View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by NASA Technical Reports Server

(Preprint) AAS 18-401

A NOVEL MULTI-SPACECRAFT INTERPLANETARY GLOBAL TRAJECTORY OPTIMIZATION TRANSCRIPTION

Sean W. Napier[*,†], **Jay W. McMahon**[‡]

As the frontier of space exploration continues to advance, so does the design complexity of future interplanetary missions. This increasing complexity includes a class of designs known as *Distributed Spacecraft Missions*; missions where multiple spacecraft coordinate to perform shared objectives. Current approaches for global trajectory optimization of these Multi-Vehicle Missions (MVMs) are prone to shortcomings, including laborious iterative design, considerable human-in-the-loop effort, treatment of the multi-vehicle problem as multiple, separate trajectory optimization subproblems, and poor handling of coordination objectives and constraints. This leads to suboptimal solutions where the whole is less than the sum of its parts. There are only a handful of software platforms in existence capable of fully-automated, rapid, interplanetary global trajectory optimization, including the Gravity Assisted Low-thrust Local Optimization Program (GALLOP), and the Evolutionary Mission Trajectory Generator (EMTG). However, none of these tools is capable of performing such tasks for MVM designs. We present a fully-automated technique which frames interplanetary MVMs as Multi-Objective, *Multi-Agent* Hybrid Optimal Control Problems (MOMA HOCP). First, the basic functionality of this technique is validated on the single-vehicle problem of reproducing the Cassini interplanetary cruise. The technique is then applied to explore the possibility of a dual-manifest mission to the Ice Giants, Uranus, and Neptune. A single trajectory with flybys of both planets has been shown to be infeasible with only a single spacecraft anytime between 2020 and 2070.

INTRODUCTION

Framing interplanetary spacecraft trajectory optimization as a hybrid optimal control problem (HOCP) has proven an effective approach.[2,3] In this framework, trajectory optimization is a Mixed-Integer Programming (MIP) problem. Some decision variables are discrete (integers) while others are continuous (floating point), necessitating distinct optimization routines for each category of variable. Furthermore, the resulting mission designs are points within a solution space spanned by multiple objectives (i.e., minimum fuel versus minimum time of flight). Thus, in order to effectively characterize the solution space for a given mission design problem, a *multi-objective* HOCP framework is essential. However, while tools exist to solve interplanetary multi-objective HOCPs for a *single* spacecraft, no tool exists to optimize *multi-spacecraft*, multi-objective global optimization problems. Addressing the shortcomings of current approaches to MVM optimization, including the methods for handling of coordination objectives and constraints, are key to enabling the design of future MVMs.

[*]Navigation and Mission Design Branch, NASA Goddard Space Flight Center, sean.napier@nasa.gov
[†]Ph.D. Student, Colorado Center for Astrodynamics Research
[‡]Assistant Professor, Colorado Center for Astrodynamics Research, mcmahojw@colorado.edu

This work explores the application of a fully automated (i.e., requiring no initial guess) Multi-Objective Multi-Agent (MOMA) optimization to an interplanetary global trajectory optimization problem: designing an Ice Giant Multi-Mission. The individual spacecraft in the MVM are treated as *agents*, i.e., intelligent tokens, that have optimizable states and behavior, which cooperate to achieve coordinated objectives in a decision space bounded by multiple coordination constraints.[4,5] With this technique, the user only needs to specify the bounds of the MVM problem, and the MOMA HOCP scheme optimizes the solution space with no *human-in-the-loop effort*. Further, in treating the constituent spacecraft as agents within a coupled decision space, resulting point solutions are more indicative of the true behavior of the solution space compared to those gleaned from an approaches where the MVM is split into separate sub-problems.

The optimization technique of this work is comprised of four nested components: an *outer-loop* which only optimizes discrete variables via a population based method, an *inner-loop* stochastic global search method which traverses the continuous variable decision space bounded by its outer-loop decision vector, a local optimizer which finds a local minima in the inner-loop's global search space, and a trajectory transcription to evaluate a hybrid decision vector with respect to an objective. The contributions of this paper are threefold: 1) an outer-loop transcription to pose a MVM as a single, coupled trajectory optimization problem, 2) three new outer-loop coordination constraints, and 3) an outer-loop coordination objective approach.

PROBLEM FORMULATION

The goal of a multi-objective optimization problem is to find the optimal non-dominated front that depicts the fundamental trades between objectives.[6] The outer-loop performs this task efficiently via a *cap and optimize* approach.[11] For a given decision vector, one objective, such as ΔV, is optimized by the inner-loop, while secondary objectives are treated caps which the outer-loop imposes on inner-loop decision variables. The outer-loop only optimizes integer decision vectors, where integers encode items such as gravity assist targets. Each outer-loop decision vector defines a trajectory optimization problem which is in turn optimized by an instance of the inner-loop. The result of this optimization is passed back to the outer-loop for ranking. Secondary objectives for each inner-loop solution are evaluated by the outer loop during ranking. The larger the population size in the outer-loop, the more instances of the inner-loop must be run, ideally in parallel.

The outer-loop uses the Non-Dominated Sort Genetic Algorithm-II (NSGA-II).[6,7] In this work, each point on the outer-loop's Pareto front represents a spacecraft fleet. One level down, the inner-loop exclusively chooses continuous decision variables such as launch date, C_3, or time of flight (TOF) between flyby targets. The choice of continuous variables is dependent upon the choice of discrete variables. During the course of this work, the authors have experimented with different versions of the inner-loop to improve reliability. NSGA-II reliably finds the multi-objective Pareto front for a given problem space, but the quality of each point on this Pareto front is limited by the power of the inner-loop to reliably find the global optimum. The first iteration of this inner-loop used an evolutionary algorithm known as Differential Evolution (DE/best/2/bin).[2] This produced results that varied widely when applied to a modestly wide global search space. This approach was improved wrapping a Monotonic Basin Hopping (MBH) global search algorithm around DE/best/2/bin, at which point DE/best/2/bin proved a modestly reliable local optimizer. MBH *hops* through the global search space across local minima found by the local optimizer to arrive, given ample run time, within statistical *striking distance* of the global minimum of the cost function.[17] Finally, trajectories are transcribed using the Vinko-Izzo Multiple Gravity Assists with one Deep

Space Maneuver (MGA1DSM) transcription, a direct method that is easy to formulate and ideally suited to optimization with evolutionary algorithms.[8]

MBH without a gradient search was tested as a standalone inner-loop, but MBH and DE/best/2/bin were found to perform better together than either did alone. The MBH+DE/best/2/bin inner-loop's performance is first demonstrated on reproducing Cassini's EVVEJ interplanetary cruise, proving the functionality of the inner-loop to find the global optimum trajectory, in terms of minimum ΔV. C_3, Right Ascension, and Declination of the launch asymptote (RLA and DLA) are free to vary while the launch date is constrained to occur between Oct 1 and Oct 31 1997. The TOF of each trajectory phase was bounded to within ± 10 days of the nominal Cassini trajectory phase flight times. No initial guess seed was provided to the inner-loop optimizer. While evolutionary algorithms make suitable breadboards for exploring a problem which cannot leverage gradient information, they lack the ability to handle most constraints that were of interest in the problem posed by this work. The latest inner-loop uses MBH as the global search method and MATLAB's *fmincon* as the local optimizer, which enables faster acquisition of solutions and explicit linear constraints.

All the code in this work is written in MATLAB. Planetary orbit states were acquired from the JPL Horizons database at a single epoch, and were propagated inside the optimizer, along with spacecraft trajectories, using a universal variable Kepler propagator. A single impulsive maneuver is allowed between each pair of flyby bodies. With the inner-loop's functionality validated, the full MOMA HOCP technique was put to work on an Ice Giant Multi-Mission preliminary design. The mission design consists of two high-thrust chemical propulsion spacecraft, one of which must intercept Uranus and the other Neptune. The MOMA HOCP optimizer is tasked with optimizing the flyby sequences for each spacecraft for total ΔV, while also trading against time of flight. All spacecraft are subject to the initial coordination constraint of sharing a launch vehicle. This constraint is formulated by holding the launch epoch, C_3, RLA and DLA as identical for all spacecraft in the fleet. This novel active coordination constraint approach guides the optimizer through the true multi-mission solution space, coupling the performance of one spacecraft to the other.

MVM Outer-Loop Transcription

An outer-loop integer decision vector is treated as a chromosome by the NSGA-II algorithm. Each row is a vector corresponding to the trajectory of a single spacecraft. Each column contains a gene whose value is the index from a particular decision variable menu. In each row, the first four elements comprise a header of bounding parameters: launch window bounds, global TOF cap, C_3 bounds, and minimum number of shared flyby genes. Beyond this header, all remaining genes encode flyby targets. The destination planet is fixed — it does not evolve during successive generations of the NSGA-II algorithm. While the header parameters need not necessarily be identical for both spacecraft, the shared launch asymptote constraint forces them to be. For example:

$$X = \begin{bmatrix} LW & T_{cap} & C_3 & N_{shfb} & \text{Planet}_{1,1} & \text{Planet}_{1,2} & \text{Planet}_{1,3} & \text{Destination}_1 \\ LW & T_{cap} & C_3 & N_{shfb} & \text{Planet}_{2,1} & \text{Planet}_{2,2} & \text{Planet}_{2,3} & \text{Destination}_2 \end{bmatrix} \tag{1}$$

$$= \begin{bmatrix} 1 & 5 & 2 & 1 & 1 & 10 & 4 & 6 \\ 1 & 5 & 2 & 1 & 2 & 3 & 5 & 7 \end{bmatrix} \tag{2}$$

$$= \begin{bmatrix} \{1/1/2030, 5/1/2030\} & 10\,yr & [50,60]\,km^2/s^2 & (0\,shfb) & \text{Venus} & \text{NULL} & \text{Jupiter} & \text{Uranus} \\ \{1/1/2030, 5/1/2030\} & 10\,yr & [50,60]\,km^2/s^2 & (0\,shfb) & \text{Earth} & \text{Mars} & \text{Saturn} & \text{Neptune} \end{bmatrix} \tag{3}$$

The number of rows of this chromosome equals the maximum allowed number of spacecraft in a fleet while the number of columns is equal to the length of the header plus the maximum number of allowed flyby targets for a spacecraft. Note that the gene 10 in equation (2) evaluates to NULL. All flyby genes when generated have a 50% chance of being null genes. This is accomplished by generating a gene as an index equal to a random number between 1 and double the length of the allowable flyby genes menu. If the chosen gene corresponds to an index beyond the length of the flyby menu, it is pruned by the inner loop as NULL.

Outer-Loop Coordination Objectives (Minimax)

The approach taken for multi-agent constraints was a novel *weakest link* formulation. For a given objective function, the cost assigned to the fleet is equal to that of the highest cost spacecraft. In the outer-loop, minimal-cost individuals are dominant. Over successive iterations, evaluating the fleets' cost this way leads to decreasing cost of each spacecraft in the fleet. This is known in game theory as a *minimax* problem. This approach has proven highly effective in optimization via integer genetic algorithms and in multi-agent optimization problems, but has never been applied to a interplanetary MVM design. In this work, TOF is a handled as an outer-loop minimax objective, with the cost assigned to a trial fleet equal to the TOF of slowest spacecraft to reach its destination.

Trajectory Transcription

Trajectories in this work are parametrized according to the MGA1DSM transcription.[8] Mission phases are bracketed by control nodes, which in this work are planets. All trajectories begin at the Earth, and propagate according to two-body patched conics assumptions. Each mission phase is assigned a TOF and is composed of a forward propagated Kepler arc for some fraction of that TOF, hereon known as the DSM Index, followed by an impulsive deep space maneuver (DSM) given by the minimum energy Lambert transfer arc to intercept the next flyby target within the remaining fractional TOF. In addition to choices of DSM index η and total phase TOF, the first phase arc begins with the choice of launch date in Modified Julian Date (MJD), C_3, RLA, and DLA which all together define the initial position and outgoing velocity of the spacecraft. All subsequent phases are transcribed with four parameters: h_{flyby} (flyby altitude), β (angle of the flyby plane in the body frame of the planet), η, and total phase TOF. Shown in Fig. 1, this transcription ensures unpowered flybys.

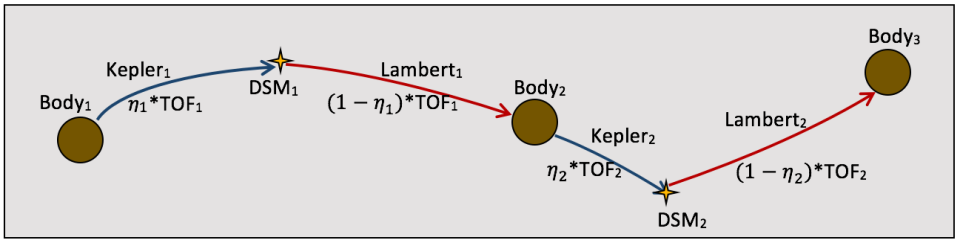


Figure 1: Illustration of the MGA1DSM transcription. This transcription is well suited for use with evolutionary algorithms and is readily applied to trajectory optimization problems with small bodies, planets, or both.

Inner-Loop Global Search Method

The inner-loop is executed in real-time by the outer-loop, with no user intervention. This requires the inner-loop's routine to be robust, reliable, and requiring no initial guess. It is necessary to wrap a global search method around the local optimizer to insure statistical acuquisition of the global optimum. The stochastic global search method employed in this work is version of Monotonic Basin Hopping (MBH).[13] This algorithm derives its name from the supposition that the generally nonconvex cost function being optimized contains local optima within *basins*, which can be traversed by a local optimizer.

Once the local optimizer drives a sub-optimal initial guess down to the bottom of a basin, MBH perturbs the current decision vector by some random hop distance out of the basin and begins exploring the global space. This process is repeated until either a maximum number of hops have been performed or an elapsed time has passed (both of which are user-selectable parameters). This process is illustrated in Fig. 2. MBH thus hops through the global search space stochastically. Its performance has been greatly improved by using a Nonlinear Programming (NLP) solver to quickly optimize local minima.[11] The following pseudocode captured in **Algorithm 1** describes the version of MBH developed for this work.

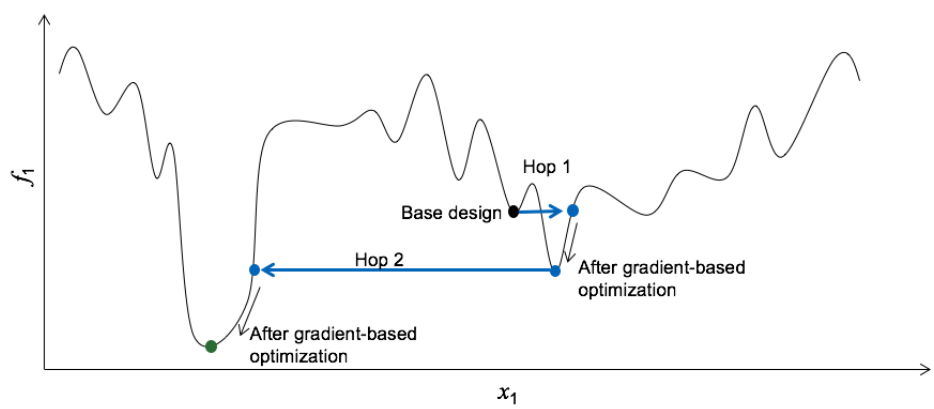


Figure 2: Illustration of the MBH process.[11] While the cost function is not monotonically increasing or decreasing everywhere, there exist local basins with monotonically decreasing intervals, enabling convergence to local minima.

Algorithm 1: Monotonic Basin Hopping

```
Initialize f_best
while current iteration < max iterations
    Generate random point x.
    Run local optimizer to find point x* using initial guess x.
    x_current = x*
    if x* is a feasible point then
        if f(x*) < f_best then
            f_best = f(x*)
            save x* to archive
        end if
        while N_not improve < N_max
            generate x' by randomly perturbing x_current
            Run local optimizer on x' to find x*
            if (x* is a feasible point) & (f(x*) < f(x_current)) then
                N_not improve = 0
                x_current = x*
                if f(x*) < f_best then
                    f_best = f(x*)
                    save x* to archive
                end if
            else
                N_not improve = N_not improve + 1
            end if
        end while
    end if
end while
return best x in archive
```

Single Objective Local Optimizer

Now that we have described how a trajectory is parametrized for the optimizer, we discuss the implementation of the first layer of the global optimization process in this work. Of many varieties of so-called *gradient-free* approaches to optimization of continuous variables, Differential Evolution (DE) has been shown to be among the most well-suited to flight dynamics applications. There are many varieties of DE, but the one used in this work is known as DE/best/2/bin (**Algorithm 2**). Instead of using a Nonlinear Programming (NLP) solver to find local optima with the assistance of analytical derivatives, DE is a population-based evolutionary algorithm which performs a stochastic downhill walk via a genetic crossover operator, using no analytical derivative information. Approaches like this are well-suited to problems where the analytical behavior is unknown or untenable. In this work, we augment the robustness of the DE/best/2/bin routine with the addition of a mutation operator to deter the algorithm from getting stuck in local optima. DE/best/2/bin may

also be used as an inner-loop global search in place of MBH, however, due to its inability to eject itself from a sub-optimal local minimum, MBH is more suitable. We report that in the benchmark problems examined in this work, the approach of nesting DE/best2/bin within MBH was found to be superior to employing either method alone.The key feature which separates DE from other evolutionary algorithms is its use of a *difference vector*- – a random direction along which the current best decision vector is perturbed that estimates the local gradient. This feature more actively facilitates the downhill random walk as opposed to using the canonical genetic operators alone. The pseudocode for this routine is shown below.

Algorithm 2: Differential Evolution (DE/best/2/bin) with Mutation

---

Given a cost function f and maximum number of iterations, generate a random population **P** of decision vectors $\boldsymbol{u}^*$ and evaluate their cost $J^* = f(\boldsymbol{u}^*)$.

while current iteration $<$ max iterations
  for each decision vector $\boldsymbol{u*}$
    Randomly choose decision vectors $\boldsymbol{u_1}$, $\boldsymbol{u_2}$, $\boldsymbol{u_3}$, $\boldsymbol{u_4}$
    Calculate the difference vector $\boldsymbol{d} = \boldsymbol{u_1} - \boldsymbol{u_2} + \boldsymbol{u_3} - \boldsymbol{u_4}$
    for each decision vector element $\boldsymbol{u_i^*}$
      Generate: F, ω, CR, MR $\in$ rand(0,1); MR$\ll$CR
      if $\omega > CR$
        $u_{trial,i} = u_{best,i} + Fd_i$
      else
        $u_{trial,i} = u_i*$
      end if

      $\omega_{Mutation} = rand(0,1)$
      if $\omega_{Mutation} < MR$
        $u_{trial,i} = rand(x_{lb,i}\ ,\ x_{ub,i})$
      end if

      if $u_{trial,i} < x_{lb,i}$ **or** $u_{trial,i} > x_{ub,i}$
        $u_{trial,i} = rand(x_{lb,i}\ ,\ x_{ub,i})$
      end if
    end for
  if $f(\boldsymbol{u_{trial}}) < f(\boldsymbol{u*})$
    $\boldsymbol{u*} = \boldsymbol{u_{trial}}$
    if $f(\boldsymbol{u_{trial}}) < f(\boldsymbol{u_{best}})$
      $\boldsymbol{u_{best}} = \boldsymbol{u_{trial}}$
    end if
  end if
return $\boldsymbol{u_{best}}$

Outer-Loop Multiple Objective Optimizer

Genetic algorithms are viable assets for solving global optimization problems which 1) may not be well-seeded by analytical initial guess information and 2) contain large numbers of decision variables, objectives and constraints. These methods are presently being introduced into numerous flight dynamics applications.[2,3]. The inner-loop optimizes only a single objective, in this case, the total mission ΔV. However, to understand the true behavior of the solution space, we employ a multi-objective optimization technique. The goal of a multi-objective optimization routine is to generate the Pareto front of the solution space.[12] This N-dimensional curve depicts the fundamental trade-offs between the objective functions considered in a study, beyond which no improvement in the solutions can be made. No point solution on the front completely dominates any other. Thus, traversing from one point on the front to any other in any direction requires degrading the performance of at least one objective.

The Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) is an effective method for tackling multi-objective problems, employing a fast non-dominated sort routine to evaluate a problem spanned by M objectives via the Pareto criterion.[6,13] It has proven effective on a wide range of flight dynamics applications.[1,2,11,13] Furthermore, a cap and optimize approach is used to enable simultaneous evaluation of all M objectives on a design.[13] The outer-loop, which operates solely on a population of integer decision vectors, selects from integer-encoded menus of caps for different objective functions and thus constrains the inner-loop problem, binning the secondary objective functions to a particular range for each candidate design. Thus, a user may evaluate N objectives in the same timespan it takes to evaluate just one. Any speed-limiting factors in evaluating the HOCP's solution space stem solely from inefficiencies within the inner-loop.

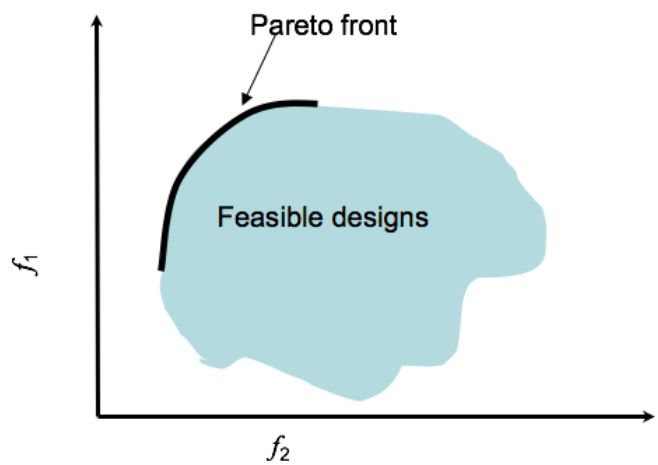


Figure 3: Concept illustration of the Pareto front for a mission with two objectives, f_1 and f_2.[11] Feasible designs exist in the blue region below the front, optimal solutions upon it, and no designs may exist in the region above it.

Null Gene Transcription

The null gene transcription allows the genetic algorithm outer-loop to insert and delete genes in accordance with the fitness of the trade space despite fixed array dimensions. The power of this

transcription to explore single vehicle trade spaces has been repeatedly demonstrated.[2] However, this transcription has never been applied to multi-spacecraft problems. While not presented here, we explore a multi-dimensional null gene transcription as a means to vary fleet size, in an upcoming paper investigating Near-Earth Asteroid multi-tours.

RESULTS

The preliminary results in this work serve to illustrate several capabilities. The first benchmark, Cassini's interplanetary cruise, validates the inner-loop's functionality applied to a challenging trajectory optimization problem with a known solution. With the inner-loop validated, the second benchmark problem explores the a cutting edge mission concept using the novel techniques in this work, notably the application of inter-spacecraft coordination constraints/objectives to a MVM.

Cassini's Interplanetary Cruise

As a benchmark test of the inner-loop's capability, the inner-loop's performance was validated by reproducing the Cassini interplanetary cruise. The nominal mission itinerary is compared to the itinerary found here, with minor differences. The parameters for the DE/best/2/bin local optimizer and the MBH global search algorithm are given in Table 1 and Table 2, respectively.

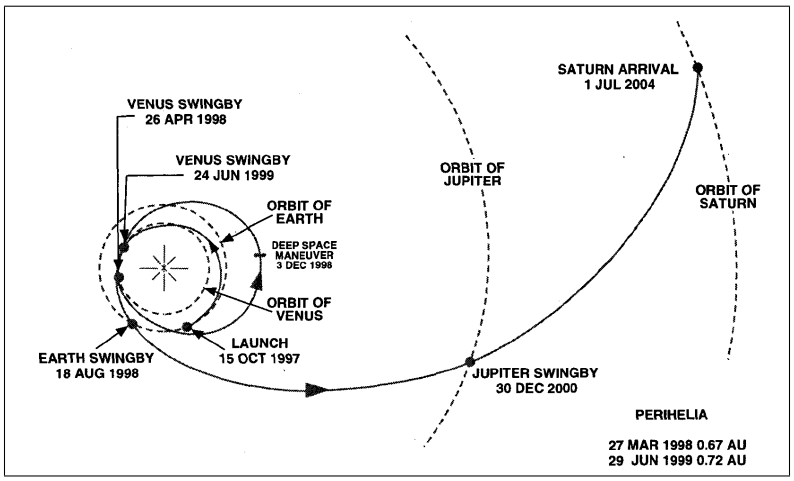


Figure 4: Cassini pre-launch design of nominal interplanetary cruise.[14].

Table 1

DE/best/2/bin Parameter	Value
Population Size	20
Generations	200
Difference Vector Throttle	1.0E-2
Launch Window (days)	90
Mutation Rate	0.05
C_3 bounds (km^2/s^2)	[15, 20]
RLA bounds (degrees)	[0, 360]
DLA bounds (degrees)	[-90, 90]
DSM Index bounds	[0, 1]
β (degrees)	[0, 360]

Table 2

MBH Parameter	Value
Maximum Global Search Hops	5
Local Hop Magnitude	2.5E-3
Improvement Criterion	1.0E-5
N_{max}	20
Maximum Runtime (minutes)	60

The minimum ΔV solution found in this inner-loop test is shown in Fig. 5. The resulting Cassini-like trajectory was produced from a one hour inner-loop run, and consumed a total ΔV of 696 m/s compared to the 550 m/s in the nominal pre-launch design of the equivalent cruise portion. This difference is likely due to the difference of event dates between the two trajectories. Time stamps of launch, flyby, and encounter events varied from nominal by under 15 days. Extra runtime would likely improve the performance of this solution, but determining the correct runtime is a problem-specific tuning task.

Ice Giant Multi-Mission

The 2013 Planetary Science Decadal Survey identifies both Uranus and Neptune as high priority science targets for future missions. However, no one spacecraft can perform flybys of both targets within the next 50 years.[1] This is, in part, due to the lacking conjunction geometry which existed during the Voyager spacecraft launches and occurs roughly once every 170 years. Therefore, in order to visit both targets, two vehicles are needed. This entails two possibilities: either two separate single-vehicle missions, or a dual vehicle multi-mission where both spacecraft are deployed from the same launch vehicle. The approach of designing the latter via cross-referenced independent grid searches for each spacecraft is insufficient to provide an understanding of the optimal solution space.

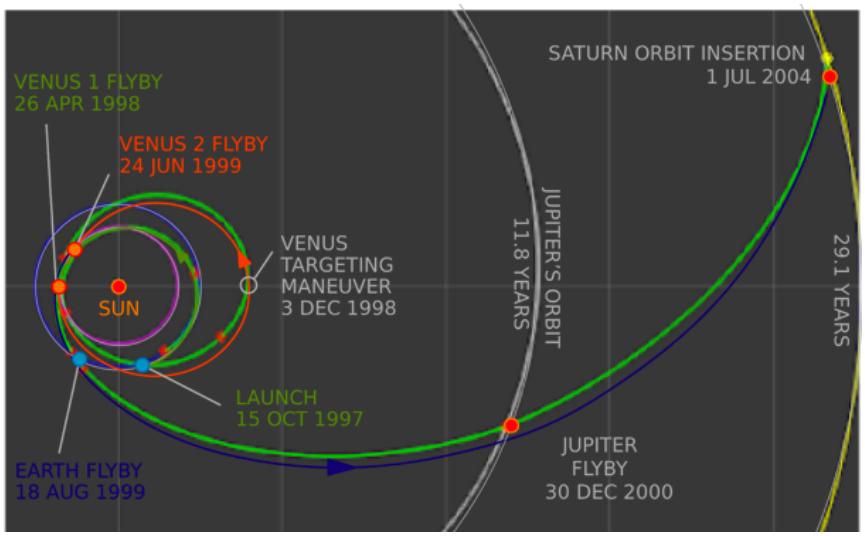


Figure 5: Cassini-like Trajectory found by the inner-loop (green) overlaying nominal trajectory (red to blue).

Independent searches, filtered *a posteriori* into a constrained space of shared launch opportunities, are likely to miss optimal opportunities and provide only a handful of point solutions rather than a full Pareto-optimal front. Here we present an early analysis of an Ice Giant Multi-Mission design using the novel MOMA HOCP technique described in this work. During the analysis period, the difference in ecliptic right ascension between Uranus and Neptune is considerable. This leads to the performance of one trajectory being inversely proportional to that of the other. Launched to the same asymptote, as the Uranus probe's trajectory improves, the Neptune probe's trajectory worsens and vice versa. Below, in Table 3, are the parameters bounding used in the first instantiation of the problem setup.

Table 3

NSGA-II Parameter	Value
Population Size	50
Generations	100
Maximum Flyby Targets	5
Mutation Rate	0.05
Main Objective	minimize total mission ΔV
Secondary Objectives	minimize time of flight; launch date (not sorted)
Launch Date Cap Menu	1/1/2025 : 1 year step : 1/1/2031
Phase Time of Flight Cap Menu	2 to 7 years in 1 year steps
Planetary Flyby Menu	Venus, Earth, Mars, Jupiter, Saturn
MBH Parameter	**Value**
Maximum Global Search Hops	5
Local Hop Magnitude	2.5E-2
Improvement Criterion	1.0E-5
N_{max}	20
Maximum Runtime (minutes)	30
DE Parameter	**Value**
Population Size	30
Generations	250
Difference Vector Throttle	1.0E-2
Launch Window (days)	90
Mutation Rate	0.05
C_3 bounds (km^2/s^2)	[5, 25]
RLA bounds (degrees)	[0, 360]
DLA bounds (degrees)	[-90, 90]
DSM Index bounds	[0, 1]
β (degrees)	[0, 360]

The first implementation of the MOMA-HOCP technique considered a relatively small search space using low C_3 values and a single multi-agent constraint: a shared launch asymptote for the two spacecraft. Subsequent larger scale studies begin from this same multi-agent terminal constraint. The bounds used in the first cut analysis are shown in Table 3. However, these particular problem bounds produced few numerically feasible solutions, and no practically feasible solutions.

The second version of this technique improves upon the first version by replacing of the DE inner-loop with an NLP solver *fmincon*. This allows for the explicit enforcement of linear and nonlinear constraints on the inner-loop problem. Specifically, a global TOF constraint is used, replacing the piece-wise phase caps approach. A performance increase occurred even though analytical derivatives were not used. The key additions in the second version are two new outer-loop multi-agent constraints: a minimum number of flyby genes shared by both spacecraft, and a minimum number of shared trajectory phase genes.

Minimum Shared Flyby Genes Constraint

For various reasons, a mission designer may want multiple spacecraft in a fleet to flyby identical targets, not necessarily at the same time. For example, the Voyager spacecraft performed staggered flybys of Jupiter and Saturn to leverage favorable turning angles. These staggered flybys also enabled interesting secondary science objectives including imaging the planets at different points

along their orbits. This constraint is a applied via an outer-loop decision variable menu. If a trial fleet is allowed N intermediate flybys to its destinations, then a user sets the shared flyby genes menu to allow $0 - -M$ shared flyby genes (where $M \leq N$).

One subtle caveat is that given the outer-loop's null gene transcription, flyby genes are not equivalent to physical flybys. With fixed non-null destination genes, the fleet size is constrained to be constant at the user-specified maximum number. But with intermediate flybys, the number of flybys can and does evolve over generations thanks to the null gene transcription's insertion and deletion behavior. For example, consider the following outer-loop decision vector:

$$X = \begin{bmatrix} LW & T_{\text{cap}} & C_3 & (3\,\text{shared flyby genes}) & \text{Venus} & \text{NULL} & \text{NULL} & \text{Jupiter} & \text{Uranus} \\ LW & T_{\text{cap}} & C_3 & (3\,\text{shared flyby genes}) & \text{Venus} & \text{NULL} & \text{NULL} & \text{Saturn} & \text{Neptune} \end{bmatrix} \tag{4}$$

The outer-loop has generated a vector where both spacecraft share at minimum the first three flyby genes, but 2 of those genes were generated null. So functionally, the inner-loop problem is only optimizing trajectories with one intermediate flyby each rather than 3. Applying this constraint allows the outer-loop to explore a greater diversity of solutions than with the shared launch asymptote constraint alone.

Minimum Shared Trajectory Phases Constraint

This boolean constraint enables the transcription of a trajectory where both payloads fly the exact same trajectory for a portion of the mission. This constraint may only be enabled if the minimum shared flyby genes constraint is also enabled. If so, then for any non-NULL flyby target genes, the two spacecraft are required to share the same trajectory until all specified shared flybys are completed. However, if any of the shared flyby genes are NULL, such as in Eq. (4), those flybys are pruned away by the inner-loop, and thus their corresponding trajectory phases do not exist. This constraint is enforced by the inner-loop via a header of shared parameters. The 2D hybrid decision vector is transformed into a 1D vector by the inner loop by parsing it into blocks in the following order: a header of shared decision variables including launch asymptote, and shared trajectory phases, followed by unique decision variables for spacecraft 1, then unique decision variables for spacecraft 2, etc. The indices that encode these blocks are stored in a separate structure parsed during the evaluation of an objective function. By making this transformation, the shared trajectory phases constraint may be strictly rather than loosely enforced. This constraint produced the least feasible solution space, due to the relative geometry of Uranus and Neptune.

Two modestly sized low C_3 studies were run on the CU Boulder Summit cluster followed by several otherwise equivalent high C_3 studies. Each of the studies included the shared launch asymptote constraint, but two studies separately included the shared flyby genes constraint and shared trajectory constraint. The low C_3 studies biased results towards higher numbers of intermediate flybys, while the higher C_3 studies biased more direct transfers. Without a more robust inner-loop transcription, the geometric complexity of adding more flybys yielded impractical results. The higher C_3 studies, however, begin to unearth more practical solutions due to less lengthy flyby permutations and less ΔV needed by the spacecraft to achieve the required energy changes. The parameters for the low C_3 studies are shown in Table 4. **Changes** made for the high C_3 studies are **bolded**. The choice of C_3 bounds for the latter study was made based on C_3 values obtained from JPL's pre-decadal study, specifically the point solution for an Ice Giant Dual Manifest mission using an

Table 4

Outer-Loop Parameter	**Value**
Population Size	72
Number of Workers	72
Generations	100
Maximum Intermediate Flyby Targets	5
Mutation Rate	10%
Main Objective	minimize fleet ΔV
Secondary Objective(s)	[minimax TOF]
Launch Window Menu	{[1/1/2030, 5/1/2030] : 4 mo : [9/1/2040, 1/1/2040]}
Global TOF Cap Menu	[10 years : 1 year : 16 years]
Planetary Flyby Menu	[Venus, Earth, Mars, Jupiter, Saturn]
Minimum Shared Flyby Genes Constraint Menu	[0, 1, 2, 3, 4]
Shared Trajectories Constraint (boolean)	[0, 1]
C_3 (km^{2}s^{-2}) Bounds Menu	{[0.0, 2.5] : 2.5 : [22.5, 25.0]}
MBH Parameter	**Value**
Maximum Global Search Hops	10,000
Local Hop Magnitude	$\pm$5% of current decision parameter value
Improvement Criterion	1.0E-5
N_{max}	25
Maximum Runtime (minutes)	60
Outer-Loop Parameter	**Value**
Maximum Intermediate Flyby Targets	**4**
Planetary Flyby Menu	**[Mars, Jupiter, Saturn]**
C_3 **(km^{2}s^{-2}) Bounds Menu**	{**[25.0, 30.0] : 5.0 : [210.0, 220.0]**}

optimal, purpose-built kick stage.[16]

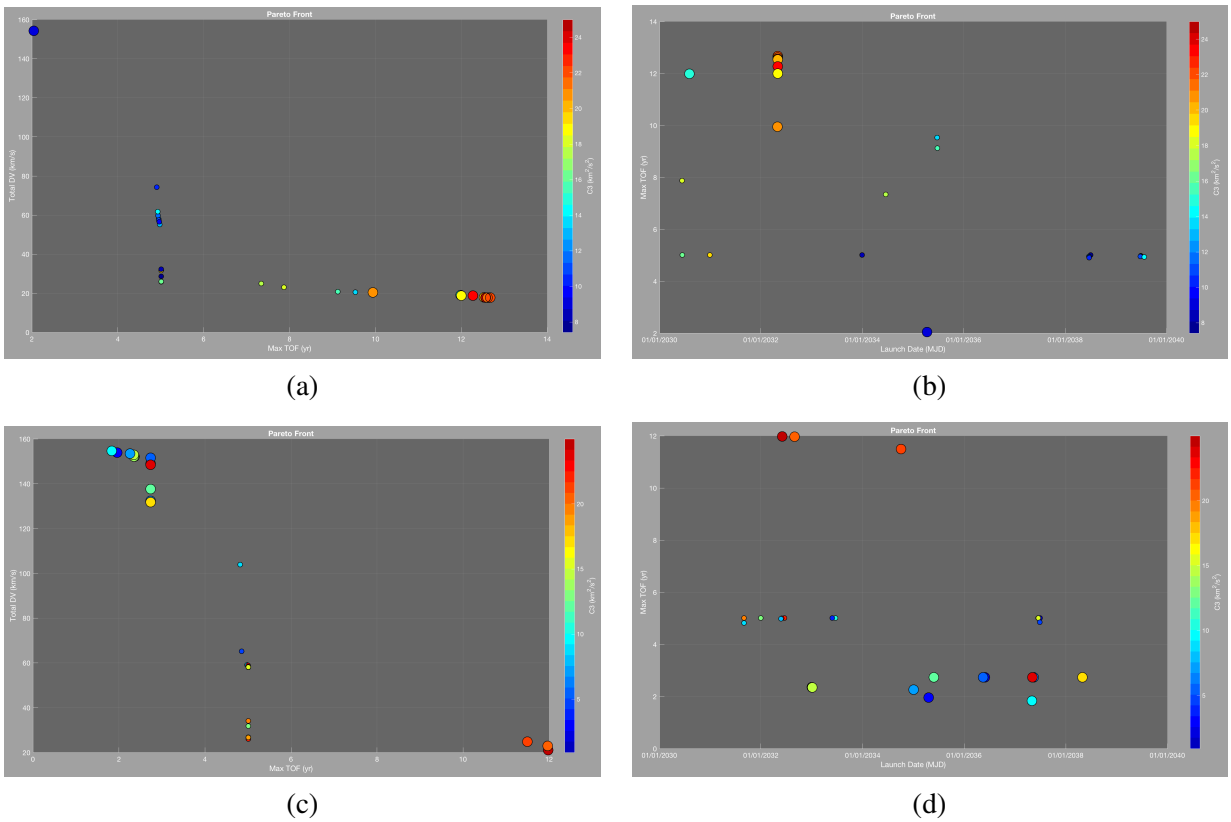


Figure 6: Low C_3 study results: (a) and (b) show, respectively, the Y-Z and X-Y views of Pareto front for the minimum shared flyby genes constraint (SHFB) study while (c) and (d) show analogous views for the shared trajectory phases constraint (SHTR) study.

Fig. 6 depicts the Pareto fronts of ΔV versus minimaxed TOF for the fleet at the 22nd outer-loop generation for the minimum shared flyby genes constraint SHFB and SHTR constraint studies. Each circle represents a spacecraft fleet. The size of each marker qualitatively reports the total number of intermediate flyby targets used by the spacecraft in the fleet with the least amount of flybys. In these plots, there are two sizes: zero intermediate flybys, and one intermediate flyby. The SHTR study produces a front with a noticeably sparser tail.

The low C_3 studies both found the minimum ΔV solution to be one that shares no trajectory phases at all, but where each spacecraft performs a flyby of one identical target (Jupiter, white orbit). The result is shown in Fig. 7. Red star markers depict events including launch, deep space maneuvers, and flybys. The Neptune probe performs a *solar system escape* maneuver to achieve the energy change needed to reach Neptune. That is, the spacecraft launches from Earth, performs a Jupiter flyby to target the Sun, and performs a solar flyby effectively achieving an Oberth Effect energy increase to reach Neptune, analogous to the maneuver designed by Arora et al. for a Kuiper

Table 5

Low C_3, SHFB	Date	C_3 (km^2/s^2)	RLA °	DLA °	ΔV (km/s)	Altitude (r_{planet})
Spacecraft 1	—	—	—	—	—	—
Launch	1 May 2032	25	319.4	4.1	—	—
DSM 1 (km/s)	5 Jul 2034	—	—	—	4.981	—
Flyby Jupiter	24 May 2036	—	—	—	—	50.0
DSM 2 (km/s)	24 May 2036	—	—	—	2.622	—
Encounter Uranus	7 May 2044	—	—	—	—	—
Spacecraft 2	—	—	—	—	—	—
Launch	1 May 2032	25	319.4	4.1	—	—
DSM 1 (km/s)	11 Jun 2032	—	—	—	4.528	—
Flyby Jupiter	27 Dec 2036	—	—	—	—	50.1
DSM 2 (km/s)	1 Mar 2039	—	—	—	5.591	—
Encounter Neptune	1 Jan 2045	—	—	—	—	—

Belt Object encounter.[15] This mission is summarized in Table 5 and has a total ΔV of 18 km/s with a TOF of 12.3 years for the slowest spacecraft to reach its target.

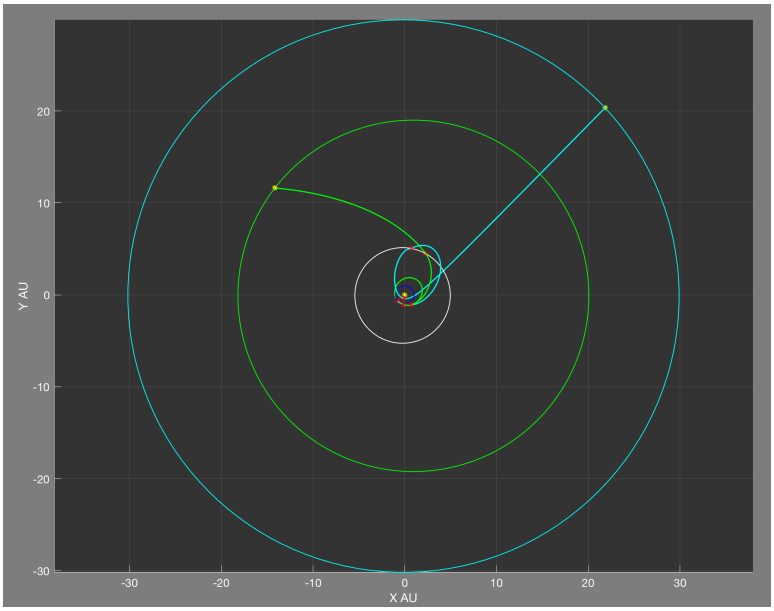
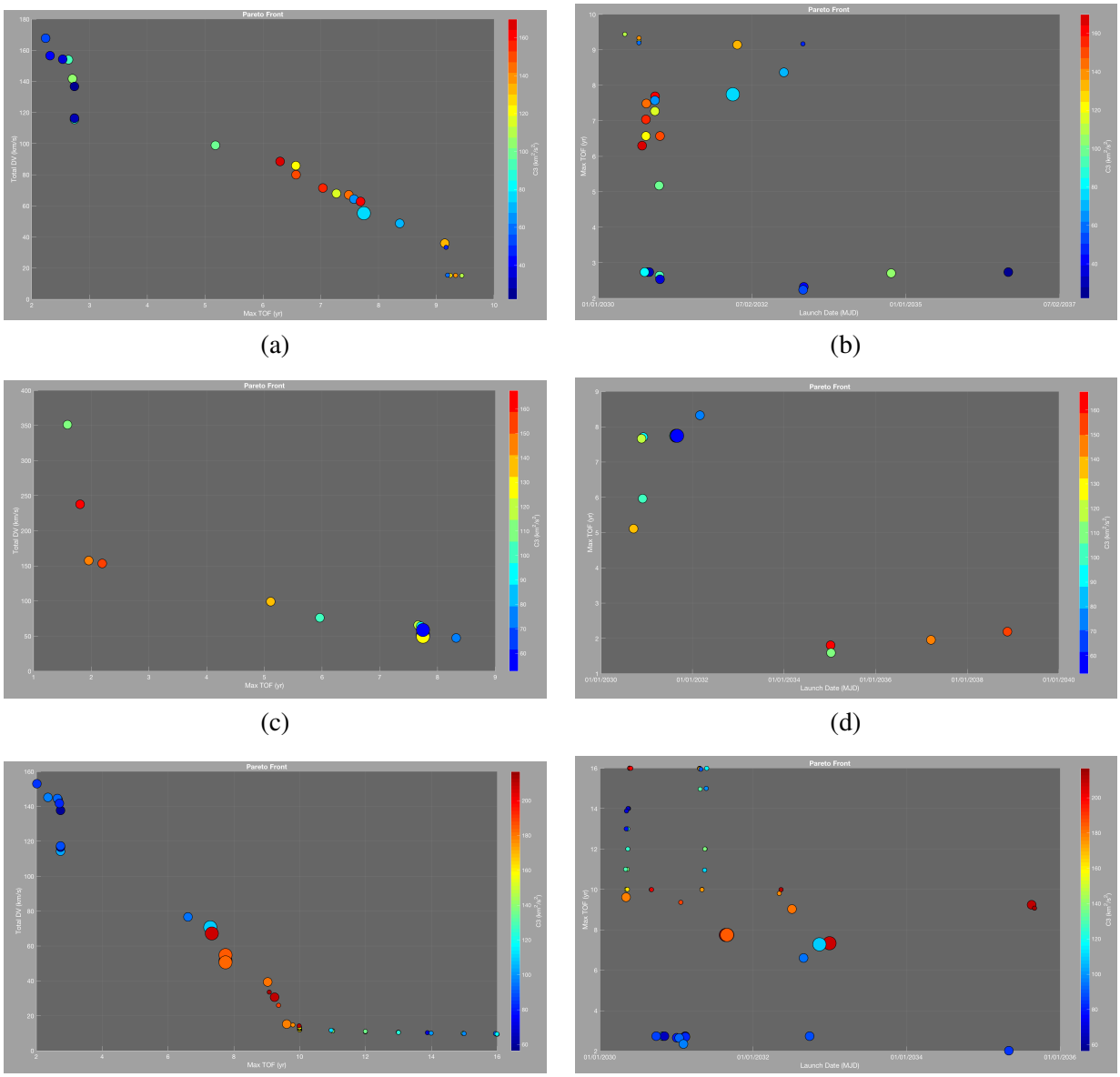


Figure 7: Minimum ΔV solution for the low C_3 SHFB study.



(a) (b) (c) (d) (e) (f)

Figure 8: High C_3 studies: All fleets shown after 22 outer-loop generations. (a) and (b) depict the Pareto front for the shared launch asymptote constraint only (SHLV) while (c) and (d) offer views of the SHTR study, and (e) and (f) depict the front for the SHFB study.

Table 6

High C_3, SHLV	**Date**	**C_3 (km^2/s^2)**	**RLA °**	**DLA °**	**ΔV (km/s)**	**Altitude (r_{planet})**
Spacecraft 1	—	—	—	—	—	—
Launch	8 Jun 2030	121.2	40.4	-5.2	—	—
DSM 1 (km/s)	8 Sep 2030	—	—	—	4.546	—
Encounter Uranus	17 Nov 2039	—	—	—	—	—
Spacecraft 2	—	—	—	—	—	—
Launch	8 Jun 2030	121.2	40.4	-5.2	—	—
DSM 1 (km/s)	13 Dec 2033	—	—	—	10.596	—
Flyby Jupiter	12 Oct 2034	—	—	—	—	93.9
DSM 2 (km/s)	20 Oct 2035	—	—	—	0.047	—
Encounter Neptune	9 Oct 2039	—	—	—	—	—

For the high C_3 studies, results where in higher abundance. The Pareto fronts for each are captured in Fig. 8. In (c) and (d) the structure of the front is markedly sparser than either of its less tightly constrained counterparts. Due to the unfavorable phasing of Uranus and Neptune during the study window, coupled with the constraint against using inner planet flybys to leverage an energy increase, numerically feasible solutions were few and far between. No practically feasible or near-practically feasible solutions were found in either the low or high C_3 SHTR studies. Among other concerns to be addressed in future work, a more thorough search is needed. In the SHFB study shown in (e) and (f), the Pareto front exhibits a distinct tail at the 10-year TOF mark, where local families of otherwise equivalent ΔV solutions exist for all TOFs over 10-years. Across all studies, one observes that launch dates later than 1/1/2032 have markedly fewer, if any, solutions. This is due mainly to the fact that Uranus' heliocentric angular velocity is greater than Neptune's, which means the Ice Giants' relative difference in ecliptic right ascension grows as time moves forward. Conversely, the earlier the launch date, likely up until the era of the Voyager launches, the number of feasible opportunities improves as the planets approach conjunction.

In Fig. 8a, the high C_3 SHLV study's minimum ΔV solution finds a trajectory whose geometry suggests that the Neptune probe would benefit from an Earth flyby, while the Uranus probe could potentially realize a ballistic trajectory with a slightly higher C_3. Since the bounds of this study did not allow for inner planet flybys, an Earth flyby was not an available option, but would arguably eliminate the large DSM if a flyby opportunity exists. In (b), the high C_3 SHFB study, the minimum ΔV solution in this study is the most promising found among all studies run yet. Its primary shortcoming is the clear need for a Jupiter flyby which was simply not found due to the relatively small span of the search given by the small outer-loop population. Secondarily, the 'DSM' on the Neptune trajectory is performed on the launch date which implies a need for a higher C_3 to eliminate this maneuver. Variants of this trajectory are found across different flight times, as captured by the tail of the Pareto front. Mission itineraries for the high C_3 SHLV and SHFB studies are given in Table 6 and Table 7 respectively.

Table 7

High C_3, SHFB	**Date**	**C_3 (km^2/s^2)**	**RLA °**	**DLA °**	**ΔV (km/s)**	**Altitude (r_{planet})**
Spacecraft 1	—	—	—	—	—	—
Launch	16 May 2030	150.0	339.3	-37.0	—	—
DSM 1 (km/s)	30 May 2034	—	—	—	7.584	—
Encounter Uranus	16 May 2046	—	—	—	—	—
Spacecraft 2	—	—	—	—	—	—
Launch	16 May 2030	150.0	339.3	-37.0	—	—
DSM 1 (km/s)	16 May 2030	—	—	—	1.854	—
Encounter Neptune	16 May 2046	—	—	—	—	—

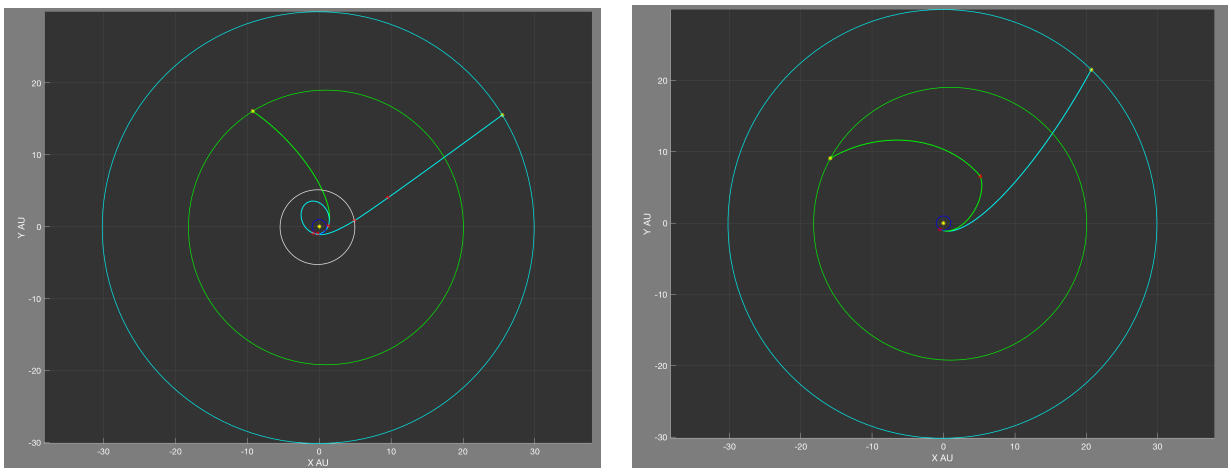


Figure 9: (a) The high C_3 SHLV study's minimum ΔV solution and (b), the high C_3 SHFB study's, the minimum ΔV solution.

CONCLUSIONS

In this paper, we have described the formulation of a novel technique for Multi-Objective, Multi-Agent, Hybrid Optimal Control optimization applied to interplanetary multi-spacecraft global trajectory optimization. The results of this work demonstrate both promise for the current technique, while also highlighting the need for improvement in the reliability of the inner-loop, and the population size of the outer-loop. Higher C_3 studies produced artificially lower ΔV solutions in part due to the inner-loop's inability to optimize complex trajectories with multiple flybys. Optimal solutions produced by the outer-loop also highlighted need for flybys that the outer-loop did not find autonomously, which is likely due to the small population size used. This shortcoming will be addressed with greater computing resources. Future work will focus on improving the robustness of the inner-loop, acquiring more significant distributed computing resources to evaluate larger

search spaces, and exploring performance under the influence of a wider variety of coordination constraints/objectives.

ACKNOWLEDGMENTS

The authors would like to acknowledge the members of the Global Trajectory Optimization Lab at NASA Goddard Space Flight Center, particularly Dr. Jacob Englander and Dr. Kyle Hughes, for lending their expertise in support of this research. We also acknowledge the support of the Goddard Space Flight Center's Internal Research and Development (IRAD) program which provided funding for this work.

REFERENCES

[1] Kyle. M. Hughes, Jeremy M. Knittel, Jacob A. Englander, "Gravity-Assist Trajectories to the Ice Giants: An Automated Method to Catalog Mass- or Time- Optimal Solutions," AAS/AIAA Astrodynamics Specialist Conference; 20–24 Aug. 2017.
[2] J. Englander, B. Conway, and T. Williams, "Automated Mission Planning via Evolutionary Algorithms," Journal of Guidance, Control, and Dynamics, Vol. 35, No. 6, 2012, pp. 1878–1887.
[3] M. Vavrina and K. Howell,"Multiobjective Optimization of Low-Thrust Trajectories Using a Genetic Algorithm Hybrid," AAS/AIAA Space Flight Mechanics Meeting, February 2009. AAS 09-141.
[4] Angelia Nedic, Asuman Ozdaglar, and Pablo A. Parrilo, "Constrained Consensus and Optimization in Multi-Agent Networks," IEEE Transactions on Automatic Control, Vol. 55, No. 4, April 2010.
[5] Marc Pujol-Gonzalez, "Multi-Agent Coordination: DCOPs and Beyond," Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, July 2011.
[6] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, April 2002.
[7] Kalyanmoy Deb and Himanshu Gupta, "Introducing Robustness in Multi-Objective Optimization," The Journal of Evolutionary Computation, Vol. 4, 2006.
[8] Izzo, D. (2010). Global Optimization and Space Pruning for Spacecraft Trajectory Design. In B. Conway (Ed.), Spacecraft Trajectory Optimization (Cambridge Aerospace Series, pp. 178–201). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511778025.008
[9] "PaGMO (Parallel Global Multiobjective Optimizer)," http://pagmo.sourceforge.net/pagmo/index.html.
[10] T. McConaghy, "GALLOP Version 4.5 User's Guide." School of Aeronautics and Astronautics, Purdue University, 2005.
[11] Vavrina, Matthew A., Englander, Jacob Aldo, Ghosh, Alexander R., "Coupled Low-thrust Trajectory and System Optimization via Multi-Objective Hybrid Optimal Control," AAS/AIAA Spaceflight Mechanics Meeting; 25th; 11–15 Jan 2015.
[12] V. Pareto, Manuale di Economica Politica. Milano, Italy: Societa Editrice Libraria, 1906. translated into English by A. Schwier as Manual of Political Economy, MacMillan Press, New York, 1971.
[13] J. Englander, M. Vavrina, and A. Ghosh, "Multi-Objective Hybrid Optimal Control for Multiple-Flyby Low-Thrust Mission Design," AAS/AIAA Space Flight Mechanics Meeting, January 2015.

[14] Troy D. Goodson, Donald L. Gray, Yungsun Hahn, and Fernando Peralta, "Cassini Maneuver Experience: Launch and Early Cruise," Proc. of the American Institue of Aeronautics and Astronautics, 1998 Guidance, Navigation, and Control Conference, Boston MA, 10 Aug. 1998, JPL Beacon eSpace collection JPL TRS 1992+, file 98-0900.pdf
[15] N. Arora, N. Strange, and L. Alkalai, "Trajectories for a near term mission to the Interstellar Medium," presented at the AAS/AIAA Astrodynamics Specialist Conference, Vail, Colorado, 2015.
[16] Hofstdater, Mark; Simon, Amy; "Ice Giants Pre-Decadal Survey Mission Study Report," June 2017.
[17] J. A. Englander and A. C. Englander, "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.