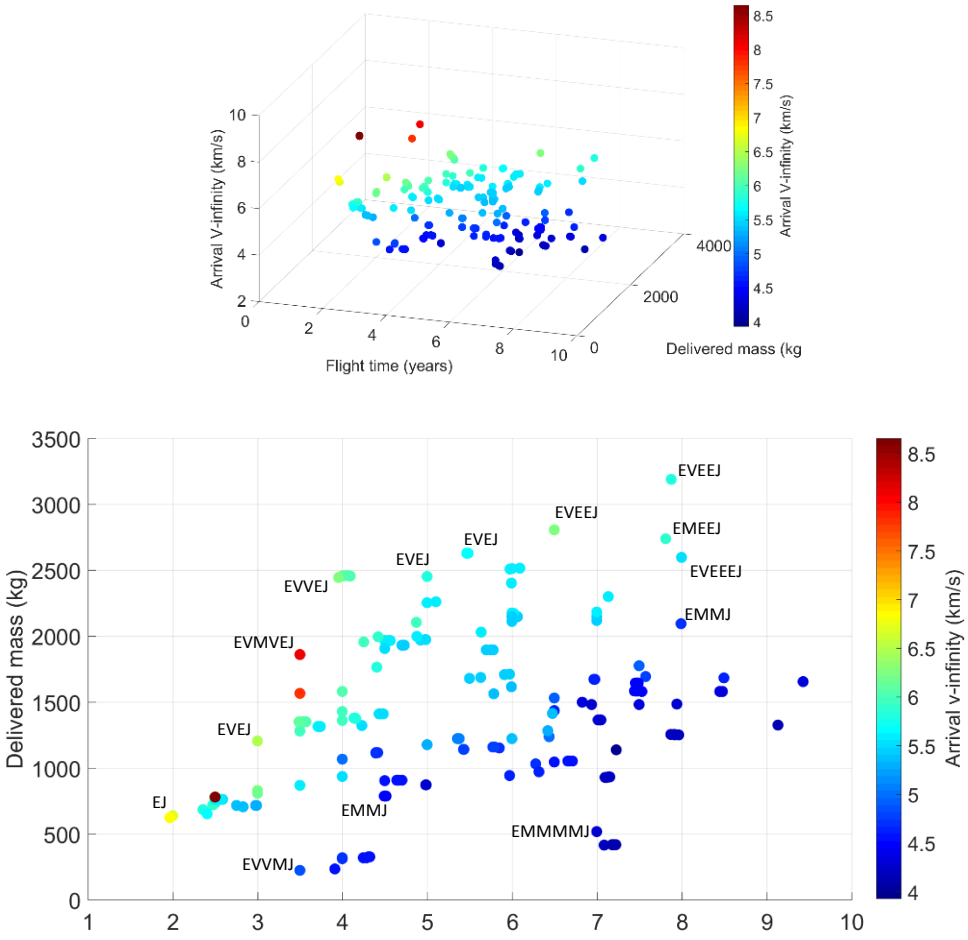

Figure 8: Representation of the Pareto front for the Jupiter example

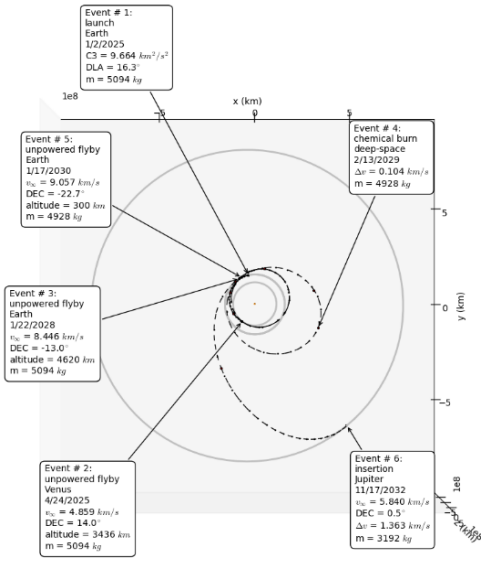

Figure 9: Maximum delivered mass solution (EVEEJ) from Jupiter example Pareto front

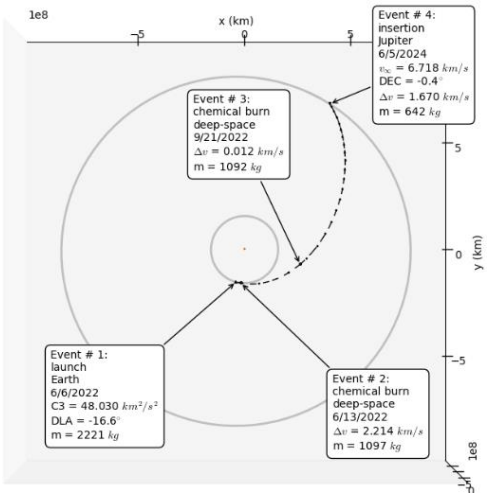

Figure 10: Minimum time of flight solution (EJ) from Jupiter example Pareto front

A number of other Pareto-optimal solutions have multiple DSMs in a single phase for improved performance over a single DSM per phase. Figure 11 highlights two two-DSM examples from the Pareto front. In several optimal trajectories two maneuvers are applied to efficiently rotate the line of apsides leading into a flyby.

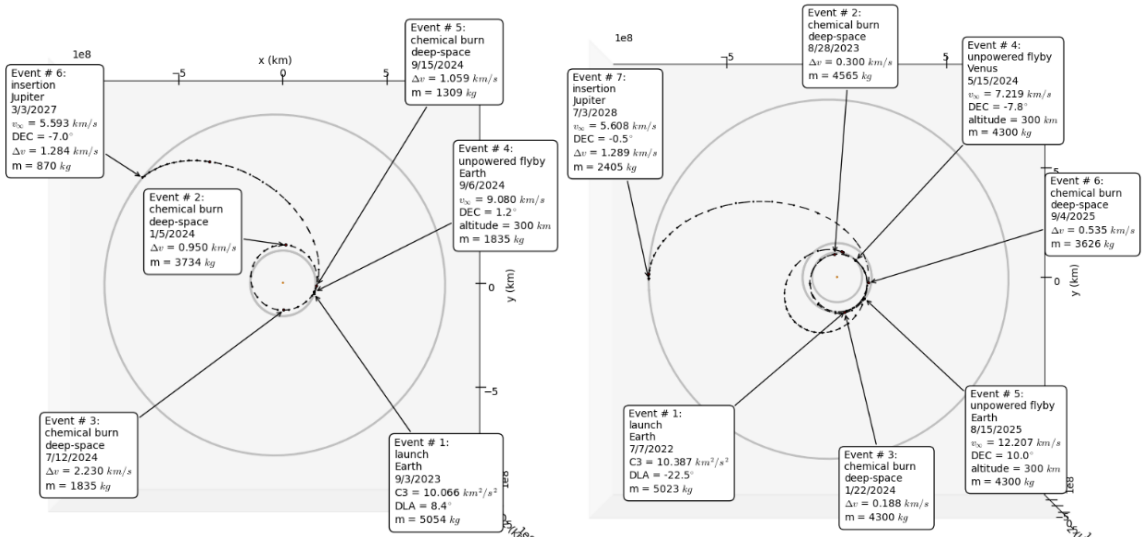

Figure 11: Pareto-optimal solutions with two DSMs in one phase: EEJ (left), EVEJ (right)

CONCLUSIONS

The MGA*n*DSM*s* trajectory transcription provides global optimization of high-thrust trajectories with an arbitrary number of maneuvers. Single-objective and multi-objective implementations are available, and the approach is capable of efficiently exploring the large design spaces associated with multiple gravity-assist scenarios. The multi-objective formulation is particularly well suited to exploring the design space for problems in which a grid search is computationally intractable. Moreover, the destination body and flyby bodies can be incorporated as design variables for a more thorough examination of the design space. Though not studied in this paper, the transcription allows for trajectory optimization of spacecraft with hybrid propulsion systems (i.e., both chemical and electric). The appropriate propulsion system for each mission phase could be traded within the construct of the multi-objective approach. Additionally, a variety of operational constraints can be directly incorporated into the optimization problem.

The transcription is analyzed on a range of mission design problems, providing a significant increase in computational efficiency in 1-DSM trajectories as compared to a Lambert targeting based transcription because of the robust multiple shooting scheme and incorporation of analytic match point derivatives. A representation of the three-dimensional Pareto front of globally-optimal solutions for a multiple gravity assist mission to Jupiter is developed, allowing for an understanding of the tradeoff between maximum delivery mass, minimum time of flight, and minimum arrival v_∞.

ACKNOWLEDGEMENTS

The authors thank Jeremy Knittel for valuable testing of the algorithm. Additionally the authors are grateful to Michael Shoemaker and Alex Pini for excellent feedback on this paper.

REFERENCES

[1] Lawden, D., *Optimal Trajectories for Space Navigation*, Butterworths Publishers, London, 1963.

[2] Edelbaum, T. N., "How Many Impulses?," *Astronautics and Aeronautics*, Vol. 5, No. 11, 1967, pp. 64-69.

[3] Gobetz, F. W., and Doll J. R., "A Survey of Impulsive Transfers," *AIAA Journal*, Vol. 7, No. 5, 1969, pp. 801-834.

[4] Eckel, K. G., "Optimum Transfer in a Central Force Field with n Impulses, "*Astronautica Acta*, Vol. 9, No. 5/6, 1963, pp. 302-324.

[5] Prussing, J. E., "Optimal Four-Impulse Fixed-Time Rendezvous in the Vicinity of a Circular Orbit, *AIAA Journal*, Vol 7, No. 5, 1969, pp. 928-935.

[6] Prussing, J. E., and Chiu, J. H., "Optimal Multiple-Impulse Time-Fixed Rendezvous Between Circular Orbits," *Journal of Guidance, Control, and Dynamics*, Vol. 9, Jan. – Feb. 1986, pp. 17-22.

[7] Petropoulos, A. E., Longuski, J. M., Bonfiglio, E. P., "Trajectories to Jupiter via Gravity Assist from Venus, Earth, and Mars," *Journal of Spacecraft and Rockets*, Vol. 37, No.6, November-December 2000, pp. 776-783.

[8] Handelsman, M., and Lion, P. M., "Primer vector on fixed-time impulsive trajectories." *AIAA Journal*, Vol. 6, No. 1, 1968, pp. 127-132.

[9] Jezewski, D. J., and Rozendall, H. L., "An Efficient Method for Calculating Optimal Free-Space N-Impulsive Trajectories, *AIAA Journal*, Vol. 6, No. 11, 1968, pp. 2160-2165.

[10] Sauer, C. G., Jr., "MIDAS: Mission Design and Analysis Software for the Optimization of Ballistic Interplanetary Trajectories," Journal of the Astronautical Sciences, Vol. 37, July-Sept. 1989, pp. 251–259.

[11] Olympio, J. T., and Izzo, D., "Designing Optimal Multi-Gravity-Assist Trajectories with Free Number of Impulses," 21st International Symposium on Space Flight Dynamics, Toulouse, France, September 2009.

[12] Byrnes, D. V. and Bright, L. E., "Design of High-Accuracy Multiple Flyby Trajectories Using Constrained Optimization," Paper AAS 95-307, AAS/AIAA Astrodynamics Specialist Conference, Halifax, Nova Scotia, Canada, August 1995.

[13] Sims, J. A., Finlayson, P. A., Rinderle, E. A., Vavrina, M. A., and Kowalkowski, T. D., "Implementation of a Low-Thrust Trajectory Optimization Algorithm for Preliminary Design," Paper AIAA-2006-6746, AIAA/AAS Astrody-namics Specialist Conference, Keystone, CO, August 2006.

[14] Atchison, J. A., Ozimek, M. T., Scott, C. J., and Siddique, F. E., "Robust High-Fidelity Gravity-Assist Trajectory Generation Using Forward/Backward Multiple Shooting," AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA, January 2015.

[15] Williams, S. N., "Automated Design of Multiple Encounter Gravity-Assist Trajectories," M.S. Thesis, School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, August 1990.

[16] Patel, M. R. and Longuski, J. M., "Automated Design of Delta-V Gravity-Assist Trajectories for Solar Sys-tem Exploration," AAS Paper 93-682, AAS/AIAA Astrodynamics Conference, Victoria, British Columbia, Canada, August 16-19, 1993.

[17] Lantukh, D. V., and Russell, R. P., " V-Infinity Leveraging Boundary-Value Problem and Application in Spacecraft Trajectory Design," *Journal of Spacecraft and Rockets*, Vol. 52, No. 3, 2015, pp. 697-710.

[18] P. J. Gage, R. D. Braun, I. M. Kroo, "Interplanetary Trajectory Optimization Using a Genetic Algorithm," *Journal of the Astronautical Sciences*, 43 (1) ,1995, pp. 59–75.

[19] Vasile, M., and Pascale, P. D., "Preliminary Design of Multiple Gravity-Assist Trajectories," Journal of Spacecraft and Rockets, Vol. 43, No. 4, July–Aug. 2006, pp. 794–805.

[20] Conway, B., Chilan, C., and Wall, B., "Evolutionary principles applied to mission planning problems," *Celestial Mechanics and Dynamical Astronomy*, Vol. 97, No. 2, 2007, pp. 73-86..

[21] Abdelkhalik, O., and Mortari, D., "N-Impulse Orbit Transfer Using Genetic Algorithms," *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, March–April 2007, pp. 456–459.

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

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

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

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

[26] Peralta, F., Flanagan, S., "Cassini interplanetary trajectory design," *Control Engineering Practice*, Vol. 3, Issue 11, November 1995, pp. 1603-1610.

[27] Lauretta, D. S., and OSIRIS-Rex Team. "An overview of the OSIRIS-REx asteroid sample return mission." Lunar and Planetary Institute Science Conference, vol. 43, p. 2491. 2012..

[28] Vavrina, M. A., and Howell, K. C., "Multiobjective Optimization of Low-Thrust Trajectories Using a Genetic Algorithm Hybrid," AAS/AIAA Space Flight Mechanics Meeting, Savannah, GA, February 2009.

[29] Vavrina, M. A., Englander, J. A., and Ghosh, A., "Coupled Low-Thrust Trajectory and Systems Optimization via Multi-objective Hybrid Optimal Control," AAS/AIAA Space Flight Mechanics Meeting Williamsburg, Virginia, January 2015.

[30] Englander, J. A., Vavrina, M. A., Ghosh, A.R., "Multi-objective Hybrid Optimal Control for Multiple-Flyby Low-Thrust Mission Design," AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, Virginia, January 2015.

[31] Englander, J. A., and Vavrina, M. A., "Multi-Objective Hybrid Optimal Control For Multiple-Flyby Interplanetary Mission Design Using Chemical Propulsion," AAS/AIAA Astrodynamics Conference, Vail, Colorado, August 2015.

[32] Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, 2001.

[33] Englander, J. A., "Automated Trajectory Planning for Multiple-Flyby Interplanetary Missions," PhD thesis, University of Illinois at Urbana-Champaign, April 2013.

[34] Uphoff, C., "Orbit Design Concepts for Jupiter Orbiter Missions," AIAA Mechanics and Control Conference, AIAA Paper 74-781, Anaheim, California, August 1974.

[35] Englander, J. A., Vavrina, M. A., and Hinckley, D. H., "Global Optimization of Low-Thrust Interplanetary Trajectories Subject to Operational Constraintes," AAS/AIAA Spaceflight Mechanics Meeting, Napa, California, February, 2016.

[36] 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," AAS/AIAA Spaceflight Mechanics Meeting, Santa Fe, New Mexico, January 2014.

[37] Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*, Reston, Virginia, American Institute of Aeronautics and Astronautics, Inc., 1999.

[38] Ellison, D. H., Englander, J. A., and Conway, B. A., "Robust Global Optimization of Low-Thrust, Multiple-Flyby Trajectories," AAS/AIAA Astrodynamics Specialist Conference, Hilton Head, SC, August 2013.

[39] Gill, P.E., Murray, W., and Saunders, M.A., "SNOPT: An SQP algorithm for large-scale constrained optimization." *SIAM journal on optimization*, Vol. 12 No. 4, 2002, pp. 979-1006.

[40] Vavrina, M.A., Howell, K.C., "Global Low Thrust Trajectory Optimization through Hybridization of a Genetic Algorithm and a Direct Method," AIAA/AAS Astrodynamics Specialist Conference, Honolulu, HI, August 2008.

[41] Pareto, V. Manuale di economica politica, societa editrice libraria, Milano, Italy, 1906; translated into English by A. Schwier as Manual of Political Economy, MacMillan Press, New York, 1971.